



# Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect

Revision: r0p1

## Technical Reference Manual

**Non-Confidential**

**Issue 04**

Copyright © 2022–2024 Arm Limited (or its affiliates). 102756\_0001\_04\_en  
All rights reserved.



# Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect

## Technical Reference Manual

Copyright © 2022–2024 Arm Limited (or its affiliates). All rights reserved.

## Release information

### Document history

Issue	Date	Confidentiality	Change
0000-01	11 May 2022	Confidential	First Beta release for r0p0
0000-02	22 February 2023	Non-Confidential	First Early Access release for r0p0
0001-03	27 June 2023	Non-Confidential	First Early Access release for r0p1
0001-04	5 January 2024	Non-Confidential	Second Early Access release for r0p1

## Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND

REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

Copyright © 2022–2024 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

## Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

## Product Status

The information in this document is Final, that is for a developed product.

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

## Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

# Contents

<b>1. Introduction.....</b>	<b>24</b>
1.1 Product revision status.....	24
1.2 Intended audience.....	24
1.3 Conventions.....	24
1.4 Useful resources.....	26
<b>2. CoreLink NI-710AE Network-on-Chip Interconnect.....</b>	<b>28</b>
2.1 Key features.....	28
2.1.1 Test features.....	30
2.2 Compliance.....	30
2.2.1 Supported AMBA protocol features.....	30
2.2.2 Unsupported AMBA protocol features.....	33
2.2.3 TrustZone technology and security.....	34
2.3 Product design flow.....	35
2.4 Product documentation.....	36
2.5 Interfaces.....	38
2.6 Architecture overview.....	39
2.7 Functional units.....	41
2.7.1 ASNI.....	42
2.7.2 AMNI.....	43
2.7.3 HSNI.....	44
2.7.4 HMNI.....	47
2.7.5 PMNI.....	47
2.7.6 Power and Clock Domain Crossing unit.....	48
2.7.7 Router.....	49
2.7.8 SERDES unit.....	49
2.7.9 Performance Monitoring Unit.....	49
2.8 Configurable options.....	50
2.8.1 FuSa parameters.....	52
2.8.2 ASNI configuration options.....	55
2.8.3 AMNI configuration options.....	58
2.8.4 HSNI configuration options.....	61

2.8.5 HMNI configuration options.....	63
2.8.6 PMNI configuration options.....	65
2.8.7 PCDC configuration options.....	66
2.8.8 Router configuration options.....	66
<b>3. Fault Detection and Control mechanisms.....</b>	<b>68</b>
3.1 External interface protection.....	70
3.1.1 AMBA interfaces.....	70
3.1.2 Asynchronous interfaces.....	71
3.1.3 Miscellaneous external interface protection.....	73
3.2 Logic protection.....	74
3.3 Internal network protection.....	76
3.4 Memory access protection and the Access Protection Unit.....	77
3.4.1 APU architecture.....	78
3.4.2 APU definitions.....	79
3.4.3 APU transaction filtering.....	80
3.4.4 Configuring the APU.....	82
3.5 Device isolation with IDM wire interface.....	91
3.5.1 Soft reset entry mode scenarios.....	93
3.6 Hang detection.....	94
3.7 Clock protection.....	94
3.8 Reset protection.....	95
3.8.1 Transient faults.....	95
3.8.2 Internal reset faults.....	96
3.8.3 FuSa reset requirements.....	96
3.9 Fault Management Unit.....	97
3.9.1 FMU error logging process.....	98
3.9.2 FMU error record table.....	99
3.9.3 Controlling which safety mechanisms report errors to the FMU.....	99
3.9.4 Configuring which errors are critical.....	100
3.9.5 FMU interrupts.....	100
3.9.6 FMU error injection.....	101
3.9.7 Software initialization and the FMU.....	101
3.9.8 FMU register protection mechanism.....	103
3.9.9 Fault Management Unit resets.....	104
<b>4. Power, clock, and reset management.....</b>	<b>106</b>

4.1 Power.....	106
4.1.1 Power state requirements and characteristics.....	107
4.1.2 P-Channel low-power interface.....	108
4.2 Clocks.....	109
4.2.1 Levels of clock gating.....	109
4.2.2 Hierarchical clock gating.....	110
4.2.3 Q-Channel low-power interface.....	110
4.2.4 External clock controller.....	111
4.2.5 Clock domain wake up.....	112
4.2.6 Network FIFO and clocking function.....	112
4.3 Power control.....	113
4.3.1 Power control sequences.....	116
4.3.2 External power domain boundaries.....	117
4.3.3 AHB address phase buffering in HSNIs.....	119
4.4 Clock and reset control.....	120
4.4.1 Clock control sequences.....	121
4.4.2 Reset control sequences.....	122
<b>5. Node, interface, and transaction identifiers.....</b>	<b>124</b>
5.1 Node ID calculation.....	126
5.2 Interface ID calculation.....	127
5.3 Output ID calculation.....	128
<b>6. Data width conversion.....</b>	<b>130</b>
6.1 Upsizing AXI and ACE-Lite data width function.....	130
6.1.1 Upsizing INCR bursts.....	131
6.1.2 Upsizing WRAP bursts.....	131
6.2 Downsizing AXI and ACE-Lite data width function.....	132
6.2.1 Downsizing INCR bursts.....	132
6.2.2 Downsizing WRAP bursts.....	133
6.2.3 Downsizing FIXED bursts.....	133
6.3 User signals.....	133
6.4 Flit resizing and collating.....	136
<b>7. Transaction handling.....</b>	<b>137</b>
7.1 Exclusive and locked accesses.....	137
7.2 AHB locked transfers.....	138

7.3 Memory tagging support.....	138
<b>8. Reliability, Availability, and Serviceability.....</b>	<b>140</b>
8.1 Support for transporting data parity, ECC, and poison information.....	140
<b>9. Secure and Non-secure accesses.....</b>	<b>141</b>
9.1 Security access permissions of AXI requests.....	141
9.2 Security access permissions of AHB requests.....	141
9.3 Security access permissions of APB requests.....	142
9.4 Register security attribute and security classification.....	144
9.5 Secure access register.....	144
9.6 Secure debug.....	145
9.7 Interrupt and error logging register security.....	145
<b>10. Interconnect Device Management.....</b>	<b>146</b>
10.1 IDM and device discovery.....	148
10.2 Timeout detection through the IDM block.....	148
10.3 Error logging through the IDM block.....	149
10.4 IDM soft reset mode.....	150
10.4.1 Hardware-initiated entry based on timeout detection.....	151
10.4.2 Software-initiated entry.....	152
10.4.3 Reset initialization input pin.....	152
10.5 IDM access control.....	153
10.6 Soft reset use case examples for xSNIs and xMNIs.....	154
10.7 Access control use case example for xMNIs and xSNIs.....	159
10.8 Example interrupt handling sequence.....	160
10.9 Soft reset sequence.....	161
<b>11. Address decode and mapping.....</b>	<b>164</b>
11.1 ASNI address decode.....	164
11.2 HSNI address decode.....	164
11.3 PMNI address decode.....	164
11.4 Address striping.....	165
11.5 Remap.....	166
<b>12. Transaction tracking and ordering.....</b>	<b>170</b>
12.1 Transaction reorder buffers.....	170
12.2 Cyclic Dependency Avoidance Scheme.....	170



12.2.1 Single completer for each ID.....	171
12.2.2 Ordered Write Observation.....	171
<b>13. Traffic arbitration schemes.....</b>	<b>172</b>
13.1 Resource Planes.....	172
13.2 Quality of Service.....	173
13.2.1 Hard bandwidth regulation.....	173
13.2.2 Soft bandwidth regulation.....	181
13.2.3 QoS value override programmable registers.....	183
13.3 Memory System Resource Partitioning and Monitoring.....	184
<b>14. Performance monitoring.....</b>	<b>185</b>
14.1 PMU organization.....	185
14.2 PMU system programming.....	187
14.2.1 Set up the PMU counters.....	187
14.2.2 Program PMU snapshot functionality.....	188
14.2.3 Program PMU interrupts.....	189
14.2.4 Performance monitoring and Secure Debug.....	189
14.3 AMNI performance events.....	190
14.4 ASNI performance events.....	191
14.5 Data bandwidth at ASNI and AMNI.....	193
14.5.1 Read and write bandwidth at ASNI and AMNI.....	193
14.5.2 Delays at ASNI and AMNI because of backpressure.....	194
14.5.3 Delays at ASNI because of structural backpressure.....	194
14.6 AHB performance event mapping.....	194
14.7 HSNI performance events.....	195
14.8 HMNI performance events.....	197
14.9 Data bandwidth at HSNI and HMNI.....	199
14.9.1 Read and write bandwidth at HSNI and HNMI.....	199
14.9.2 Delays at HSNI and HMNI because of backpressure.....	200
14.9.3 Delays at HSNI because of structural backpressure.....	200
14.10 PMNI performance events.....	200
<b>15. Error handling and interrupts.....</b>	<b>202</b>
15.1 IDM error logging interrupts and status flags.....	202
15.2 IDM error logging registers.....	203
15.3 IDM error processing sequence.....	204

15.4 Interrupts.....	204
15.5 Interrupt generation.....	205
15.6 Error interrupt handler flow.....	206
15.7 Error handling and interrupt security.....	207
15.8 Error responses.....	208
<b>16. Programmers model.....</b>	<b>212</b>
16.1 Requirements of configuration register reads and writes.....	213
16.2 Discovery.....	214
16.2.1 Configuration nodes.....	214
16.2.2 Discovery flow.....	221
16.3 Configuration register address region calculation.....	223
16.4 Configuration address space example for design with multiple voltage, power, and clock domains.....	224
16.5 Global register summary.....	227
16.5.1 Global node_type register.....	227
16.5.2 Global child_node_info register.....	228
16.5.3 Global vd_pointers register.....	229
16.5.4 Global secure_access register.....	230
16.5.5 Global peripheral_id4 register.....	231
16.5.6 Global peripheral_id5 register.....	232
16.5.7 Global peripheral_id6 register.....	233
16.5.8 Global peripheral_id7 register.....	234
16.5.9 Global peripheral_id0 register.....	234
16.5.10 Global peripheral_id1 register.....	235
16.5.11 Global peripheral_id2 register.....	236
16.5.12 Global peripheral_id3 register.....	237
16.5.13 Global component_id0 register.....	238
16.5.14 Global component_id1 register.....	239
16.5.15 Global component_id2 register.....	240
16.5.16 Global component_id3 register.....	241
16.6 Voltage domain register summary.....	242
16.6.1 Voltage domain node_type register.....	242
16.6.2 Voltage domain child_node_info register.....	243
16.6.3 Voltage domain pd_pointers register.....	244
16.6.4 Voltage domain secure_access register.....	245
16.7 Power domain register summary.....	246

16.7.1 Power domain node_type register.....	247
16.7.2 Power domain child_node_info register.....	248
16.7.3 Power domain cd_pointers register.....	249
16.7.4 Power domain endpoint_pd_irq_status register.....	250
16.7.5 Power domain endpoint_pd_irq_control register.....	251
16.7.6 Power domain idm_pd_error_status register.....	252
16.7.7 Power domain idm_pd_error_control register.....	253
16.7.8 Power domain idm_pd_timeout_status register.....	254
16.7.9 Power domain idm_pd_timeout_control register.....	256
16.7.10 Power domain idm_pd_reset_status register.....	257
16.7.11 Power domain idm_pd_reset_control register.....	258
16.7.12 Power domain idm_pd_access_status register.....	259
16.7.13 Power domain idm_pd_access_control register.....	260
16.7.14 Power domain endpoint_pd_irq_status_ns register.....	261
16.7.15 Power domain endpoint_pd_irq_control_ns register.....	262
16.7.16 Power domain idm_pd_error_status_ns register.....	264
16.7.17 Power domain idm_pd_error_control_ns register.....	265
16.7.18 Power domain idm_pd_timeout_status_ns register.....	266
16.7.19 Power domain idm_pd_timeout_control_ns register.....	267
16.7.20 Power domain idm_pd_reset_status_ns register.....	268
16.7.21 Power domain idm_pd_reset_control_ns register.....	269
16.7.22 Power domain idm_pd_access_status_ns register.....	270
16.7.23 Power domain idm_pd_access_control_ns register.....	272
16.7.24 Power domain secure_access register.....	273
16.8 Clock domain register summary.....	274
16.8.1 Clock domain node_type register.....	274
16.8.2 Clock domain child_node_info register.....	275
16.8.3 Clock domain component_pointers register.....	276
16.8.4 Clock domain secure_access register.....	277
16.9 PMU register summary.....	278
16.9.1 PMU node_type register.....	280
16.9.2 PMU secure_access register.....	281
16.9.3 PMU pmevcntr0 register.....	282
16.9.4 PMU pmevcntr1 register.....	283
16.9.5 PMU pmevcntr2 register.....	284
16.9.6 PMU pmevcntr3 register.....	285

16.9.7 PMU pmevcntr4 register.....	286
16.9.8 PMU pmevcntr5 register.....	287
16.9.9 PMU pmevcntr6 register.....	288
16.9.10 PMU pmevcntr7 register.....	289
16.9.11 PMU pmccntr_l register.....	290
16.9.12 PMU pmccntr_u register.....	291
16.9.13 PMU pmevtyper0 register.....	292
16.9.14 PMU pmevtyper1 register.....	293
16.9.15 PMU pmevtyper2 register.....	294
16.9.16 PMU pmevtyper3 register.....	295
16.9.17 PMU pmevtyper4 register.....	296
16.9.18 PMU pmevtyper5 register.....	298
16.9.19 PMU pmevtyper6 register.....	299
16.9.20 PMU pmevtyper7 register.....	300
16.9.21 PMU pmssr register.....	301
16.9.22 PMU pmovssr register.....	302
16.9.23 PMU pmccntr_l register.....	303
16.9.24 PMU pmccntr_u register.....	304
16.9.25 PMU pmevcntr0 register.....	305
16.9.26 PMU pmevcntr1 register.....	306
16.9.27 PMU pmevcntr2 register.....	307
16.9.28 PMU pmevcntr3 register.....	308
16.9.29 PMU pmevcntr4 register.....	309
16.9.30 PMU pmevcntr5 register.....	310
16.9.31 PMU pmevcntr6 register.....	311
16.9.32 PMU pmevcntr7 register.....	312
16.9.33 PMU pmsscr register.....	313
16.9.34 PMU pmcntenset register.....	314
16.9.35 PMU pmcntenclr register.....	317
16.9.36 PMU pmintenset register.....	319
16.9.37 PMU pmintenclr register.....	322
16.9.38 PMU pmovsclr register.....	325
16.9.39 PMU pmovsset register.....	328
16.9.40 PMU pmcccgr register.....	331
16.9.41 PMU pmcfgr register.....	332
16.9.42 PMU pmcr register.....	333

16.10 APU register summary.....	334
16.10.1 APU PRBAR_LOW register.....	335
16.10.2 APU PRBAR_HIGH register.....	336
16.10.3 APU PRLAR_LOW register.....	337
16.10.4 APU PRLAR_HIGH register.....	338
16.10.5 APU PRID_LOW register.....	338
16.10.6 APU PRID_HIGH register.....	339
16.10.7 APU APU_CTLR register.....	340
16.10.8 APU APU_IIDR register.....	342
16.11 FMU register summary.....	343
16.11.1 FMU FMU_ERR_FR_0 register.....	344
16.11.2 FMU FMU_ERR_CTLR_0 register.....	345
16.11.3 FMU FMU_ERR_STATUS register.....	346
16.11.4 FMU FMU_ERR_MISCO register.....	348
16.11.5 FMU FMU_ERR_FR register.....	349
16.11.6 FMU FMU_ERR_CTLR register.....	350
16.11.7 FMU FMU_ERRGSR register.....	351
16.11.8 FMU FMU_ERRIIDR register.....	352
16.11.9 FMU FMU_KEY register.....	353
16.11.10 FMU FMU_SMEN register.....	354
16.11.11 FMU FMU_SMINJERR register.....	356
16.11.12 FMU FMU_SMINFO register.....	357
16.11.13 FMU FMU_ERRDEVARCH register.....	358
16.11.14 FMU FMU_ERRDEVID register.....	359
16.11.15 FMU FMU_ERRPIDR0 register.....	360
16.11.16 FMU FMU_ERRPIDR1 register.....	361
16.11.17 FMU FMU_ERRPIDR2 register.....	362
16.11.18 FMU FMU_ERRPIDR3 register.....	363
16.11.19 FMU FMU_ERRPIDR4 register.....	364
16.11.20 FMU FMU_ERRCIDR0 register.....	365
16.11.21 FMU FMU_ERRCIDR1 register.....	366
16.11.22 FMU FMU_ERRCIDR2 register.....	367
16.11.23 FMU FMU_ERRCIDR3 register.....	368
16.12 ASNI register summary.....	369
16.12.1 ASNI node_type register.....	371
16.12.2 ASNI node_info register.....	372

16.12.3 ASNI secure_access register.....	375
16.12.4 ASNI pmusela register.....	376
16.12.5 ASNI pmuselb register.....	377
16.12.6 ASNI interface_id_0_3 register.....	378
16.12.7 ASNI num_sub_features register.....	379
16.12.8 ASNI sub_feature_0_type register.....	380
16.12.9 ASNI sub_feature_0_pointer register.....	381
16.12.10 ASNI burst_split_control register.....	382
16.12.11 ASNI address_remap register.....	384
16.12.12 ASNI hang_detector_ctrl register.....	385
16.12.13 ASNI silicon_debug register.....	389
16.12.14 ASNI qosctl register.....	390
16.12.15 ASNI wdatthrs register.....	391
16.12.16 ASNI arqos_value register.....	392
16.12.17 ASNI awqos_value register.....	393
16.12.18 ASNI atqosot register.....	394
16.12.19 ASNI arqosot register.....	395
16.12.20 ASNI awqosot register.....	396
16.12.21 ASNI axqosot register.....	397
16.12.22 ASNI qosrdpk register.....	398
16.12.23 ASNI qosrdbur register.....	399
16.12.24 ASNI qosrdavg register.....	400
16.12.25 ASNI qoswrpk register.....	401
16.12.26 ASNI qoswrbur register.....	402
16.12.27 ASNI qoswragv register.....	403
16.12.28 ASNI qoscompk register.....	404
16.12.29 ASNI qoscombur register.....	405
16.12.30 ASNI qoscomavg register.....	406
16.12.31 ASNI qosrdbqv register.....	407
16.12.32 ASNI qoswrbqv register.....	408
16.12.33 ASNI qoscombqv register.....	409
16.12.34 ASNI read_channel_mpam_override register.....	410
16.12.35 ASNI write_channel_mpam_override register.....	411
16.12.36 ASNI idm_device_id register.....	412
16.12.37 ASNI idm_config register.....	413
16.12.38 ASNI idm_errctlr register.....	414

16.12.39 ASNI idm_errstatus register.....	416
16.12.40 ASNI idm_erraddr_lsb register.....	419
16.12.41 ASNI idm_erraddr_msb register.....	419
16.12.42 ASNI idm_errmisc0 register.....	420
16.12.43 ASNI idm_errmisc1 register.....	421
16.12.44 ASNI idm_access_control register.....	422
16.12.45 ASNI idm_access_status register.....	423
16.12.46 ASNI idm_access_readid register.....	425
16.12.47 ASNI idm_access_writeid register.....	426
16.12.48 ASNI idm_reset_control register.....	427
16.12.49 ASNI idm_reset_status register.....	429
16.12.50 ASNI idm_reset_readid register.....	431
16.12.51 ASNI idm_reset_writeid register.....	432
16.12.52 ASNI idm_timeout_control register.....	433
16.12.53 ASNI idm_timeout_value register.....	434
16.12.54 ASNI idm_interrupt_status register.....	435
16.12.55 ASNI idm_interrupt_mask register.....	436
16.12.56 ASNI idm_errstatus_ns register.....	437
16.12.57 ASNI idm_erraddr_lsb_ns register.....	440
16.12.58 ASNI idm_erraddr_msb_ns register.....	440
16.12.59 ASNI idm_errmisc0_ns register.....	441
16.12.60 ASNI idm_errmisc1_ns register.....	442
16.12.61 ASNI idm_access_status_ns register.....	443
16.12.62 ASNI idm_access_readid_ns register.....	445
16.12.63 ASNI idm_access_writeid_ns register.....	446
16.12.64 ASNI idm_reset_status_ns register.....	447
16.12.65 ASNI idm_reset_readid_ns register.....	448
16.12.66 ASNI idm_reset_writeid_ns register.....	449
16.12.67 ASNI idm_interrupt_status_ns register.....	450
16.12.68 ASNI idm_interrupt_mask_ns register.....	451
16.13 AMNI register summary.....	452
16.13.1 AMNI node_type register.....	454
16.13.2 AMNI node_info register.....	455
16.13.3 AMNI secure_access register.....	456
16.13.4 AMNI pmusela register.....	457
16.13.5 AMNI pmuselb register.....	458

16.13.6 AMNI interface_id_0_3 register.....	460
16.13.7 AMNI num_sub_features register.....	461
16.13.8 AMNI sub_feature_0_type register.....	461
16.13.9 AMNI sub_feature_0_pointer register.....	462
16.13.10 AMNI node_features register.....	463
16.13.11 AMNI silicon_debug register.....	464
16.13.12 AMNI qos_accept_control register.....	465
16.13.13 AMNI cmoovrd register.....	466
16.13.14 AMNI rddata_agg_control register.....	467
16.13.15 AMNI interrupt_status register.....	468
16.13.16 AMNI interrupt_mask register.....	469
16.13.17 AMNI interrupt_status_ns register.....	470
16.13.18 AMNI interrupt_mask_ns register.....	471
16.13.19 AMNI idm_device_id register.....	472
16.13.20 AMNI idm_config register.....	473
16.13.21 AMNI idm_errctlr register.....	474
16.13.22 AMNI idm_errstatus register.....	476
16.13.23 AMNI idm_erraddr_lsb register.....	479
16.13.24 AMNI idm_erraddr_msb register.....	479
16.13.25 AMNI idm_errmisc0 register.....	480
16.13.26 AMNI idm_errmisc1 register.....	481
16.13.27 AMNI idm_access_control register.....	482
16.13.28 AMNI idm_access_status register.....	483
16.13.29 AMNI idm_access_readid register.....	485
16.13.30 AMNI idm_access_writeid register.....	486
16.13.31 AMNI idm_reset_control register.....	487
16.13.32 AMNI idm_reset_status register.....	489
16.13.33 AMNI idm_reset_readid register.....	491
16.13.34 AMNI idm_reset_writeid register.....	492
16.13.35 AMNI idm_timeout_control register.....	493
16.13.36 AMNI idm_timeout_value register.....	494
16.13.37 AMNI idm_interrupt_status register.....	495
16.13.38 AMNI idm_interrupt_mask register.....	496
16.13.39 AMNI idm_errstatus_ns register.....	497
16.13.40 AMNI idm_erraddr_lsb_ns register.....	500
16.13.41 AMNI idm_erraddr_msb_ns register.....	500



16.13.42 AMNI idm_errmisc0_ns register.....	501
16.13.43 AMNI idm_errmisc1_ns register.....	502
16.13.44 AMNI idm_access_status_ns register.....	503
16.13.45 AMNI idm_access_readid_ns register.....	505
16.13.46 AMNI idm_access_writeid_ns register.....	506
16.13.47 AMNI idm_reset_status_ns register.....	507
16.13.48 AMNI idm_reset_readid_ns register.....	508
16.13.49 AMNI idm_reset_writeid_ns register.....	509
16.13.50 AMNI idm_interrupt_status_ns register.....	510
16.13.51 AMNI idm_interrupt_mask_ns register.....	511
16.14 HSNi register summary.....	512
16.14.1 HSNi node_type register.....	514
16.14.2 HSNi node_info register.....	515
16.14.3 HSNi secure_access register.....	518
16.14.4 HSNi pmusela register.....	519
16.14.5 HSNi pmuselb register.....	520
16.14.6 HSNi interface_id_0_3 register.....	521
16.14.7 HSNi num_sub_features register.....	522
16.14.8 HSNi sub_feature_0_type register.....	523
16.14.9 HSNi sub_feature_0_pointer register.....	524
16.14.10 HSNi node_control register.....	525
16.14.11 HSNi address_remap register.....	527
16.14.12 HSNi hang_detector_ctrl register.....	528
16.14.13 HSNi silicon_debug register.....	532
16.14.14 HSNi qosctl register.....	533
16.14.15 HSNi wdatthrs register.....	534
16.14.16 HSNi qos_values register.....	535
16.14.17 HSNi qosot register.....	536
16.14.18 HSNi qoscompk register.....	537
16.14.19 HSNi qoscombur register.....	538
16.14.20 HSNi qoscomavg register.....	539
16.14.21 HSNi qoscombqv register.....	540
16.14.22 HSNi mpam_control register.....	541
16.14.23 HSNi interrupt_status register.....	542
16.14.24 HSNi interrupt_mask register.....	543
16.14.25 HSNi interrupt_status_ns register.....	544

16.14.26 HSNi interrupt_mask_ns register.....	545
16.14.27 HSNi idm_device_id register.....	546
16.14.28 HSNi idm_config register.....	547
16.14.29 HSNi idm_errctlr register.....	548
16.14.30 HSNi idm_errstatus register.....	550
16.14.31 HSNi idm_erraddr_lsb register.....	552
16.14.32 HSNi idm_erraddr_msb register.....	553
16.14.33 HSNi idm_errmisc0 register.....	554
16.14.34 HSNi idm_errmisc1 register.....	555
16.14.35 HSNi idm_access_control register.....	556
16.14.36 HSNi idm_access_status register.....	557
16.14.37 HSNi idm_access_readid register.....	559
16.14.38 HSNi idm_access_writeid register.....	560
16.14.39 HSNi idm_reset_control register.....	561
16.14.40 HSNi idm_reset_status register.....	563
16.14.41 HSNi idm_reset_readid register.....	565
16.14.42 HSNi idm_reset_writeid register.....	566
16.14.43 HSNi idm_timeout_control register.....	567
16.14.44 HSNi idm_timeout_value register.....	568
16.14.45 HSNi idm_interrupt_status register.....	569
16.14.46 HSNi idm_interrupt_mask register.....	570
16.14.47 HSNi idm_errstatus_ns register.....	571
16.14.48 HSNi idm_erraddr_lsb_ns register.....	574
16.14.49 HSNi idm_erraddr_msb_ns register.....	574
16.14.50 HSNi idm_errmisc0_ns register.....	575
16.14.51 HSNi idm_errmisc1_ns register.....	576
16.14.52 HSNi idm_access_status_ns register.....	577
16.14.53 HSNi idm_access_readid_ns register.....	579
16.14.54 HSNi idm_access_writeid_ns register.....	580
16.14.55 HSNi idm_reset_status_ns register.....	581
16.14.56 HSNi idm_reset_readid_ns register.....	582
16.14.57 HSNi idm_reset_writeid_ns register.....	583
16.14.58 HSNi idm_interrupt_status_ns register.....	584
16.14.59 HSNi idm_interrupt_mask_ns register.....	585
16.15 HMNI register summary.....	586
16.15.1 HMNI node_type register.....	588

16.15.2 HMNI node_info register.....	589
16.15.3 HMNI node_features register.....	591
16.15.4 HMNI secure_access register.....	592
16.15.5 HMNI node_control register.....	593
16.15.6 HMNI pmusela register.....	594
16.15.7 HMNI pmuselb register.....	595
16.15.8 HMNI interface_id_0_3 register.....	596
16.15.9 HMNI num_sub_features register.....	597
16.15.10 HMNI sub_feature_0_type register.....	598
16.15.11 HMNI sub_feature_0_pointer register.....	599
16.15.12 HMNI silicon_debug register.....	600
16.15.13 HMNI interrupt_status register.....	601
16.15.14 HMNI interrupt_mask register.....	602
16.15.15 HMNI interrupt_status_ns register.....	603
16.15.16 HMNI interrupt_mask_ns register.....	604
16.15.17 HMNI idm_device_id register.....	605
16.15.18 HMNI idm_config register.....	606
16.15.19 HMNI idm_errctlr register.....	607
16.15.20 HMNI idm_errstatus register.....	609
16.15.21 HMNI idm_erraddr_lsb register.....	611
16.15.22 HMNI idm_erraddr_msb register.....	612
16.15.23 HMNI idm_errmisc0 register.....	613
16.15.24 HMNI idm_errmisc1 register.....	614
16.15.25 HMNI idm_access_control register.....	615
16.15.26 HMNI idm_access_status register.....	616
16.15.27 HMNI idm_access_readid register.....	618
16.15.28 HMNI idm_access_writeid register.....	619
16.15.29 HMNI idm_reset_control register.....	620
16.15.30 HMNI idm_reset_status register.....	622
16.15.31 HMNI idm_reset_readid register.....	624
16.15.32 HMNI idm_reset_writeid register.....	625
16.15.33 HMNI idm_timeout_control register.....	626
16.15.34 HMNI idm_timeout_value register.....	627
16.15.35 HMNI idm_interrupt_status register.....	628
16.15.36 HMNI idm_interrupt_mask register.....	629
16.15.37 HMNI idm_errstatus_ns register.....	630

16.15.38 HMNI idm_erraddr_lsb_ns register.....	633
16.15.39 HMNI idm_erraddr_msb_ns register.....	633
16.15.40 HMNI idm_errmisc0_ns register.....	634
16.15.41 HMNI idm_errmisc1_ns register.....	635
16.15.42 HMNI idm_access_status_ns register.....	636
16.15.43 HMNI idm_access_readid_ns register.....	638
16.15.44 HMNI idm_access_writeid_ns register.....	639
16.15.45 HMNI idm_reset_status_ns register.....	640
16.15.46 HMNI idm_reset_readid_ns register.....	641
16.15.47 HMNI idm_reset_writeid_ns register.....	642
16.15.48 HMNI idm_interrupt_status_ns register.....	643
16.15.49 HMNI idm_interrupt_mask_ns register.....	644
16.16 PMNI register summary.....	645
16.16.1 PMNI node_type register.....	647
16.16.2 PMNI node_info register.....	648
16.16.3 PMNI secure_access register.....	649
16.16.4 PMNI pmusela register.....	651
16.16.5 PMNI pmuselb register.....	652
16.16.6 PMNI interface_id_0_3 register.....	653
16.16.7 PMNI interface_id_4_7 register.....	654
16.16.8 PMNI interface_id_8_11 register.....	655
16.16.9 PMNI interface_id_12_15 register.....	656
16.16.10 PMNI num_sub_features register.....	657
16.16.11 PMNI sub_feature_0_type register.....	657
16.16.12 PMNI sub_feature_0_pointer register.....	658
16.16.13 PMNI secure_info register.....	659
16.16.14 PMNI node_features register.....	661
16.16.15 PMNI node_control register.....	663
16.16.16 PMNI silicon_debug register.....	665
16.16.17 PMNI idm_device_id register.....	666
16.16.18 PMNI idm_config register.....	667
16.16.19 PMNI idm_errctlr register.....	668
16.16.20 PMNI idm_errstatus register.....	670
16.16.21 PMNI idm_erraddr_lsb register.....	672
16.16.22 PMNI idm_erraddr_msb register.....	673
16.16.23 PMNI idm_errmisc0 register.....	674

16.16.24 PMNI idm_errmisc1 register.....	675
16.16.25 PMNI idm_access_control register.....	676
16.16.26 PMNI idm_access_status register.....	677
16.16.27 PMNI idm_access_readid register.....	679
16.16.28 PMNI idm_access_writeid register.....	680
16.16.29 PMNI idm_reset_control register.....	681
16.16.30 PMNI idm_reset_status register.....	683
16.16.31 PMNI idm_reset_readid register.....	685
16.16.32 PMNI idm_reset_writeid register.....	686
16.16.33 PMNI idm_timeout_control register.....	687
16.16.34 PMNI idm_timeout_value register.....	688
16.16.35 PMNI idm_interrupt_status register.....	689
16.16.36 PMNI idm_interrupt_mask register.....	690
16.16.37 PMNI idm_errstatus_ns register.....	691
16.16.38 PMNI idm_erraddr_lsb_ns register.....	694
16.16.39 PMNI idm_erraddr_msb_ns register.....	694
16.16.40 PMNI idm_errmisc0_ns register.....	695
16.16.41 PMNI idm_errmisc1_ns register.....	696
16.16.42 PMNI idm_access_status_ns register.....	697
16.16.43 PMNI idm_access_readid_ns register.....	699
16.16.44 PMNI idm_access_writeid_ns register.....	700
16.16.45 PMNI idm_reset_status_ns register.....	701
16.16.46 PMNI idm_reset_readid_ns register.....	702
16.16.47 PMNI idm_reset_writeid_ns register.....	703
16.16.48 PMNI idm_interrupt_status_ns register.....	704
16.16.49 PMNI idm_interrupt_mask_ns register.....	705

## **A. Signal descriptions.....707**

A.1 Signal timing constraints and clock associations.....	707
A.2 ASNI external interface types and associated signal groups.....	709
A.2.1 ASNI AXI4 write address channel signals.....	710
A.2.2 ASNI AXI5 extension write address channel signals.....	711
A.2.3 ASNI ACE-Lite write address channel signals.....	712
A.2.4 ASNI ACE5-Lite extension write address channel signals.....	712
A.2.5 ASNI AXI4 write data channel signals.....	713
A.2.6 ASNI AXI5 extension write data channel signals.....	714

A.2.7 ASNI AXI4 write response channel signals.....	715
A.2.8 ASNI AXI5 extension write response channel signals.....	716
A.2.9 ASNI ACE5-Lite extension write response channel signals.....	716
A.2.10 ASNI AXI4 read address channel signals.....	717
A.2.11 ASNI AXI5 extension read address channel signals.....	718
A.2.12 ASNI ACE-Lite read address channel signals.....	719
A.2.13 ASNI AXI4 read data channel signals.....	719
A.2.14 ASNI AXI5 extension read data channel signals.....	720
A.2.15 Other ASNI signals.....	721
A.2.16 ASNI Cortex-R52 and Cortex-R52+ AXIM interface signals.....	722
A.2.17 ASNI LLPP interface signals.....	727
A.2.18 ASNI Flash interface signals.....	731
A.3 AMNI external interface types and associated signal groups.....	732
A.3.1 AMNI AXI4 write address channel signals.....	734
A.3.2 AMNI AXI5 extension write address channel signals.....	735
A.3.3 AMNI ACE-Lite write address channel signals.....	736
A.3.4 AMNI ACE5-Lite extension write address channel signals.....	736
A.3.5 AMNI AXI4 write data channel signals.....	737
A.3.6 AMNI AXI5 extension write data channel signals.....	738
A.3.7 AMNI AXI4 write response channel signals.....	739
A.3.8 AMNI AXI5 extension write response channel signals.....	739
A.3.9 AMNI AXI4 read address channel signals.....	740
A.3.10 AMNI AXI5 extension read address channel signals.....	742
A.3.11 AMNI ACE-Lite read address channel signals.....	743
A.3.12 AMNI AXI4 read data channel signals.....	743
A.3.13 AMNI AXI5 extension read data channel signals.....	744
A.3.14 AMNI AXI3 interface configuration signal changes.....	745
A.3.15 AXIS interface signals.....	746
A.4 HSNi external interface types and associated signal groups.....	750
A.4.1 HSNi AHB-Lite request signals.....	751
A.4.2 HSNi AHB5 extension request signals.....	752
A.4.3 HSNi AHB-Lite response signals.....	752
A.4.4 HSNi AHB5 extension response signals.....	753
A.4.5 Other HSNi AHB signals.....	753
A.5 HMNI external interface types and associated signal groups.....	753
A.5.1 HMNI AHB-Lite request signals.....	754

A.5.2 HMNI AHB5 extension request signals.....	755
A.5.3 HMNI AHB-Lite response signals.....	755
A.5.4 HMNI AHB5 extension response signals.....	756
A.5.5 Other HMNI AHB signals.....	756
A.6 PMNI external interface types and associated signal groups.....	757
A.6.1 PMNI APB signals.....	758
A.6.2 PMNI APB3 signals.....	758
A.6.3 PMNI APB4 signals.....	759
A.7 Miscellaneous AXI interface signals.....	760
A.8 Clock and reset signals.....	760
A.9 Clock management signals.....	761
A.10 Power management signals.....	761
A.11 IDM interface signals.....	762
A.12 Interrupt signals.....	762
A.13 Configuration strap signals.....	763
A.14 DFT interface signals.....	764
A.15 Debug and Performance Monitoring Unit interface signals.....	764
A.16 Fault Management Unit interface signals.....	765
A.17 Access Protection Unit interface signals.....	766
<b>B. Revisions.....</b>	<b>767</b>

# 1. Introduction

## 1.1 Product revision status

The  $r_xp_y$  identifier indicates the revision status of the product described in this manual, for example,  $r1p2$ , where:

<b><math>r_x</math></b>	Identifies the major revision of the product, for example, $r1$ .
<b><math>p_y</math></b>	Identifies the minor revision or modification status of the product, for example, $p2$ .

## 1.2 Intended audience

This book is written for system designers, system integrators, and programmers who are designing or programming a System on Chip (SoC) that uses the Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect.

## 1.3 Conventions

The following subsections describe conventions used in Arm documents.

### Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

### Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
<b>bold</b>	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.



Convention	Use
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example:  <pre>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</pre>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .



Recommendations. Not following these recommendations might lead to system failure or damage.



Requirements for the system. Not following these requirements might result in system failure or damage.



Requirements for the system. Not following these requirements will result in system failure or damage.



An important piece of information that needs your attention.



A useful tip that might make it easier, better or faster to perform a task.



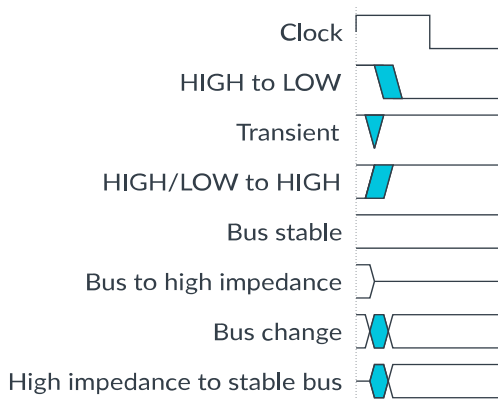
A reminder of something important that relates to the information you are reading.

## Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

**Figure 1-1: Key to timing diagram conventions**



## Signals

The signal conventions are:

### Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

### Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

## 1.4 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at [developer.arm.com/documentation](https://developer.arm.com/documentation). Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Configuration and Integration Manual	102757	Confidential
Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Development Interface Report	108037	Confidential
Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Release Note	107960	Confidential
Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Safety Manual	102758	Confidential

Arm architecture and specifications	Document ID	Confidentiality
<a href="#">AMBA® AHB Protocol Specification</a>	IHI 0033C	Non-Confidential
<a href="#">AMBA® APB Protocol Specification</a>	IHI 0024E	Non-Confidential
<a href="#">AMBA® AXI and ACE Protocol Specification</a>	IHI 0022H.c	Non-Confidential
<a href="#">AMBA® Low Power Interface Specification</a>	IHI 0068D	Non-Confidential
<a href="#">Arm® Architecture Reference Manual for A-profile architecture</a>	DDI 0487J.a	Non-Confidential
<a href="#">Principles of Arm® Memory Maps White Paper</a>	DEN 0001C	Non-Confidential

Non-Arm resources	Document ID	Organization
<a href="#">JEDEC Standard Manufacturer's Identification Code, JEP106</a>	JEP106	<a href="http://www.jedec.org">www.jedec.org</a>



Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at <http://www.adobe.com>

## 2. CoreLink NI-710AE Network-on-Chip Interconnect

Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect is a highly configurable AMBA®-compliant system-level interconnect that enables functional safety for automotive and industrial applications. With NI-710AE, you can create a non-coherent interconnect that is optimized to the Power, Performance, and Area (PPA) requirements of your SoC design.

Designed to scale, NI-710AE is suitable for large designs as a backplane interconnect. Using multiple routers and various topology options, you can connect multiple upstream and downstream devices that use different AMBA protocols to NI-710AE.

The features of NI-710AE provide flexibility and enable the interconnect to adapt to a wide range of system requirements. For more information, see [Key features](#).

NI-710AE provides safety features for critical components. For more information on the safety features, see [Key features](#).

NI-710AE supports various AMBA protocols and complies with the relevant specifications. For more information, see [Compliance](#).

Separate NI-710AE interfaces receive inputs and send outputs over the different supported protocols. For more information, see [Interfaces](#).

The architecture of NI-710AE is designed for high frequency with low latency while also optimizing system bandwidth and PPA. For more information, see [Architecture overview](#).

An NI-710AE implementation comprises a network of functional units that process and route traffic. For more information, see [Functional units](#).

Both the overall topology and individual functional units in NI-710AE can be configured according to the system requirements. For more information, see [Configurable options](#).

To optimize the performance of an NI-710AE implementation, Arm® recommends that the steps in the product design flow are completed in order. For more information, see [Product design flow](#).

NI-710AE includes documentation that provides detailed information to support each stage of the product design flow. For more information, see [Product documentation](#).

### 2.1 Key features

NI-710AE supports various features to enable you to use the interconnect in an SoC.

NI-710AE includes the following key features:

- Functional safety features:

- Support for Dual Lock Step (DLS) protection. For more information, see [Logic protection](#).
- AMBA interface protection using parity. For more information, see [AMBA interfaces](#).
- Protection for all non-AMBA interface signals. For more information, see [Miscellaneous external interface protection](#).
- Support for Arm® Cortex®-R52 and Arm® Cortex®-R52+ interface protection scheme using Error Correcting Code (ECC).
- Support for protecting address regions using the Access Protection Unit (APU). For more information, see [Memory access protection and the Access Protection Unit](#).
- Support for propagation of a source identifier (APUID) from ingress to egress port to allow the implementation of custom firewalls near to the peripheral. For more information, see [Memory access protection and the Access Protection Unit](#).
- Support for Cyclic Redundancy Check (CRC) protection on the internal network. For more information, see [Internal network protection](#).
- Support for hang detection at the ingress endpoint. For more information, see [Hang detection](#).
- Support for a central Fault Management Unit (FMU). For more information, see [Fault Management Unit](#).
- Native support for the following AMBA protocols:
  - AXI5, AXI-G, and AXI-H
  - AHB5
  - APB3, APB4, and APB5
  - AXI3 on NI-710AE AMNIs only.

For more information, see [Compliance](#).

- Packet transfer over multiple clock, power, and voltage domains. For more information, see [Power, clock, and reset management](#).
- Source-based packet routing. For more information, see [Functional units](#).
- Worm-hole routing with support for multiple Resource Planes (RPs). For more information, see [Resource Planes](#).
- Flit-level credit-based flow control. For more information, see [Functional units](#).
- Quality of Service (QoS) features for prioritization of information transfer. For more information, see [Quality of Service](#).
- Distributed switching mechanism to enable traffic management and protect against network saturation. For more information, see [Functional units](#).
- Variable, configurable topology that is specified through Arm® Socrates™. For more information, see [Configurable options](#).
- Support for transporting data parity, ECC, or poison information through the interconnect. For more information, see [Support for transporting data parity, ECC, and poison information](#).
- Support for scan cell insertion as part of the Design for Test strategy. For more information, see [Test features](#).

## 2.1.1 Test features

NI-710AE supports a scan cell insertion methodology for your SoC Design for Test (DFT) strategy. DFT control signals allow you to achieve a very high coverage for your NI-710AE test strategy.

The DFT control signals provide the following capabilities:

- Disabling internal resets.
- Controlling architectural clock gating.
- Clock disable pin. Use the DFT<CLKNAME>DISABLE inputs to disable specific clock regions to reduce power consumption during testing.

For more information about the test features of NI-710AE, see the NI-710AE Configuration and Integration Manual.

## 2.2 Compliance

NI-710AE complies with various AMBA specifications and standards.

This product is compliant with:

- [AMBA® AXI and ACE Protocol Specification](#)

NI-710AE does not support AXI4-Lite.

- [AMBA® AHB Protocol Specification](#)
- [AMBA® APB Protocol Specification](#)
- [AMBA® Low Power Interface Specification](#)

This manual must be read with the AMBA specifications. Information from these specifications is not reproduced in this document. For more information, see [1.4 Useful resources](#) on page 26.

### 2.2.1 Supported AMBA protocol features

NI-710AE supports the AMBA AXI5, ACE5-Lite, AHB5, APB3, APB4, and APB5 protocols.

The following specific AXI5 capabilities are supported:

#### AXI5

- Atomic transactions:

Transactions that perform more than just a single access and have an associated operation.

- Additional QoS Accept signals:

Two extra signals that enable a completer to indicate the minimum QoS value of transactions that it accepts.

- Trace signals:

Signals that can be associated with each channel to support the debugging, tracing, and performance measurement of systems.

- Loopback signals:

Signals that permit an agent that is issuing transactions to store information that is related to the transaction in an indexed table.

- Wake up signals:

Signals that are used to indicate that there is activity that is associated with the interface.

- Non-secure Access Identifiers:

IDs that control access to particular Non-secure memory locations.



All OPTIONAL AXI5 capabilities, except for wake up signaling, can be disabled when NI-710AE is integrated with an AXI4-based system.

---

## ACE5-Lite

- Cache stashing transactions:

Transactions that enable one component to indicate that a particular cache line must be placed in the cache of another component in the system.

- Deallocating transactions:

Transactions that are primarily used to deallocate cache lines when they are no longer required.

- Persistent Cache Maintenance Operations (CMOs):

Operations that are used to ensure that a store operation, potentially held in a Dirty cache line, is moved downstream to persistent memory.

## AXI5-H

- Memory Tagging Extensions (MTE):

A feature that provides a mechanism that can be used to detect memory safety violations.

- Prefetch request:

Requests that enable a requester to signal to the system to prepare a location for reading before making an actual read request.

- Data writes combined with CMOs:

Operations that allow CMOs to be used in conjunction with a write to memory for improved efficiency.

- Check type:

The Odd\_Parity\_Byte\_All scheme protects the AMBA interfaces.

## AXI5-F

- Memory System Resource Partitioning and Monitoring (MPAM):

A technology for partitioning and monitoring memory system resources for physical and virtual machines.

- Unique ID indicator:

A flag that shows when a request is using an AXI identifier that is unique for in-flight transactions.

- Read data chunking:

A feature that enables a completer interface to send read data for a transaction in any order using a 128-bit granule.

- CMOs on the write channel:

Operations that consist of a request on the AW channel and a response on the B channel.

- Read interleaving property:

A property that indicates whether an interface supports the interleaving of read data beats from different transactions.



ASNI cannot guarantee that data beats between transactions with different IDs do not interleave. Therefore, the `Read_Interleaving_Disabled` property is always False for ASNI.

---

## AXI3

You can configure NI-710AE AMNIs with an AXI3 requester interface that connects to completer devices. For supported AXI3 features, see [Configurable options](#).

For more information on the AXI and ACE protocols, see the [AMBA® AXI and ACE Protocol Specification](#).

NI-710AE supports the following AMBA interfaces:

- AXI5. For supported AXI5 features, see [Configurable options](#).
- AHB5. The AHB5 specification adds a set of OPTIONAL capabilities to AHB-Lite.



To connect an AHB-Lite requester or completer to NI-710AE, you must disable the OPTIONAL capabilities on the HMNI or HSNL.

- ACE5-Lite
- APB3 and APB4
- AXI3 on NI-710AE AMNIs only

## 2.2.2 Unsupported AMBA protocol features

NI-710AE does not support all AMBA protocol features.

The following AMBA protocol features are not supported:

### AXI

- AXI region identifiers (AxREGION signaling) are not supported.
- Barrier transactions (AxBAR signaling) are not supported.

### AXI3

- In NI-710AE AXI3 support is only available on AMNIs. NI-710AE ASNIs do not support AXI3.
- Write data interleaving is not supported.
- Locked accesses are not supported. NI-710AE supports normal and exclusive accesses only.
- Write data dependencies.
  - From AXI4 onwards, the AXI protocol added an extra dependency for write transactions. For NI-710AE, an AXI3 completer that accepts all write data and provides a write response before accepting the address is not compliant with AXI4 or later. The AXI specification strongly recommends that any new AXI3 completer implementation includes this additional dependency.
  - The NI-710AE AMNI conforms to the AXI4 dependency requirement. If a write response is received before the write address phase is accepted, the AMNI behavior is **UNPREDICTABLE**. Downstream AXI3 completers must conform to the AXI4 requirement to integrate directly with NI-710AE.
  - An external wrapper is required to integrate a downstream AXI3 completer. The external wrapper ensures that a returning write response is not provided until the completer has accepted the appropriate address.

### AHB

- Locked transfers are not supported. HMASTLOCK indication is ignored at the HSNL.
- If early write response is enabled, multi-copy atomicity is not supported. However, if early write response is disabled, multi-copy atomicity is supported.
- Multiple completer select (HSELx) signaling is not supported.
- Split and retry are not supported.

## APB5

- There is no support for WAKEUP signaling.
- USER\*\_WIDTH=0. There is no support for USER\*\_WIDTH.

## 2.2.3 TrustZone technology and security

NI-710AE supports the Secure and Non-secure capabilities that are provided by Arm® TrustZone® technology.

The AXI AxPROT signal conveys a Secure or Non-secure attribute for each individual request. This attribute is passed from the requester device through NI-710AE to the downstream device. The completer device determines the appropriate action from the security access permission of the request.

For accesses to NI-710AE internal configuration registers and performance monitoring counters, the security attribute determines the appropriate action. For example, Non-secure accesses to Secure configuration registers are not permitted to read or update the register. If there is a mismatch, reads return zero data and writes are ignored. However, the transaction completes in a protocol-compliant fashion, without indicating any error on the response.

TrustZone technology security checks only cover the configured network, so it is important to consider other potential attack vectors. For more information, see [TrustZone technology scope](#).

### 2.2.3.1 TrustZone technology scope

The security checks that TrustZone technology implements cover the scope of a configured network.

However, a system architect needs to mitigate the following potential risks because TrustZone is not running:

#### Physical attack

Physical attack on the device.

#### System implementation information

If you do not consider all the upstream devices that have access to the programmer's view, security vulnerabilities can occur. For example:

- If a Non-secure state upstream device can set QoS requirements that affect its Non-secure transactions, then that Non-secure state device can use this capability. Traffic analysis determines the QoS and priority settings of a Secure upstream device. This feature can be a threat in particular implementations.
- A TrustZone technology-aware downstream device requires that you set the connecting network as Non-secure. The network then does not filter the traffic and leaves the downstream device to determine the correct response. Consider the security of the upstream device that can make this Non-secure configuration and the upstream device, or devices, that can program the TrustZone technology-aware downstream device.

### Topology issues

It might be possible to suffer timing attacks because of the topology configuration you choose. For example, if two cascaded switches exist with a shared AXI link between them, then continuous Non-secure accesses to a Non-secure completer might block Secure transactions to a different Secure completer.

### Resets

It might be possible to carry out a Secure attack by resetting only parts of a data path. The data path might be a section in an individual clock domain within a network, or within a device.

### Hierarchical clock gating

It might be possible to carry out a denial-of-service attack by gating clock domains. Only upstream devices in the Secure domain must access the clock controller.

## 2.3 Product design flow

Before using NI-710AE, several processes must be performed. To obtain the best performance, we recommend that some of the implementation stages are carried out before integrating NI-710AE into the wider SoC.

The product design flow comprises the following processes:

#### Implementation

The implementer configures and synthesizes the Register Transfer Level (RTL).

#### Integration

The integrator connects the implemented design into an SoC, including establishing connections to:

- Memory system
- Processors
- Peripherals

#### Final SoC implementation

The final, fully-integrated SoC is implemented in silicon.

Arm can only provide guidance on the final implementation of Arm semiconductor IP products. If Arm provides such guidance for a product, then a separate document is included in the implementation bundle for that product.

#### Programming

The system programmer develops the software that is necessary to configure and initialize NI-710AE, and tests the application software.

Each process in the product design flow:

- Is separate and a different individual or team can complete each process

- Can include implementation and integration choices that affect the behavior and features of NI-710AE, and therefore the other tasks in the flow

Various NI-710AE documents provide more information on these processes. For more information, see [Product documentation](#).

When configuring NI-710AE, Socrates provides a physically aware tooling canvas with integrated performance feedback. You can use the tool to optimize the selected path for faster timing closure.

The operation of the final device depends on:

### **Build configuration**

The implementer chooses the configuration options that affect the preprocessing of the RTL source files. These options usually include or exclude the logic that affects one or more of the features, which can be:

- Area
- Maximum frequency
- Performance of the resulting macrocell

For example, the implementer can set the number of outstanding transactions that each requester and completer interface supports.

### **Configuration inputs**

The integrator configures some features of NI-710AE by tying inputs to specific values. These configurations affect the start-up behavior before the software configuration is specified. The configurations can also limit the options that are available to the software.

### **Software configuration**

The programmer configures NI-710AE by programming values into registers. These values affect the behavior of NI-710AE, for example, by enabling QoS features.

## **2.4 Product documentation**

Each NI-710AE document is aimed at a particular audience and is associated with specific tasks in the design flow.

These documents do not reproduce information that is available in the Arm architecture and protocol specifications. For architecture and protocol information that relates to NI-710AE, see [1.4 Useful resources](#) on page 26.

The NI-710AE documentation comprises:

### **Technical Reference Manual**

The Technical Reference Manual (TRM) describes the functionality and the effects of functional options on the behavior of NI-710AE. This document is useful at all stages of the product design flow.

The choices that are made in the design flow can mean that some behaviors that the TRM describes are not relevant. If you are programming NI-710AE, then contact:

- The implementer to determine:
  - The build configuration of the implementation
  - The integration, if any, that was performed before implementing NI-710AE
- The integrator to determine the pin configuration of the device that you use

## **Configuration and Integration Manual**

The Configuration and Integration Manual (CIM) contains:

- Design-time configuration options
- Reset-time configuration options
- Available build configuration options and related considerations
- Instructions for configuring the RTL with the build configuration options
- Instructions for running test vectors
- Sign-off processes for the configured design
- Considerations when integrating NI-710AE into your system

The Arm semiconductor IP product deliverables include reference scripts and information about using these scripts to implement your design. The reference methodology flows that Arm supplies are example reference implementations only. For EDA tool support, contact your EDA tool vendor.

The CIM is a Confidential document that is only available to licensees of NI-710AE.

## **Safety Manual**

The Safety Manual (SM) describes:

- Context in which the SM was developed and how it should be used to develop supporting documentation for your product
- Lifecycle for development of the functional safety documentation for NI-710AE, including details of functional safety audits and assessments
- Fault detection and control mechanisms in NI-710AE
- RTL configuration options related to the NI-710AE fault detection and control features
- Features within NI-710AE that can be used to increase fault detection and handling capabilities
- Assumptions of Use to be used as recommendations for an SoC design
- Safety analyses and dependent failure analyses conducted for NI-710AE
- Techniques and measures used for the development and test of NI-710AE features

The SM is a Confidential document that is only available to licensees of NI-710AE.

## Development Interface Report

The Development Interface Report (DIR) describes the activities conducted by Arm that are related to the safety architecture of the NI-710AE interconnect. This document also describes additional activities that are typically required for safety-related designs, but which have not been conducted by Arm.

The DIR is a Confidential document that is only available to licensees of NI-710AE.

## 2.5 Interfaces

NI-710AE has both completer and requester interfaces which support various AMBA protocols.

The terminology that we use in the NI-710AE documentation might differ from the terminology you are expecting. We use the following definitions to describe entities and interfaces relating to NI-710AE:

### Requester

An entity or device that issues requests.

### Completer

An entity or device that receives and completes requests.

### Requester interface

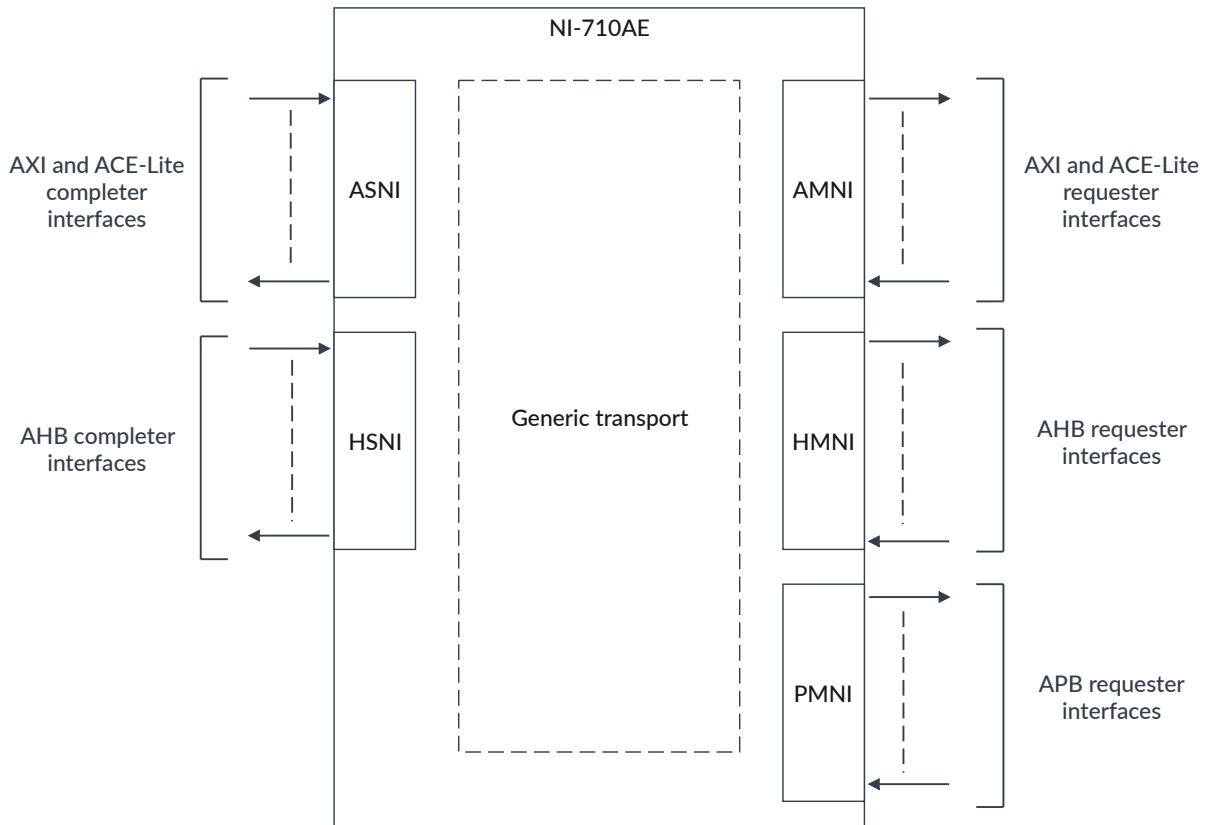
An interface that sends output to a completer device. Depending on the context, these interfaces might also be known as upstream interfaces.

### Completer interface

An interface that receives input from a requester device. Depending on the context, these interfaces might also be known as downstream interfaces.

The following figure shows how AXI requester and completer devices connect to NI-710AE through completer interfaces and requester interfaces, respectively.

**Figure 2-1: NI-710AE top-level interfaces**



NI-710AE has Low-Power Interfaces (LPIs) to control the clock and power functions. These interfaces are not shown in the preceding diagram.

## 2.6 Architecture overview

The architecture of NI-710AE is designed to provide a high frequency, low latency interconnect.

Except for HSNI requests, all NI-710AE endpoints and transport components have a minimum latency of one cycle for each block. HSNI requests have a minimum latency of two cycles. An NI-710AE interconnect with a configured link width of 512 bits operating at a frequency of 1GHz provides 64GB/s of raw bandwidth.

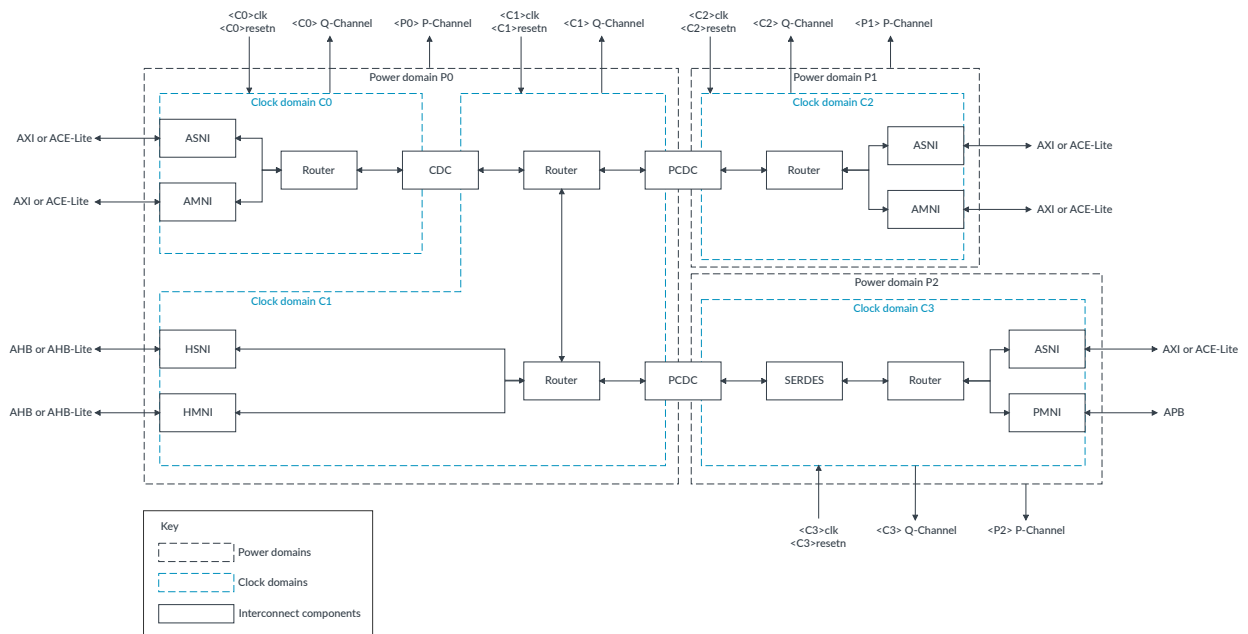
To optimize system bandwidth and PPA, NI-710AE provides the following architectural features:

- Multiple requesters and completers with a combination of the following protocols:
  - AXI5
  - ACE5-Lite
  - ACE5-LiteACP
  - AHB5

- APB5
- APB4
- APB3
- AXI3 protocol on AMNIs only
- Packetizing mechanism that enables configurable link widths from 32–2048 bits
- Independent widths for the defined sideband signals for each channel
- Resource Planes (RPs) to permit traffic isolation
- Management of shared resources so RPs do not obstruct each other
- Configurable duplicate links between pairs of router units
- Bandwidth regulators for improved QoS
- Support for address striping to load balance between memory resources
- Highly flexible timing closure options
- Support for multiple clock domains and hierarchical clock gating
- Support for multiple power domains and power gating

The following figure shows an example of the NI-710AE top-level architecture, with defined inputs and outputs.

**Figure 2-2: Top-level architecture example for NI-710AE**



The NI-710AE Power and Clock Domain Crossing (PCDC) unit is responsible for bridging between power domains and clock domains. You can configure the PCDC unit to provide only clock domain crossings or both power and clock domain crossings.



Routers direct traffic through the interconnect. The SERDES units resize flits so that they move between interconnect regions with different flit widths.

## 2.7 Functional units

NI-710AE is constructed from various functional units. Each functional unit has its own transfer function.

There are two broad categories of functional units in NI-710AE: endpoints and network components. You connect these functional units connect together to create an interconnect topology.

### Endpoints

NI-710AE contains endpoint components, also known as network interfaces, which connect to requester and completer devices in your system. These components form bridges between the interconnect and system devices. The endpoints are categorized according to whether they connect to requester devices or completer devices and the AMBA protocol the endpoint supports. These categories are reflected by the endpoint naming convention. The following table summarizes the basic connections of each endpoint type.

**Table 2-1: NI-710AE endpoint types and connections**

Endpoint	External interface type	Connects to
ASNI	AXI or ACE-Lite completer interface	AXI or ACE-Lite requester device
HSNI	AHB or AHB-Lite completer interface	AHB or AHB-Lite requester device
AMNI	AXI or ACE-Lite requester interface	AXI or ACE-Lite completer device
HMNI	AHB or AHB-Lite requester interface	AHB or AHB-Lite completer device
PMNI	APB requester interface	APB completer device

ASNIs and HSNIs are collectively referred to as xSNIs. xSNIs send requests and write data from the attached requester device into the interconnect. They also return responses and read data that are sent through the interconnect to the requester device.

AMNIs, HMNIs, and PMNIs are collectively referred to as xMNIs. xMNIs transfer requests and write data from the interconnect to the attached completer device. They also route responses and read data from the downstream completer device into the interconnect.

### Network components

The NI-710AE network components perform various functions throughout the interconnect.

#### Power and Clock Domain Crossing (PCDC) unit

PCDCs form bridges between different power domains, clock domains, or both power and clock domains in the interconnect. For more information, see [Power and Clock Domain Crossing unit](#).

#### Router

Routers channel Generic Transport (GT) flits through the network layer of the interconnect. For more information, see [Router](#).

### Interconnect link upsizing and downsizing (SERDES) unit

SERDES units resize GT flits in the network layer of the interconnect. For more information, see [SERDES unit](#).

### Performance Monitoring Unit (PMU)

The PMU counts performance events generated by the other interconnect functional units. For more information, see [Performance Monitoring Unit](#).

### Building an interconnect structure

You use Socrates to create and configure network topologies from the functional units. All functional units have the following configurable options:

- Number of credits available for each channel
- Flit width for each channel

Together, the functional units process and route network traffic across the NI-710AE network layer by performing the following tasks:

- Converting between AXI, AHB, or APB transactions and NI-710AE GT protocol flits
- Routing flits across the network between any completer interface and any requester interface
- Arbitrating flits according to Quality of Service (QoS) ordering and RP allocation
- Handling the passage of flits across different power and clock domains, and across areas of the network with different flit widths
- Monitoring the performance of the network

## 2.7.1 ASNI

NI-710AE ASNI components connect AXI requester devices to the NI-710AE network and process requests and responses as they pass between the requester and the network.

ASNI sit at the edge of the NI-710AE network and each ASNI connects to an AXI requester device through its AXI completer interface.

ASNI perform the following functions:

- Packetizing AXI request transactions from the upstream requester into GT request packets to send into the network
- Depacketizing GT response packets into AXI response transactions to return to the AXI requester
- Decoding transaction addresses into:
  - Target ID
  - Route vector
  - Decode Error (DECERR) indication for requests to out-of-range memory regions
  - Data width resizing indication
  - Stripe indication

- Burst splitting of incoming transactions. ASNI splits bursts if a transaction crosses a stripe boundary or if the transaction burst size is greater than the programmed ASNI burst split size.
- Reordering of read data and write response transactions through internal buffering
- Hard and soft Quality of Service (QoS) bandwidth regulation
- Timing isolation from the external requester and the network

## 2.7.2 AMNI

NI-710AE AMNI components connect AXI completer devices to the NI-710AE network and process requests and responses as they pass between the network and the completer.

AMNIs sit at the edge of the NI-710AE network and each AMNI connects to an AXI completer device through its AXI requester interface.

AMNIs perform the following functions:

- Depacketizing GT request packets into AXI request transactions to send to the AXI completer
- Packetizing AXI response transactions from the downstream completer into GT request packets to return to the requester
- Conversion between network GT requests and AXI transactions
- Routing read and write response channel traffic back to request initiators
- Burst splitting of transactions. AMNIs split bursts if the size of the original transaction is greater than the maximum burst size that the AMNI can issue.
- Data width resizing
- Memory controller bandwidth regulation through VAXQOSACCEPT
- Timing isolation from the external completer and the network

### Support for AXI3 interface types

You can configure an AMNI to have an AXI3 interface. However, there are several constraints the downstream AXI3 completer must be aware of and sometimes obey to integrate with an AMNI:

- AXI3 AMNIs have a WID pin on the interface that the downstream completer can use to connect to the WID input.
- AXI3 AMNIs observe the maximum burst length of 16 that AXI3 supports.
- AXI3 locked accesses are not supported and AMNIs do not generate locked accesses.

From AXI4 onwards, the AXI protocol adds an extra dependency for write transactions. An AXI3 completer that accepts all write data, and provides a write response before accepting the address, is not compliant with AXI4 or with later versions of the AMBA AXI protocol. The AXI specification strongly recommends that any new AXI3 completer implementation includes this additional dependency.

NI-710AE AMNIs conform to the AXI4 write data dependency requirement. Therefore, if an AMNI receives a write response before the write address phase is accepted, its behavior is

**UNPREDICTABLE.** Downstream AXI3 completers must conform to the AXI4 requirement to integrate directly with NI-710AE.

To integrate a downstream AXI3 completer that follows the AXI3 write data dependency requirement, an external wrapper is required. The external wrapper ensures a returning write response is not provided until the completer has accepted the appropriate address.

### 2.7.3 HSNi

NI-710AE HSNi components connect AHB requester devices to the NI-710AE network and process requests and responses as they pass between the requester and the network.

HSNi sit at the edge of the NI-710AE network and each HSNi connects to an AHB requester device through its AHB interface. You can configure whether the HSNi has an AHB completer interface or an AHB mirrored requester interface.

HSNi perform the following functions:

- Packetizing AHB request transactions from the upstream requester into GT request packets to send into the network
- Depacketizing GT response packets into AHB response transactions to return to the AHB requester
- Address decoding
- Hazarding. If there are any outstanding writes, a new read transaction is stopped.
- If burst promotion is enabled, conversion of AHB INCR bursts to INCR4 bursts, where possible.
- Burst splitting of incoming transactions. HSNi split bursts if a transaction crosses a stripe boundary or if the transaction burst size is larger than the programmed HSNi burst split size.
- Early write response generation and hazarding on subsequent read requests against the writes, until a write response is received from downstream
- Hard and soft QoS bandwidth regulation
- Timing isolation from the external requester and the network
- Receiving responses from xMNI that have similar or different data widths. When an HSNi receives data from a target with a smaller data width, the read data beats can arrive as fragments. The HSNi collates the fragmented responses to create a data beat of a size that corresponds to its data width.
- Receiving different error responses when combining responses for individual data fragments. In such cases, HSNi use the following priority order to create the final response that is sent to the AHB requester:
  1. DECERR or SLVERR (highest).
  2. OK.
  3. EXOKAY (lowest).

HSNi do not support the following features:

- Data widths of 512 bits and 1024 bits
- Locked transfers. The HMASTLOCK signal not supported.
- Multi-copy atomicity, unless early write response is disabled
- Multiple completer select (HSELx) signaling
- Split and retry

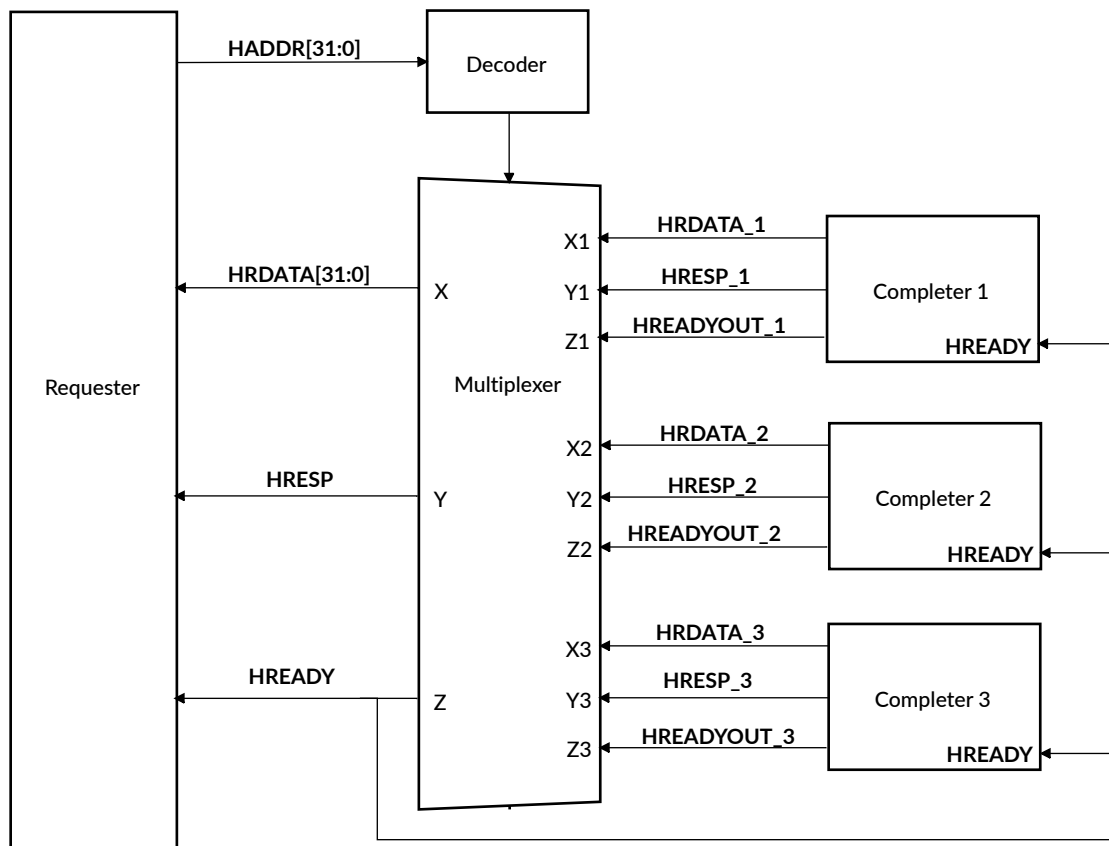


HSNIs downgrade shareable exclusive transactions to Non-shareable exclusive transactions.

## HSNI signal names

HREADYOUT is not an input to the HSNI. Rather, this signal is the HSNI port, as defined in the RTL. For clarity, the following figure in the [AMBA® AHB Protocol Specification](#) shows the HREADY and HREADYOUT signals.

**Figure 2-3: AHB protocol multiplexer interconnection scheme**



The relationship between the HREADYOUT output and the [AMBA® AHB Protocol Specification](#) is as follows:

- HREADYOUT is the equivalent of HREADY for mirrored systems. There is no HREADYIN signal to the HSNi for mirrored systems.
- HREADY (HREADYIN) and HREADYOUT are both needed in non-mirrored systems. HREADYOUT from the HSNi matches the preceding multiplexer interconnection scheme from the AHB specification. The HREADYIN input corresponds to the HREADY input port on the completers in the AHB specification multiplexer interconnection scheme.

The following table shows the HSNi system modes, signal names, and signal directions.

**Table 2-2: HSNi signals**

Mode	Signal name	Direction
Non-mirrored mode systems	HREADY.  An HREADYIN input corresponds to the HREADY input port on the completers in the <a href="#">AMBA® AHB Protocol Specification</a> .	Input to the HSNi
Non-mirrored mode systems	HREADYOUT.  HREADYOUT from an HSNi corresponds to the multiplexer interconnection scheme in the <a href="#">AMBA® AHB Protocol Specification</a> .	Output from the HSNi
Mirrored mode systems	HREADY.  HREADYHREADYOUT in the RTL corresponds to HREADY in the <a href="#">AMBA® AHB Protocol Specification</a> for mirrored systems.  There is no HREADYIN signal to HSNIs for mirrored systems.	Output to the HSNi

## HMNI signal names

For the HMNI, HREADYOUT is always an input. HMNI mirrored mode contains two extra outputs, HREADY and HSEL. The following table shows the HMNI system modes, signal names, and signal directions.

**Table 2-3: HMNI signals**

Mode	Signal name	Direction
Mirrored mode systems	HREADY	Output from the HMNI
	HSEL	Output from the HMNI
	HREADYOUT	Always an input to the HMNI
AHB requester interface (non-mirrored mode systems)	HREADYOUT	Always an input to the HMNI

## Mirrored AHB completer interface

The AHB completer interface configures the interface with output signals HSEL, HREADYOUT, and HREADY.

### AHB requester interface

The AHB requester interface does not have HSEL or HREADY input signals. It is designed to connect directly to an AHB requester.

## 2.7.4 HMNI

NI-710AE HMNI components connect AHB completer devices to the NI-710AE network and process requests and responses as they pass between the network and the completer.

HMNI sit at the edge of the NI-710AE network and each HMNI connects to an AHB completer device through its AHB interface. You can configure whether the HMNI has an AHB requester interface or an AHB mirrored completer interface.

### AHB requester interface

This option provides all the expected AHB signals on an AHB requester, so it does not have HSEL or HREADY output signals. The input AHB ready signal is named HREADY instead of HREADYOUT.

### AHB mirrored completer interface

This option provides all the AHB signals for a completer, which includes HSEL, HREADY input, and HREADY output signals. Using this option enables the direct connection of an AHB completer to an HMNI.

HMNI perform the following functions:

- Depacketizing GT request packets into AHB request transactions to send to the AHB completer
- Packetizing AHB response transactions from the downstream completer into GT request packets to return to the requester
- Transaction address decoding into route vectors
- Timing isolation from the external requester and the network
- Non-blocking flow control of concurrent traffic by supporting multiple incoming Resource Planes (RPs)
- Burst handling of incoming WRAP and INCR bursts
- Burst conversion and splitting to handle sparse writes and unaligned accesses.

When splitting any non-modifiable burst, HMNI assert HMASTLOCK to prevent other requesters accessing the same memory location during the splitting sequence.

- Handling error responses from downstream completers

## 2.7.5 PMNI

NI-710AE PMNI components connect APB completer devices to the NI-710AE network and process requests and responses as they pass between the network and the completer.

PMNI sit at the edge of the NI-710AE network and each PMNI connects to one or more APB completer devices through its APB requester interfaces.

NI-710AE is compliant with the APB3, APB4, and APB5 protocols.

PMNIs perform the following functions:

- Depacketizing GT request packets into APB request transactions to send to APB completers
- Packetizing APB response transactions from the downstream completers into GT request packets to return to the requester
- Size conversions from GT to a fixed data width of 32 bits
- Burst splitting to split incoming bursts into multiple individual APB beats
- Handle multiplexed read and write traffic on a single channel by using low-wire mode
- Non-blocking flow control of concurrent traffic by supporting multiple incoming Resource Planes (RPs)
- Route read and write responses back to initiators by using an address decoder
- Support up to 16 APB interfaces on a single PMNI. Each interface can be individually specified to be APB3, APB4, or APB5. An internal decoder is used to generate the APB PSELx signal for selecting a specific APB requester interface.
- Process WriteNoSnoop and ReadNoSnoop opcodes only. All unsupported opcodes are processed as follows:
  - For write requests, the write data is terminated instead of forwarding the data onto the APB bus. A write response is issued with an error.
  - For read requests, all zero read data beats are forwarded and a read response is issued with an error.

## 2.7.6 Power and Clock Domain Crossing unit

NI-710AE Power and Clock Domain Crossing (PCDC) units form bridges between different clock domains, power domains, or both clock and power domains. As GT flits are transferred between domains operating at different clock speeds, PCDCs synchronize passing flits to the new clock speed.

If your design contains multiple clock domains, power domains, or clock and power domains, PCDC units are used to control power and clock domain crossing.

To permit entry and exit of flits, PCDC units have one GT input port and one GT output port.

PCDC units have Q-Channel LPs for each configured power domain, allowing for power domain control. Likewise, there is a Q-Channel LPI for each configured clock domain to enable clock domain control. These Q-Channel LPs are combined at the NI-710AE top level to provide a single Q-Channel for each clock domain and a single P-Channel for each power domain.

PCDC units perform the following functions:

- Power and clock domain crossing
- Reordering flits according to RPs. PCDC units do not alter flits as they traverse the block.
- Controlling power domain quiescence



- Controlling clock domain quiescence

## 2.7.7 Router

NI-710AE routers channel GT flits through the network layer of the interconnect.

Routers perform the following functions:

- Transporting GT flits between a configurable number of input ports and output ports according to the flit routing field.
- Routing flits according to RPs. If a router has more than one output port, it updates the flit routing field. Other than this update, routers do not alter flits as they are routed through the unit.

## 2.7.8 SERDES unit

NI-710AE SERDES units resize GT flits in the network layer of the interconnect.

SERDES units have the following connections:

- One GT input port
- One GT output port
- A threshold control input

SERDES units perform the following functions:

- Converting the width of flits
- Collating multiple sequential input flits into a single output flit when implementing the upsizing function
- Splitting a single input flit into a sequence of output flits when implementing the downsizing function
- Reordering flits according to RPs

## 2.7.9 Performance Monitoring Unit

The NI-710AE Performance Monitoring Unit (PMU) counts performance events generated by the interconnect functional units. Performance events are used to monitor various behaviors of your SoC.

The PMU is distributed across all the clock domains in NI-710AE. Within each clock domain, there are the following PMU components:

- Eight 32-bit software-visible event counters
- One 64-bit cycle counter, split across two 32-bit registers
- One programmable crossbar to select a particular event for a counter to monitor
- A control network interface for programming and read access requests from the NI-710AE configuration memory space

The functional units within a clock domain in NI-710AE, such as ASNIs, can generate performance events. Generated performance events are multiplexed onto an 8-bit event bus and routed to the event counter for that clock domain.

Each event counter has shadow snapshot registers, so that all event counters can be sampled simultaneously. The event counters also have overflow functionality.

If an event or cycle counter overflows, an interrupt is triggered. This interrupt is connected to the top-level interrupt <CLKNAME>\_nPMUINTERRUPT. You can determine the counter that has overflowed by using the PMU control and configuration registers. In addition, you can use these registers to clear any counter overflow flags so that the interrupt can be cleared.

To configure the functional crossbar within a component, use the local event programming registers. By configuring the crossbar, you indicate an event type to forward to one of the eight available clock domain counters.

For more information about the PMU, see [Performance monitoring](#).

## 2.8 Configurable options

You can customize the top-level topology and the individual functional units of NI-710AE to meet your specific design requirements.

NI-710AE provides various global and unit-based FuSa parameters for configuration. For more information, see [FuSa parameters](#).

NI-710AE provides the following microarchitecture options for configuration:

- Up to a maximum total of 255 upstream and downstream interfaces
- Up to 128 xSNIs, that is, [ASNIs](#) and [HSNIs](#)
- Up to 127 xMNIs, that is, [AMNIs](#), [HMNIs](#), and [PMNIs](#)

NI-710AE provides the following voltage, power, and clock domain options for configuration:

- Up to 32 voltage domains in total
- Up to 32 power domains in total. Each power domain is contained by a single voltage domain. In other words, a power domain cannot be divided across multiple voltage domains.
- Up to 32 clock domains in total. Each clock domain is contained by a single power domain. In other words, a clock domain cannot be divided across multiple power domains.
- Power and clock domain crossing within the network, supporting synchronous, asynchronous, and integer ratio clock domain crossings
- RTL hierarchy by voltage domain, then power domain, then clock domain
- RTL hierarchy according to a configurable grouping of components

NI-710AE provides the following address map options for configuration:

- Configurable address map for address-based routing from each upstream interface to the corresponding downstream interfaces
- Separate address map for each upstream interface
- Multiple address regions in address maps, with each region aligned and sized based on a 4KB granularity
- Address map regions can target one downstream interface or can be hashed across two or four downstream interfaces.

NI-710AE provides specific options for configuring the supported cache line size. The following table lists the cache line sizes that NI-710AE supports. No other cache line sizes are supported.

**Table 2-4: Supported cache line sizes**

Data width	Cache line size
32 bits	64 bytes
64 bits, 128 bits, 256 bits, 512 bits	64 bytes or 128 bytes
1024 bits	128 bytes

NI-710AE provides the following topology options for configuration:

- [Routers](#) with up to eight inputs and up to eight outputs for flexible topology choices
- Up to four Resource Planes (RPs) to reduce Head-of-Line (HoL) blocking
- Configurable link sizes and link crediting, with the option to resize flits within the network by using SERDES components
- [PCDC units](#) for bridging between different power and clock domains
- Option to merge read and write channels to reduce wire count and area
- Option to duplicate channels for more bandwidth

NI-710AE provides the following unit-level configuration options:

- Flexible timing closures
- Configurable transaction tracker depths
- OPTIONAL burst splitting logic. This feature can be included if a design requires transactions to be split, or excluded to save area on designs where burst splitting is not required.
- Quality of Service (QoS) regulators that can update the QoS value on a transaction according to latency targets
- OPTIONAL Interconnect Device Management (IDM) feature for configuration and management of system components by the interconnect
- Configurable FIFO sizes when crossing clock and power domains

For guidance on configuring NI-710AE for optimal performance or to meet timing, see the NI-710AE Configuration and Integration Manual.

## 2.8.1 FuSa parameters

NI-710AE contains various FuSa parameters.

The following table lists the global and unit-based FuSa parameters for NI-710AE.

**Table 2-5: NI-710AE FuSa parameters**

Category	Parameter name	Description	Global or unit based
Logic protection	dlsLogicProtection	<p>Specifies the protection for logic blocks, internal networks (GT, Config AUB and error AUB), and miscellaneous interfaces (clocks, resets, DFT*, Q-Channel, P-Channel, Async, and strap inputs).</p> <p><b>Enabled</b></p> <p>Dual Lock-Step (DLS) protects xSNI, xMNI, CFGNI, clock controller, power controller, and Performance Monitoring Unit (PMU) logic blocks. Lock-step checkers and miscellaneous interface signals are also duplicated. Internal networks are protected with packet Cyclic Redundancy Check (CRC).</p> <p>: Miscellaneous interfaces are protected with CHK signals.</p> <p><b>Disabled</b></p> <p>No logic or internal network or miscellaneous interface protection is provided. This value is the default setting.</p> <p>For more information about restrictions when this parameter is disabled, see <a href="#">FuSa parameter restrictions</a>.</p>	Global
External AMBA interface protection	ambaInterfaceProtection	<p>Specifies the AMBA interface protection of an xSNI or xMNI endpoint.</p> <p><b>AMBA_PARITY</b></p> <p>The external AMBA interface (AXI, AHB, or APB) is protected using AMBA interface parity.</p> <p><b>LEGACY_ECC</b></p> <p>The external AXI interface is protected using Cortex-R52 or Cortex-R52+ legacy ECC protection. You can only select Cortex-R52 or Cortex-R52+ legacy ECC protection for AXI interfaces.</p> <p><b>Disabled</b></p> <p>No protection is provided on the AMBA interface. This value is the default setting.</p>	Unit
Hang detection	hangdetector	<p>Specifies whether an xSNI endpoint implements a hang detector.</p> <p><b>Enabled</b></p> <p>Hang detection is implemented.</p> <p><b>Disabled</b></p> <p>Hang detection is not implemented. This value is the default setting.</p>	Unit

Category	Parameter name	Description	Global or unit based
APU protection	<code>apuProtection</code>	Specifies whether an xSNI or xMNI endpoint provides access protection. <b>Enabled</b> Endpoint performs address and APUID access check. <b>Disabled</b> Endpoint does not perform address or APUID access check. This value is the default setting.	Unit
	<code>apuAddrRegions</code>	Specifies the number of address regions inside the APU.  The configurable values are 4, 8, 16, 20 and 32. The default value is 8.	Unit
	<code>apuIdWidth</code>	Specifies the width of the ID to be matched. If the value is nonzero, then interface signal APUID is available for every endpoint.  The configurable values are 0, 4, and 8. The default value is 0.	Global
	<code>apu_region_4k</code>	Specifies the minimum address region granularity in the APU.  This parameter is only applicable when the <code>apuProtection</code> parameter is enabled for an endpoint. <b>Enabled</b> The minimum address region granularity is 4KB. This value is the default setting. <b>Disabled</b> The minimum address region granularity is 64 bytes.	Unit
IDM wire interface	<code>idmWireInterface</code>	Specifies whether the IDM wire interface ports <code>idm_softreset_req</code> and <code>idm_softreset_ack</code> are present. <b>Enabled</b> The <code>idm_softreset_req</code> and <code>idm_softreset_ack</code> ports are present on the interface. <b>Disabled</b> <code>idm_softreset_req</code> and <code>idm_softreset_ack</code> are not present on the interface.	Unit
Critical error vector	<code>criticalErrVector</code>	Configures uncorrected errors reported by safety mechanisms as <code>critical</code> or not <code>critical</code> .  The critical error vector is an 18-bit vector, with one bit corresponding to each safety mechanism.  For safety mechanisms that only report corrected errors, this parameter is ignored. <b>0b1</b> Indicates that this error should be reported as a critical uncorrected error. <b>0b0</b> Indicates that this error should be reported as not critical uncorrected. This value is the default setting.	Global

Category	Parameter name	Description	Global or unit based
Legacy ECC mode	legacyEccMode	<p>Bus protection type when LEGACY_ECC protection is selected.</p> <p>BUS_PROTECTION1</p> <p>ECC only. Supported on Cortex-R52 and Cortex-R52+ cores. This value is the default setting.</p> <p>BUS_PROTECTION3</p> <p>Data ECC only. Supported on Cortex-R52+ cores.</p>	Unit
Legacy ECC interface type	legacyEccInterfaceType	<p>Specifies the requester interface connected to an ASNI.</p> <p><b>AXIM</b></p> <p>The interface is an AXIM manager interface. This value is the default setting.</p> <p><b>LLP</b></p> <p>The interface is an LLP manager interface.</p> <p><b>FLASH</b></p> <p>The interface is a flash manager interface.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>For a mapping of the bits to safety mechanisms, see the <i>FMU FMU_SMEN register</i> section in the NI-710AE Technical Reference Manual.</li> <li>Bit 14 corresponds to a corrected error and is always reported as not critical.</li> </ul>	Unit
Legacy ECC CPU	legacyEccCpu	<p>Specifies the processor making use of the LEGACY_ECC scheme connected to NI-710AE.</p> <p>This parameter is only applicable to ASNI or AMNI when the parameter <code>ambaInterfaceProtection==LEGACY_ECC</code>.</p> <p><b>Cortex-R52</b></p> <p>The processor connected is a Cortex-R52 core. This value is the default setting.</p> <p><b>Cortex-R52+</b></p> <p>The processor connected is a Cortex-R52+ core.</p>	Unit

Category	Parameter name	Description	Global or unit based
Legacy ECC flash data scheme	legacyEccFlashDataScheme	<p>Specifies the flash memory interface data ECC chunk size.</p> <p>This parameter is only applicable when the parameter <code>ambaInterfaceProtection==LEGACY_ECC</code></p> <p>This parameter is only applicable to ASNIs when both the parameters <code>ambaInterfaceProtection==LEGACY_ECC</code> and <code>legacyECCInterfaceType==FLASH</code>.</p> <p><b>1</b></p> <p>64-bit chunks. The width of RDATACODEFx is 9 bits. This value is the default setting.</p> <p><b>2</b></p> <p>128-bit chunks. The width of RDATACODEFx is 16 bits.</p>	Unit

## FuSa parameter restrictions

The following restrictions apply to the configuration of the FuSa parameters.

### dlsLogicProtection

When the `dlsLogicProtection` parameter is disabled, the following parameters must also be disabled:

- `ambaInterfaceProtection`
- `hangdetector`
- `apuProtection`
- `idmWireInterface`

## 2.8.2 ASNI configuration options

ASNI units provide various options that you can configure to meet the specific requirements of your design, including the address and data widths.

You can configure the following options:

- Address width of 32–64 bits
- One of the following data widths:
  - 32 bits
  - 64 bits
  - 128 bits
  - 256 bits
  - 512 bits
  - 1024 bits

- User sideband signal width

For more information, see [User signals](#).

- Write acceptance capability of 1–256 transactions

Sometimes an ASNI might accept more transactions than specified in the write acceptance capability. For example, configuring a register slice at the completer interface position increases the acceptance capability.

- Read acceptance capability of 1–256 transactions

Sometimes an ASNI might accept more transactions than specified in the read acceptance capability. For example, configuring a register slice at the completer interface position increases the acceptance capability.

- Minimum atomic acceptance, which provides a guarantee of the minimum number of read tracker entries that are reserved for atomics

The minimum atomic acceptance parameter only applies if the `Atomic_Transactions` property is enabled on the ASNI. If the property is enabled, then the total read tracker size is the sum of the read acceptance and the minimum atomic acceptance.

When atomic transactions are received on the write channel, the atomic variants `AtomicLoad`, `AtomicCompare`, and `AtomicSwap` also require a read response. This process uses a tracker entry in the read tracker.

- Timing isolation:
  - From the external requester
  - From the network
- Read reorder depth of 1–255 entries

Permitted read reorder depth values are one, two, all multiples of four from 4–252 inclusive, and 255. If atomic transaction support is enabled at the ASNI, then the read reorder depth plus the minimum atomic transaction acceptance depth must be less than 256.

- Write data FIFO depth of 0–32 entries
- ID width of 1–24 bits
- Ordered Write Observation, which is an AXI property that specifies whether all agents observe write transactions with the same ID in the issued order.

Set the property to enabled, disabled, or pin. For more information about this property, see the [AMBA® AXI and ACE Protocol Specification](#).

Systems that use ordered write observation are more compatible with alternative ordering models such as the PCIe model. Therefore, we recommend using ordered write observation for all ASNIs that are connected to PCIe requesters.

- Enable IDM
- IDM device ID
- Include burst splitting logic



- Include QoS regulators:
  - The read regulator present setting enables a QoS regulator for the AR channel.
  - The write regulator present setting enables a QoS regulator for the AW channel.
  - The combined regulator present setting enables a QoS regulator that regulates traffic according to the combined bandwidth across both the AR and AW channels.
- Functional Safety (FuSa) features:
  - Dual Lock-step (DLS) protection
  - AMBA interface protection (parity or legacy ECC)
  - Miscellaneous interface protection
  - Support for Cyclic Redundancy Check (CRC) protection on the internal network
  - Hang detection
- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the Configuration and Integration Manual.

The following table shows the ASNI features that are supported for each type of interface.

**Table 2-6: Supported ASNI features by interface type**

Interface type	Parameter name	Support
AXI5 and ACE-Lite	Wakeup_Signals	Required.  Devices attached to NI-710AE must support wake up signaling.
	Check_Type	Not supported
	Poison	Not supported
	Trace_Signals	Optional
	Unique_ID_Support	Optional
	QoS_Accept	Not supported
	Loopback_Signals	Optional.  If enabled, set to 8 bits only.
	Untranslated_Transactions	Not supported
	NSAccess_Identifiers	Optional
	MPAM_Support	Optional
	Read_Interleaving_Disabled	Always set to FALSE.
	Read_Data_Chunking	Optional
	Atomic_Transactions	Optional
	MTE_Support	Optional
ACE-Lite	CMO_On_Read	Optional
	CMO_On_Write	Optional
	Persist_CMO	Optional
	Write_Plus_CMO	Optional
	Cache_Stash_Transactions	Optional

Interface type	Parameter name	Support
	DeAllocation_Transactions	Optional
	Prefetch_Transaction	Optional

### 2.8.3 AMNI configuration options

AMNI units provide various options that you can configure to meet the specific requirements of your design. For example, you can configure the number of Resource Planes (RPs) in the channels.

You can configure the following options:

- Address width of 32–64 bits
- One of the following data widths:
  - 32 bits
  - 64 bits
  - 128 bits
  - 256 bits
  - 512 bits
  - 1024 bits
- User sideband signal width

User signals are applicable to all AMNI interface types, including AXI3. For more information, see [User signals](#).

- Number of RPs present in each of the read request, write request, read response, and write response channels
- Write issuing capability of 1–256 transactions
- Read issuing capability of 1–256 transactions
- Minimum atomic issuance, which provides a guarantee of the minimum number of read tracker entries that are reserved for atomics

The minimum atomic issuance parameter only applies if the `Atomic_Transactions` property is enabled on the AMNI. If the property is enabled, then the total read tracker size is the sum of the read issuance and the minimum atomic issuance.

When atomic transactions are received on the write channel, the atomic variants `AtomicLoad`, `AtomicCompare`, and `AtomicSwap` also require a read response. This process uses a tracker entry in the read tracker.

- Timing isolation:
  - From the external completer
  - From the network
- Enable IDM
- IDM device ID

- AXI ID width of 1–32 bits

To form the outgoing AXI ID, the AMNI appends the Source ID (SrcID) of the incoming request to the least significant bits of the AXI ID. For more information on output IDs, see [Output ID calculation](#). The SrcID of the incoming request is captured in the node\_id field of the [ASNI node\\_type register](#).

- Functional Safety (FuSa) features:
  - Dual Lock-step (DLS) protection
  - AMBA interface protection (parity or legacy ECC)
  - Miscellaneous interface protection
  - Packet Cyclic Redundancy Check (CRC) protection
  - Misrouted packet protection

Set the property to enabled, disabled, or pin. For more information, see the [AMBA® AXI and ACE Protocol Specification](#).

- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the Configuration and Integration Manual.



The AxREGION signal is not supported.

You can configure AMNIs with AXI5, ACE5-Lite, ACE5-LiteACP, or AXI3 as the requester interface type.

ACE5-LiteACP has several constraints, some of which are transaction constraints and some of which are interface constraints. For more information about these constraints, see the [AMBA® AXI and ACE Protocol Specification](#).

The following table shows the AMNI features that are supported for each type of interface.

**Table 2-7: Supported AMNI features by interface type**

Interface type	Parameter name	Support
AXI5 and ACE-Lite	Wakeup_Signals	Required.  AMNIs have an output AWAKEUP signal. The downstream completer can choose to use or ignore this signal.
	Check_Type	Not supported
	Poison	Not supported
	Trace_Signals	Optional
	Unique_ID_Support	Optional
	Qos_Accept	Optional

Interface type	Parameter name	Support
ACE-Lite	Loopback_Signals	Optional. If enabled, this parameter is set to 8 bits only.
	Untranslated_Transactions	Not supported
	NSAccess_Identifiers	Optional
	MPAM_Support	Optional
	Read_Interleaving_Disabled	Optional
	Read_Data_Chunking	Optional
	Atomic_Transactions	Optional
	MTE_Support	Optional
	CMO_On_Read	Optional
	CMO_On_Write	Optional
	Persist_CMO	Optional
	Write_Plus_CMO	Optional
	Cache_Stash_Transactions	Optional
	DeAllocation_Transactions	Optional
ACE5-LiteACP	Prefetch_Transaction	Optional
	Atomic_Transactions	Not supported by the ACE5-LiteACP protocol.
	Write_Plus_CMO	Not supported by the ACE5-LiteACP protocol.
	Prefetch_Transaction	Not supported by the ACE5-LiteACP protocol.
	DeAllocation_Transactions	Not supported by the ACE5-LiteACP protocol.
	Cache_Stash_Transactions	Optional
	CMO_On_Read	Not supported by the ACE5-LiteACP protocol.
	CMO_On_Write	Not supported by the ACE5-LiteACP protocol.
	Persist_CMO	Not supported by the ACE5-LiteACP protocol.
	Trace_Signals	Optional
	NSAccess_Identifiers	Not supported by the ACE5-LiteACP protocol.
	MPAM_Support	Optional
	Unique_ID_Support	Optional
	Read_Data_Chunking	Optional
AXI3	Loopback_Signals	Not supported by the ACE5-LiteACP protocol.
	MTE_Support	Not supported by the ACE5-LiteACP protocol.
	Qos_Accept	Not supported by the ACE5-LiteACP protocol.
	Read_Interleaving_Disabled	Optional
	Untranslated_Transactions	Not supported
	Check_Type	Not supported
	Poison	Not supported
	Atomic_Transactions	Not supported
	Trace_Signals	Not supported
	NSAccess_Identifiers	Not supported
	MPAM_Support	Not supported

Interface type	Parameter name	Support
	Unique_ID_Support	Not supported
	Read_Data_Chunking	Not supported
	Loopback_Signals	Not supported
	MTE_Support	Not supported
	QoS_Accept	Not supported
	Read_Interleaving_Disabled	Not supported
	DeAllocation_Transactions	Not supported
	Cache_Stash_Transactions	Not supported
	CMO_On_Read	Not supported
	CMO_On_Write	Not supported
	Persist_CMO	Not supported
	Write_Plus_CMO	Not supported
	Prefetch_Transaction	Not supported
	Untranslated_Transactions	Not supported
	Check_Type	Not supported
	Poison	Not supported

## 2.8.4 HSNl configuration options

HSNI units provide various options that you can configure to meet the specific requirements of your design, including the read and write data widths. However, some HSNl properties are fixed in NI-710AE.

You can configure the following options:

- Interface type

HSNls support the AHB5 standard and mirrored interface types.

- One of the following read and write data widths:
  - 32 bits
  - 64 bits
  - 128 bits
  - 256 bits

The read and write data widths must be set to the same value.

- User sideband signal width

For more information, see [User signals](#).

- Write acceptance capability of 1–16 transactions

Sometimes an HSNi might accept more transactions than specified in the write acceptance capability. For example, configuring a register slice at the completer interface position increases the acceptance capability.

- HMASTER width of 1–8 bits
- Enable Extended Memory Type support
- Enable Secure transfer support
- Enable exclusive transfer support
- Enable early burst termination acceptance
- Enable burst conversion support
- Enable early write response support

When early write response is enabled, you can configure an HSNi to support 1–16 outstanding writes.

- Write data buffer FIFO depth of 0–16 data beats
- Include programmable QoS regulators
- Timing isolation:
  - From the external requester
  - From the network
- Enable IDM
- IDM device ID
- Functional Safety (FuSa) features:
  - Dual Lock-step (DLS) protection
  - AMBA interface protection (parity or legacy ECC)
  - Miscellaneous interface protection
  - Hang detection
  - Packet Cyclic Redundancy Check (CRC) protection
  - Misrouted packet protection
- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the NI-710AE Configuration and Integration Manual.

The following HSNi properties are not configurable:

- Address width

This value is fixed at 32 bits.

- Endianness

HSNis only support word-invariant little-endianness.



HSNIs do not support multiple completer select (HSELx) signaling, split and retry, or locked transfers.

The following table shows the configuration options for HSNIs.

**Table 2-8: HSNi configuration options**

Parameter name	Support
Extended_Memory_Types	Optional. Can be enabled or disabled.
Secure_Transfers	Optional. Set to pin, programmable, Secure, or Non-secure.
Endianness	BE32 only. HSNIs only support word-invariant little-endianness.
Exclusive_Transfers	Optional. Can be enabled or disabled.
Mirror_Interface	Optional. Can be enabled or disabled.
Multi_Copy_Atomicity	Optional. Set to FALSE if early write response is enabled, otherwise set to TRUE.
Stable_Between_Clock	Always set to FALSE

## 2.8.5 HMNI configuration options

HMNI units provide various options that you can configure to meet the specific requirements of your design, including whether to use a standard or mirrored interface. However, some HMNI properties are fixed in NI-710AE.

You can configure the following options:

- Interface type  
HMNI support the AHB5 standard and mirrored interface types.
- One of the following read and write data widths:
  - 32 bits
  - 64 bits
  - 128 bits
  - 256 bits

The read and write data widths must be set to the same value.

- User sideband signal width. For more information, see [User signals](#).
- Enable Extended Memory Type support
- Enable Secure transfer support
- Enable exclusive transfer support
- Timing isolation:
  - From the external completer
  - From the network
- Enable IDM
- IDM device ID
- Functional Safety (FuSa) features:
  - Dual Lock-step (DLS) protection
  - AMBA interface protection (parity or legacy ECC)
  - Miscellaneous interface protection
  - Packet Cyclic Redundancy Check (CRC) protection
  - Misrouted packet protection
- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the NI-710AE Configuration and Integration Manual.

The following HMNI properties are not configurable:

- Address width

This value is fixed at 32 bits.

- Endianness

HMNI only support word-invariant little-endianness.



HMNI do not support multiple completer select (HSELx) signaling or split and retry.

The following table shows the configuration options for HMNI.

Table 2-9: HMNI configuration options

Parameter name	Support
Extended_Memory_Types	Optional.  Can be enabled or disabled.



Parameter name	Support
Secure_Transfers	Optional.  Set to pin, register, Secure, or Non-secure.
Endianness	BE32 only.  HSNIs only support word-invariant little-endianness.
Exclusive_Transfers	Optional.  Can be enabled or disabled.
Mirror_Interface	Can be enabled or disabled
Stable_Between_Clock	Always set to FALSE

## 2.8.6 PMNI configuration options

PMNI units provide various options that you can configure to meet the specific requirements of your design, including the APB protocol to use. However, some PMNI properties are fixed in NI-710AE.

You can configure the following options:

- APB protocol type

PMNIs support the APB3, APB4, and APB5 protocols.

- Enable Secure access support

You can control Secure access support through a register or you can determine it from a pin.

- Timing isolation:
  - From the external completer
- Enable IDM
- IDM device ID
- Functional Safety (FuSa) features:
  - Dual Lock-step (DLS) protection
  - AMBA interface protection (parity or legacy ECC)
  - Miscellaneous interface protection
  - Packet Cyclic Redundancy Check (CRC) protection
  - Misrouted packet protection
- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the NI-710AE Configuration and Integration Manual.

The following PMNI properties are not configurable:

- Address width

This value is fixed at 32 bits.

- Read and write data widths

These values are fixed at 32 bits.

## 2.8.7 PCDC configuration options

PCDC units provide various options that you can configure to meet the specific requirements of your design, including the flit width for each channel.

You can configure the following options:

- PCDC synchronization mode:
  - Use asynchronous mode when the clocks are not synchronized.
  - Use synchronous (1:1) mode when the clocks are identical.
  - Use synchronous (1:N) mode when the clocks are synchronized and the first clock has a lower frequency than the second clock. The positive edge of the first clock must always coincide with a positive edge of the second clock.
  - Use synchronous (M:1) mode when the clocks are synchronized and the first clock has a higher frequency than the second clock. The positive edge of the second clock must always coincide with a positive edge of the first clock.
- Maximum number of credits for each RP that can be accepted at the input and output ports
- Flit width for each channel

When a PCDC is set to asynchronous, you can also configure the following options:

- Number of synchronizer register stages from 2–4
- Buffer depth for data and credit FIFO
- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the NI-710AE Configuration and Integration Manual.

## 2.8.8 Router configuration options

Router units provide various options that you can configure to meet the specific requirements of your design, including the numbers of inputs and outputs.

You can configure the following options:

- Number of router inputs from 1–8
- Number of router outputs from 1–8
- Channel credits for each source, destination, and RP
- Frequency of arbitration decisions that disregard QoS

- Include optional pipeline register slices for retiming. These pipeline slices provide flexibility in trading latency for higher frequency. For more information, see *Configuring NI-710AE unit-level retiming options* in the NI-710AE Configuration and Integration Manual.

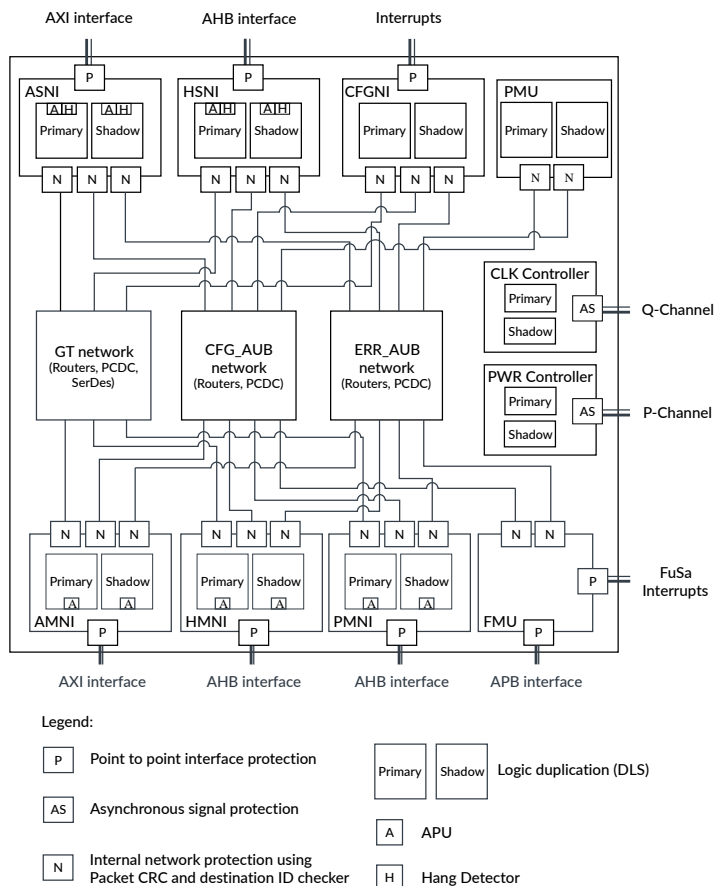
## 3. Fault Detection and Control mechanisms

This chapter describes the general mechanisms for Fault Detection and Control (FDC) that are built into the NI-710AE architecture.

NI-710AE provides several Functional Safety (FuSa) features to detect random hardware faults, systematic faults, and latent faults.

The following figure shows an overview of the NI-710AE functional safety features.

**Figure 3-1: The NI-710AE functional safety features**



NI-710AE supports the following safety features:

### External interface protection

AMBA interface parity protects all external AMBA interfaces. In addition, NI-710AE supports a mode in which AXI interfaces connected to Cortex-R52 or Cortex-R52+ are protected by Error Correcting Code (ECC). This mode maintains compatibility with the Cortex-R52 legacy interface protection scheme. For more information, see [External interface protection](#).

## Logic protection

Functional blocks are duplicated, with the primary and shadow blocks operating in lock-step with each other and with their outputs compared. The shadow block operation is delayed by 2 clock cycles with respect to the primary block, providing temporal diversity to protect against common clock or power faults. For more information, see [Logic protection](#).

## Internal network protection

NI-710AE functional blocks are connected using GT, CFG\_AUB, and ERR\_AUB internal packet transport networks. The internal networks and the network layer components are protected by packet Cyclic Redundancy Check (CRC). For more information, see [Internal network protection](#).

## Access Protection Unit (APU)

NI-710AE provides support for memory partitioning and isolation in mixed criticality systems using APUs. For more information, see [Memory access protection and the Access Protection Unit](#).

## Device isolation over a wire interface

NI-710AE supports a hardware interface to initiate IDM soft resets, significantly reducing latency. For more information on device isolation, see [Device isolation with IDM wire interface](#).

## Hang detection

NI-710AE includes a hang detector mechanism to detect transaction timeouts. For more information, see [Hang detection](#).

## Clock and reset protection

NI-710AE provides protection for common mode faults in clocks and resets. For more information, see [Clock protection](#) and [Reset protection](#).

## FuSa error reporting

When the NI-710AE safety mechanisms detect errors, they send details about the error to a central Fault Management Unit (FMU) for logging.

## Fault Management Unit

FMU functions include FuSa error logging and interrupt signaling. There is a dedicated APB interface into the FMU for fault diagnostics and control. For more information, see [Fault Management Unit](#).

The functional safety features are designed to mitigate the following fault types:

- Random hardware faults
- Systematic faults
- Latent faults

## Random hardware faults

The following functional safety features provide protection against random hardware faults that include transient and permanent faults:

- Point-to-point protection of external interfaces

- Internal network protection
- Asynchronous interface protection
- Logic protection through duplication and delayed lock-step operation

### Systematic faults

The following functional safety features provide protection against systematic faults:

- Access Protection Unit
- Transaction hang detector

### Latent faults

The following functional safety features provide protection against latent faults:

- Checker duplication
- FuSa error injection and reporting using a dedicated error AUB network

## 3.1 External interface protection

This section describes the protection AMBA interfaces, asynchronous interface signals, and miscellaneous interfaces.

### 3.1.1 AMBA interfaces

All external AMBA interfaces on NI-710AE support point-to-point protection using AMBA interface parity. These interfaces include AXI, AHB, and APB interfaces.

Optionally, the AXI interfaces can be configured to support legacy ECC instead of AMBA interface parity when connecting to Cortex-R52 IP point-to-point protection. Point-to-point protection is sufficient for wires and buffers that cannot cause Multiple Bit Errors (MBEs).

An example of an interconnect component that can cause MBEs is a switch. A single fault on a switch mux input can switch the wrong data, causing multiple bits to fail.

NI-710AE supports interface parity protection for point-to-point connections from NI-710AE to another functionally safe IP or FuSa interconnect. Interface parity errors are reported as FuSa errors.



You can configure NI-710AE with external interface protection disabled to work with IPs that do not support interface AMBA interface parity or legacy ECC.

---

### Assumptions of use

Arm expects that:

- NI-710AE is directly connected to the far-end IP with only wires and repeater buffers.
- No complex logic gates or cross bar switches exist in the path, because they could be a source of MBEs.
- The far-end IP checks the parity or ECC bits generated by NI-710AE.
- The far-end IP generates the incoming parity or ECC bits compliant with AMBA Parity Extensions or Legacy Cortex-R52 ECC.

For more information about the AMBA interface signals and the corresponding parity check signals, see the following sections in the NI-710AE Technical Reference Manual:

- *ASNI external interface types and associated signal groups*
- *HSNI external interface types and associated signal groups*
- *PMNI external interface types and associated signal groups*

### 3.1.2 Asynchronous interfaces

The external asynchronous interfaces comprise P-Channel and Q-Channel interfaces which are used for power and clock management respectively.

Duplicated signals with inverse polarity protect the asynchronous interface signals. The duplicated signal is referred to as the CHK signal.

#### Asynch checker

The async checker checks for faults on the asynchronous signal, SIG, and its CHK duplicate, SIGCHK. The async checker ensures that the signal is transmitted to the primary and shadow functional blocks only when SIG and SIGCHK are both asserted or both deasserted.

#### Transient faults

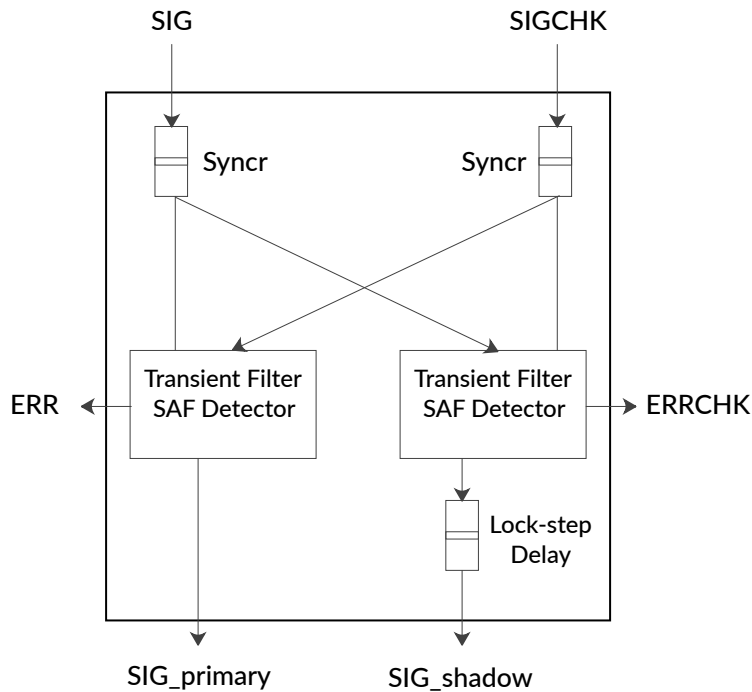
Transient assertions or deassertions of either SIG or SIGCHK are treated as transient faults. The async checker filters these faults for reliability. The protection logic prevents these faults from reaching mission mode logic and causing errors.

#### Permanent faults

Stuck-At Faults (SAFs) on SIG or SIGCHK are permanent faults. The async checker detects these faults using an SAF timeout mechanism and flags the faults as FuSa errors.

The following figure shows the functionality of the async checker.

**Figure 3-2: Asynchronous interface signal protection**



### CHK signal timing

The timing skew between SIG and SIGCHK must be less than the maximum that the SAF detection logic allows. The skew depends on the following factors:

#### Lock-step delay (D1)

The temporal skew between lock-step primary and shadow logic blocks. On NI-710AE, the lock-step delay is set to two cycles.

#### SAF timeout count (T2)

The SAF detector timeout count. On NI-710AE, the SAF timeout count is set to 256.

#### Synchronizer uncertainty delay (U2)

Uncertainty in the synchronizer delay used to synchronize SIG and SIGCHK.

#### Clock Ratio (CR)

Equal to (NI-710AE clock frequency) / (external controller clock frequency).

The permitted maximum implementation skew ( $S1_{max}$ ) between SIG and SIGCHK in external controller clock cycles is calculated as  $S1_{max} = (T2 - U2)/CR - D1$ .

### Example

Assume that:

- NI-710AE clock frequency = 1000MHz
- External controller frequency = 50MHz



- Synchronizer uncertainty delay (U2) = 2

Based on these frequencies, the CR is calculated as  $CR = 1000\text{MHz}/50\text{MHz} = 20$ .

The allowable skew that is permitted is:

$$S1_{\max} = ((T2 - U2)/CR) - D1 = ((256 - 2)/20) - 2 = \sim 10 \text{ external controller clock cycles}$$

Therefore, the SoC integrator is allowed no more than 10 cycles for implementation skew between SIG and SIGCHK.

### Assumptions of use

Arm expects that on asynchronous interfaces:

- NI-710AE is directly connected to the external controller or external IP block with only wires and repeater buffers.
- No complex logic gates or cross bar switches exist in the path, because they could be a source of MBEs.
- The external controller or external IP block has an async checker that checks the CHK bits that NI-710AE generates.
- The inbound SIG and SIGCHK bits are driven within the permitted maximum implementation skew.
- FuSa error reporting is disabled before clock gating asynchronous domain crossing logic (PCDC) to avoid spurious FuSa errors from async checkers.
- The External Clock Controller (ECC) factors in both QACTIVE and QACTIVECHK signals in making clock wake up decisions to ensure that clocks are active in the presence of faults.

For a valid comparison of QACTIVE and QACTIVECHK outputs, the NI-710AE clocks must be ON.

NI-710AE asynchronous interface signals include the following:

- Clock management signals
- Power management signals
- Interrupt and event signals
- FMU interface signals

### 3.1.3 Miscellaneous external interface protection

Miscellaneous external interface protection comprises signals that are not part of an AMBA or asynchronous interface.

The following table specifies the protection policy for these signals.

**Table 3-1: External interfaces and their signal protection schemes**

Interface	Signal protection scheme
Clocks and resets	Duplication with same polarity
DFT	Duplication with same polarity
IDM - resets	Duplication with same polarity
IDM - non reset signals	Duplication with inverse polarity
Interrupts	Duplication with inverse polarity
Configuration strap input	Duplication with inverse polarity
FMU	Odd parity
APU	Odd parity
PMU and debug	None. For more information see <a href="#">Debug trace and PMU interface</a> .

### Debug trace and PMU interface

The debug trace and PMU interface signals do not have protection.

The debug trace and PMU functions must be turned off during mission critical operation and the corresponding interface signals must be in their quiescent state. NI-710AE detects and reports violations of these requirements as FuSa errors.

### Assumptions of use

Arm makes the following assumptions of use:

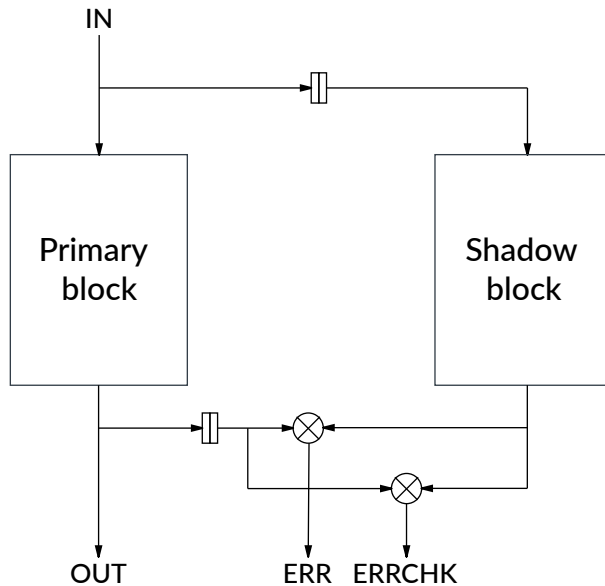
- Debug trace and PMU functions are turned off during mission critical operation.
- Debug trace and PMU interface signals remain at their quiescent values during mission critical operation.

## 3.2 Logic protection

The functional blocks of NI-710AE are protected using the Dual lock-step (DLS) scheme.

The functional block is duplicated. The original block is designated as the primary block and the duplicated block is designated as the shadow block. The shadow block operates in lock-step with the primary block, with a fixed delay of two clock cycles. The outputs of the shadow block serve as reference outputs against which the primary block outputs are checked.

**Figure 3-3: NI-710AE full duplication**



The following functional blocks are protected by duplication and DLS checking:

- ASNI
- AMNI
- HSNi
- HMNI
- PMNI
- CFGNI
- Clock and power controller
- PMU

The clocking and reset are also duplicated. To provide redundancy in the reset and clock trees, the primary and shadow blocks have separate clock and reset inputs. If a branch of the reset or clock tree fails in either the primary domain or the shadow domain, the other domain detects it.

For more information on reset assumptions and requirements that are related to lock-step logic and FuSa, see [Reset protection](#).

## Checkers

The lock-step checkers consist of an XOR tree for comparing the outputs of the primary and shadow blocks. The same parameterized checker component is instantiated throughout the design for uniformity.

The checkers are known to be power-hungry. To mitigate this condition, payload outputs from primary and shadow blocks are compressed into their respective 8-bit Cyclic Redundancy Check (CRC) values which are then compared. In addition, qualification is used wherever possible so they only check the outputs when necessary. For example, an AXI bus checker checks the bus payload only when the valid bits are asserted. This methodology is necessary to:

- Prevent flagging errors on benign glitches on the bus payload when nothing is reading the bus.
- Prevent false error from being asserted because of **UNKNOWN** values on the bus, from RAMs or from uninitialized datapath flops.

### Non-resettable flops

All non-resettable flops that could not be proven benign have been changed to resettable versions.

## 3.3 Internal network protection

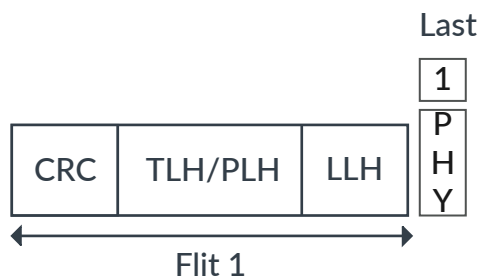
The endpoints xSNI and xMNI are connected using the internal network. This internal network consists of GT interfaces, AUB interfaces, and several network layer components, for example routers, SERDES, and PCDC (domain bridges).

Figure 3-1: The [NI-710AE functional safety features](#) on page 68 shows an example configuration with the GT and AUB network which comprise the internal network.

The internal network is protected using packet Cyclic Redundancy Check (CRC) when the config parameter `globalParameters/internalNetworkProtection` is set to enable. When enabled, the CRC is computed at the source endpoint and included as the last flit element in the packet. At the destination endpoint, the CRC is re-computed on the packet and compared with the CRC embedded in the packet. A CRC error is signaled on a miscompare. For more information about global parameters, see the *FuSa parameters* section in the NI-710AE Technical Reference Manual.

The following figure shows an example of a GT packet with CRC.

**Figure 3-4: Modified packet with CRC**

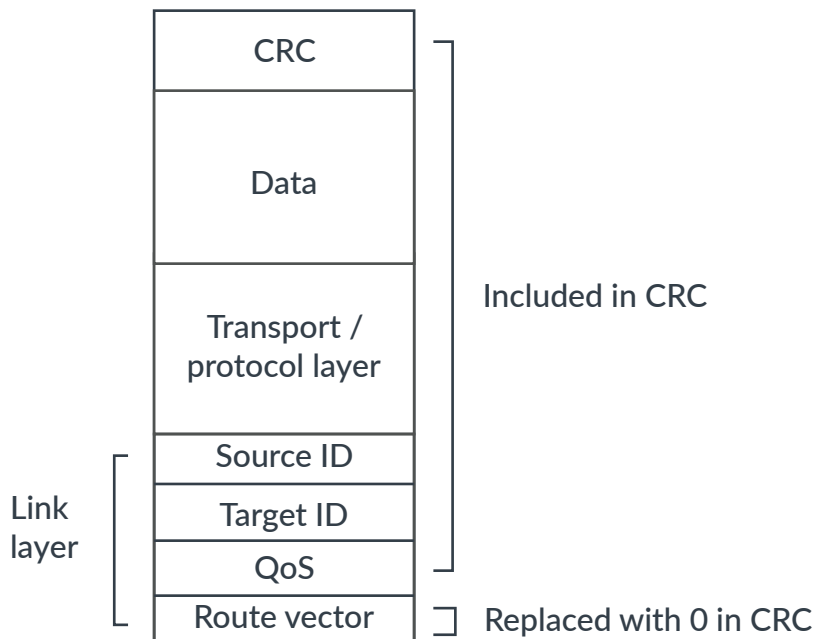


**Single flit GT packet with CRC**

The CRC is calculated across all fields of the packet except for the route vector in the link layer:

- The route vector changes as the packet progresses across the network.
- The value in the route vector field at the destination is different to the value in the route vector field at the source.
- For CRC calculations, the route vector is zeroed out at both the source and destination.

**Figure 3-5: Route vector in the link layer**



Protection for the route is covered by comparing the target ID with the destination ID when the packet arrives at the destination.

## 3.4 Memory access protection and the Access Protection Unit

NI-710AE implements optional Access Protection Units (APUs) that software can use to implement memory protection and isolation from devices with different integrity levels through your system. The APU supports freedom from interference principles in mixed criticality systems and preserves the integrity of critical memory and peripherals.

Software can use APUs to define protection levels for system memory regions and to assign access permissions to those regions for transaction generators in the system. These protection levels and access permissions can then be used to control access to the memory regions.

APUs can also be used to augment existing memory protection systems and enable fine-grained access control to address regions in peripherals. To provide this functionality, APUs include

hardware protection mechanisms that software can program and control. These mechanisms protect against corrupt but protocol-conformant transactions initiated by low integrity requesters.

You can enable APUs on individual endpoints in your design. For more information about how APUs are instantiated in a system, see [APU architecture](#).

When enabled for an endpoint, the APU defines the access permissions to address regions for entities that can send transactions to the endpoint. The APU uses the unique identifier associated with a transaction to determine the originating entity. For more information about APU entities, APU address regions, and APU IDs, see [APU definitions](#).

APUs drop unauthorized write requests and complete unauthorized memory read requests by returning null data. This behavior preserves the data integrity of safety-critical memory in the system. As a result, APUs allow low integrity subsystems to coexist with high integrity subsystems without compromising the system safety goals. You can also configure APUs to report unauthorized accesses as FuSa errors, bus errors, or both. For more information about how the APU handles transactions, see [APU transaction filtering](#).

Software can use the APU functionality to isolate entities that operate at different safety integrity levels. For example, you can program APUs to assign restrictive access rights to low integrity requesters so that critical memory is protected. For more information about enabling, configuring, and programming the APU, see [Configuring the APU](#).

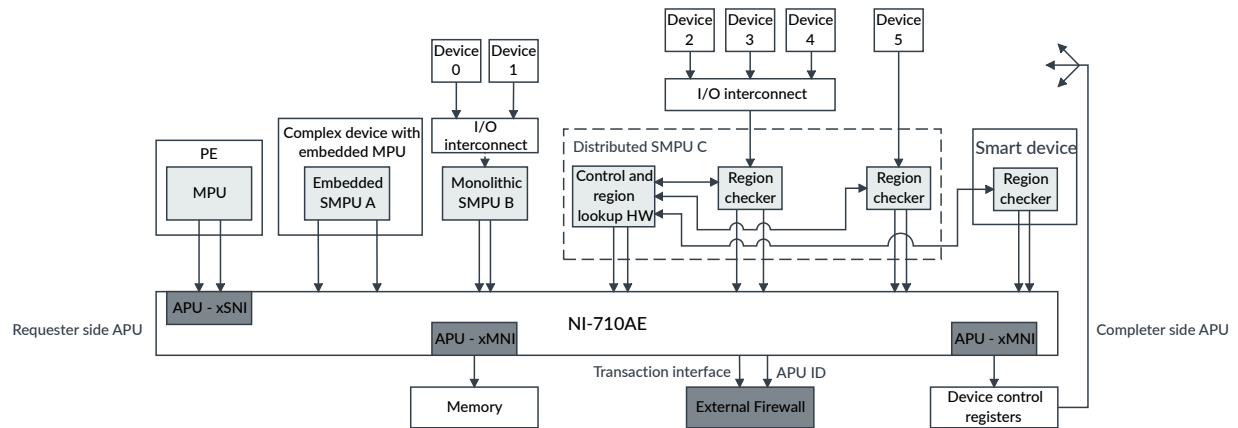
### 3.4.1 APU architecture

When you configure NI-710AE, you can choose to instantiate an APU at individual endpoints. APUs provide access control on both the requester and completer sides of the interconnect.

Requester-side APUs are in xSNIs and completer-side APUs are in xMNIs. On the requester side, you can use an APU to control requester access to specific memory regions and provide isolation between requesters. On the completer side, you can use the APU outputs in a downstream firewall to provide access protection to and from completer peripherals and SRAM.

The following figure shows an example system with APUs instantiated at the requester and completer sides of NI-710AE.

**Figure 3-6: Example NI-710AE system with APUs on both xSNIs and xMNIs**



For more information about enabling, configuring, and programming the APU, see [Configuring the APU](#).

### 3.4.2 APU definitions

APU entities send transactions to an endpoint, and each transaction is associated with a unique value that identifies the entity. APU address regions are regions for which the access requirements are defined in the APU.

When an endpoint with an APU receives a transaction, the endpoint checks whether the transaction address is in a region that is protected by the APU. If so, the endpoint uses the APU ID of the entity that sent the transaction to check whether the entity is permitted to access the address region. For more information about how the APU uses entities, APU IDs, and address regions to check transaction permissions, see [APU transaction filtering](#).

There are specific definitions, requirements, and configuration options for APU entities, APU IDs, and APU address regions.

#### APU entities

An APU entity is any source of transactions into the interconnect. For example, an APU entity could be an individual component, a group of components, or a software context, such as a subsystem or virtual machine. The specific grouping of components in an entity depends on the freedom from interference requirements of the system.

An APU entity can only access memory regions for which the entity has appropriate access permissions and attributes. Software can define access rights separately for each APU entity.

#### APU IDs

The APU ID is a unique value that is assigned to an entity. Every transaction processed by the interconnect is associated with an APU ID value. The APU uses the APU ID for an entity to define access permissions and filter transactions accordingly. APU ID values are driven through APUID signals on a dedicated sideband interface.

You can configure the width of the APUID signals as 0 bits, 4 bits, or 8 bits. The width of the signals determines the maximum number of entities that you can specify in your configuration. Setting the width of the APUID signals to 0 bits removes the signals from the configuration.

If the APUID signal width is nonzero, each APU entity must have a unique identifier. The way in which the system derives the APU ID values is **IMPLEMENTATION DEFINED** and is the responsibility of the system integrator.

### APU address regions

An APU address region is an address region for which you define access requirements. You can configure 4, 8, 16, 20, or 32 address regions for each APU. You define the properties of the APU address regions, including the size and memory attributes, by programming specific APU configuration registers. For more information, see [Configuring the APU](#).

An APU address region can be defined as a background region or a foreground region. A background region is a larger address region with a single set of attributes. A foreground region is a subregion of a background region, which might have a different set of attributes. You can specify multiple foreground regions inside a background region.

The address regions for an APU can have a minimum granularity of either 64 bytes or 4KB, depending on the value of the **APU Region Granularity of 4K** configuration option. If this configuration option is enabled, then the minimum granularity is 4KB. If **APU Region Granularity of 4K** is disabled, then the minimum granularity is 64 bytes. You can set the value of this parameter for each endpoint.



When configured to 64 bytes, the APU needs to check if the incoming request size accounting for burst type crosses the 64 byte boundary. This extra logic may impact the achievable frequency.

---

You can define address region permissions for up to four entities, in other words up to four different APUID values. The access permissions can be any combination of the following types:

- Secure read
- Secure write
- Non-secure read
- Non-secure write

### 3.4.3 APU transaction filtering

When enabled for an interface, the Access Protection Unit (APU) filters transactions to protected address regions. The APU handles transactions differently according to the transaction type.

You can configure individual APUs in separate endpoints. When enabled, the APU checks each transaction that arrives at the endpoint. You can also add extra functionality for further handling of blocked transactions.



The transaction checking differs slightly depending on the endpoint in which an APU is instantiated:

#### **ASNI**

All request errors are checked for each transaction during the address decode request phase in the ASNI.

#### **AMNI**

All request errors are checked for each transaction during the request converter phase in the AMNI.

#### **HSNI**

All request errors are checked for each transaction during the address decode request phase in the HSNI.

#### **HMNI**

All request errors are checked for each beat during the address phase in the HMNI.

#### **PMNI**

All request errors are checked for each transaction. In other words, only the initial Generic Transport (GT) address is presented to the APU during the request phase in the PMNI.

When a transaction is received from an entity, the endpoint looks up the address of the transaction in the APU registers. The endpoint determines whether the address is in a protected region by comparing the transaction address against the programmed address regions. Where the transaction address is in a protected region, the endpoint uses the entity APUID signal to check whether the entity is permitted to access the region. For more information about APU entities, APU IDs, and APU address regions, see [APU definitions](#).

If the entity and the transaction both have sufficient privileges for the address region, then the APU allows the endpoint to send the transaction through the interconnect. When the transaction reaches the target xMNI, the interface drives the APUID signal downstream as a sideband signal. You can use this signal in a custom firewall outside the interconnect, closer to the peripheral.

When either the entity or the transaction does not have sufficient privileges for the address region, the APU terminates the transaction. The APU completes unauthorized memory read requests by returning null data, and drops unauthorized write requests. You can also configure the APU to generate an SLVERR error response, signal a FuSa error to the Fault Management Unit (FMU), or both. For more information, see [Configuring the APU](#).

You can specify default access permissions for the APU to apply to entities that do not have specific programmed permissions. Additionally, if the APU is not configured to check APU IDs, then you can set overall permissions for all entities. For more information about these settings, see [Specifying entities and access permissions for APU address regions](#).

### 3.4.4 Configuring the APU

To configure the Access Protection Unit (APU), you must enable the APU in endpoints in your configuration and choose some general options in Socrates. You must also enable and configure the functionality to your requirements by programming the configuration registers.

#### Enabling APU support and configuring number of APU address regions

When you configure NI-710AE in Socrates, you choose which endpoints have an APU. You can also choose to configure 4, 8, 16, 20, or 32 APU address regions. The configured address region number also determines the number of APU registers that are instantiated for programming the address regions.

For more information about the registers that define the properties of the APU address regions, see [APU address region registers](#).

#### Programming the APU

To set up the properties of the APU, including the address regions and entity access permissions, program the APU configuration registers. The APU has a dedicated 4KB configuration node which requesters can access through the configuration network.

For examples of how to program the APU to achieve specific outcomes, see [APU programming examples](#).

##### 3.4.4.1 Defining address ranges for APU address regions

Each NI-710AE Access Protection Unit (APU) has registers for configuring the address regions it protects. There are some constraints on how you configure the address regions that you must be aware of.

To define the address range for a region, there are register fields for programming the region base address and limit address. You can define the address regions for an APU at either a 64-byte or 4KB granularity, using the **APU Region Granularity of 4K** configuration option. If this configuration option is enabled, then the minimum granularity is 4KB. If it is disabled, then the minimum granularity is 64 bytes. You can set the value of this parameter for each endpoint.

Regions can range in size from the minimum granularity setting to the maximum memory size that your system supports. You define bits[63:6] for the base address and limit address of each region using the following registers, where <n> refers to the address region number:

- PRBAR<n>\_LOW
- PRBAR<n>\_HIGH
- PRLAR<n>\_LOW
- PRLAR<n>\_HIGH

The base address is defined as {PRBAR<n>\_HIGH.region\_base\_addr : PRBAR<n>\_LOW.region\_base\_addr, 0x00}. The limit address is defined as {PRLAR<n>\_HIGH.region\_limit\_addr : PRLAR<n>\_LOW.region\_limit\_addr, 0x3F}.



Each address region can be either a foreground or a background region. For more information, see [Memory access protection and the Access Protection Unit](#).

The following restrictions apply to how you define address ranges when programming the APU address regions:

- Two foreground regions must not overlap.
- Two background regions must not overlap.
- The region\_base\_addr fields together must be aligned to the size of the defined address region. If the combined region\_base\_addr fields are not aligned to the size, then they are aligned to the minimum granularity. In other words, if **APU Region Granularity of 4K** is disabled, the lower 6 bits of the address are ignored. If **APU Region Granularity of 4K** is enabled, the lower 12 bits of the address are ignored.

If two foreground regions overlap or two background regions overlap, then a transaction which hits both address regions triggers a permission fault.

As long as there is no overlap of regions of the same type, foreground regions can overlap with background regions. Therefore, if an address matches both a foreground and background region, then the APU prioritizes the foreground region access permissions.

#### 3.4.4.2 Specifying entities and access permissions for APU address regions

Each Access Protection Unit (APU) address region supports access permissions for up to four entities. You configure these properties by programming the APU address region registers.

Entity transaction permissions can be either Non-secure, Secure, or both. You can specify the permissions for reads and writes separately for each entity.

You can use the APU address region registers to achieve various outcomes for an address region:

- Assigning specific access permissions for separate entities
- Assigning default access permissions for entities that do not have specific programmed permissions
- Assigning overall permissions for all entities when the APU does not use APU IDs

#### Assigning specific access permissions for separate entities

To assign entities to an address region, program the entity APU ID values into the following fields:

- The id0 and id1 fields of the PRID<n>\_LOW register
- The id2 and id3 fields of the PRID<n>\_HIGH register

To set the access permissions for each entity, program the required values into the following fields:

- The access\_permission0 and access\_permission1 fields of the PRID<n>\_LOW register
- The access\_permission2 and access\_permission3 fields of the PRID<n>\_HIGH register

For example, to enable Non-secure and Secure reads and writes for entity 0 of an address region, set the access\_permission0 field to 0b00001111.



Do not enter the same APU ID value into two different register fields for the same region. If the same entity identifier is entered twice, then two different access permissions could be applied to a single entity, which represents a security risk. For example, consider a scenario in which one access permission allows Secure accesses only, but the other allows all access types. In this case, Non-secure accesses from the entity to the memory region are permitted.

To enable APU address filtering for an entity, that entity must be marked as valid in the APU registers. The id\_valid field of the PRLAR<n>LOW register contains a validity bit for each entity that can access the address region. To mark an entity as valid, and so enable APU address filtering for that entity, set the corresponding bit to 1.

If you set the validity bits to 0, the APU behavior depends on which bit is set to 0. If any of the id\_valid[3:1] bits are set to 0, then filtering for the corresponding APU ID is disabled. Therefore, if there is an APU ID match with one of these IDs, then the match is not valid, and the APU blocks the transaction.

If id\_valid[0] is set to 0, you can use this setting to specify a default access permission.

### Assigning default access permissions for entities that do not have specific programmed permissions

You can use the PRLAR<n>LOW.id\_valid[0] bit setting to assign default access permissions. The APU can use the default access permissions in the case where an incoming APU ID does not match any of the programmed ones. To assign default access permissions, use the following register settings:

- Set the PRLAR<n>LOW.id\_valid[0] bit to 0.
- Set the default access permission using the PRID<n>\_LOW.access\_permission0 field.

In this scenario, if there is no match between the incoming APU ID and the programmed IDs, the APU uses the default access permission to determine whether the access is permitted.

### Assigning overall permissions for all entities when the APU does not have assigned APU IDs

If the APU does not use APU IDs, in other words the **APUID Width** option is set to 0, then the APU does not perform ID matching. In this scenario, you can still specify overall access permissions for the APU to check transactions against. To determine the overall access permissions, the APU performs a logical OR operation on all four access\_permission{0-3} fields, and uses the result for the address region.

### 3.4.4.3 Unlocked and locked APU address regions

When programming the Access Protection Unit (APU) address region registers, you can optionally lock the registers. This setting prevents further writes to the address region registers until reset.

The APU address regions have two states: unlocked or locked. The regions are unlocked after reset. To lock a region, write 1 to the lock bit of the PRBAR<n>\_LOW register, where <n> is the number of the address region.

When a region is locked, the APU permits no further writes to the address region registers. The region remains locked until you reset NI-710AE. The lock function lets privileged software program the APU and then lock the registers before application programs launch. The software can choose to lock all regions of the APU, or leave some unlocked to allow other software entities to reprogram the region registers.

After an APU region is programmed, the corresponding APU registers must be locked to prevent inadvertent modification of the registers due to systematic faults.

### 3.4.4.4 Programming general APU properties

The Access Protection Unit (APU) APU\_CTLR register contains controls for enabling the APU and some further configuration options. There is also an expected method for enabling the APU.

The APU\_CTLR register determines the following properties for the APU:

- Whether the APU is enabled
- Whether the APU responds to the transaction with an SLVERR when an access permission fault occurs

In the expected method to enable the APU, you use a strap pin to derive the apu\_enable bit value of the APU\_CTLR register after reset. The xSNI that is connected to the most privileged entity, TEO, has the APU\_CTLR.apu\_enable bit configured to 0. All other xSNIs have this bit set to 1. In this state, TEO bypasses the APU, allowing configuration transactions to flow through without permission checks. TEO can now program the APUs for all other endpoints. Until TEO grants permissions to the other entities, accesses from non-privileged entities are treated as access permission faults.

### 3.4.4.5 Order of programming for APU address region registers

You must program the Access Protection Unit (APU) address region registers in a specific order. This order ensures that the APU region is enabled only when all the other information is valid.

You must program the registers in the following order:

1. Program all required fields in the following registers, except for the PRID<n>\_LOW.region\_enable and PRID<n>\_LOW.lock fields:
  - PRID<n>\_HIGH

- PRID<n>\_LOW
  - PRLAR<n>\_HIGH
  - PRLAR<n>\_LOW
  - PRBAR<n>\_HIGH
  - PRBAR<n>\_LOW
2. Program PRID<n>\_LOW.region\_enable.
  3. Program PRID<n>\_LOW.lock, which locks the region until reset. When the region is locked, no further writes to the address region registers are permitted.
  4. Program APU\_CTLR.

You must not reprogram the region registers after you have programmed the APU\_CTLR.apu\_enable field. The requirements to update the programmed APU region register settings depend on whether the PRID<n>\_LOW.lock field is set. If the lock field is set, a reset is required to reprogram the registers. If the lock field is not set, then to update the programmed APU region settings in the registers, use the following sequence:

1. Set the APU\_CTLR.apu\_enable field to 0.
2. Reprogram the values in the relevant PRID<n>\_{HIGH, LOW}, PRLAR<n>\_{HIGH, LOW}, and PRBAR<n>\_{HIGH, LOW} registers.
3. Set the APU\_CTLR.apu\_enable field to 1.

#### 3.4.4.6 APU address region registers

Each Access Protection Unit (APU) has a set of registers for defining the properties of the APU address regions, associated entities, and their access permissions.

The following registers determine the properties of the APU address regions, where <n> corresponds to the region number:

##### **PRBAR<n>\_LOW**

Specifies the following properties for the address region:

- Whether the address region comparison is enabled for this address region
- Whether the address region is a background or foreground region
- Whether the address region is locked
- Bits[31:6] of the base address of the region

For more information, see the *APU PRBAR\_LOW register* section in the NI-710AE Technical Reference Manual.

##### **PRBAR<n>\_HIGH**

Specifies bits[63:32] of the base address of the region. For more information, see the *APU PRBAR\_HIGH register* section in the NI-710AE Technical Reference Manual.

### PRLAR<n>\_LOW

Specifies the following properties for the address region:

- Whether the settings for each entity in the PRID<n>\_LOW and PRID<n>\_HIGH registers are valid
- Bits[31:6] of the limit address of the region

For more information, see the *APU PRLAR\_LOW register* section in the NI-710AE Technical Reference Manual.

### PRLAR<n>\_HIGH

Specifies bits [63:32] of the limit address of the region. For more information, see the *APU PRLAR\_HIGH register* section in the NI-710AE Technical Reference Manual.

### PRID<n>\_LOW

Specifies the following properties for the address region:

- ID of entity 0 and 1 for the address region
- Access permissions for entities 0 and 1 for the address region

For more information, see the *APU PRID\_LOW register* section in the NI-710AE Technical Reference Manual.

### PRID<n>\_HIGH

Specifies the following properties for the address region:

- ID of entity 2 and 3 for the address region
- Access permissions for entities 2 and 3 for the address region

For more information, see the *APU PRID\_HIGH register* section in the NI-710AE Technical Reference Manual.

## 3.4.4.7 APU programming examples

We provide a series of examples to demonstrate the Access Protection Unit (APU) register programming to achieve specific outcomes. The examples demonstrate the interaction of address regions, entities, and permissions.

### Example 1: Two regions, each exclusive to a single entity

This example has two address regions. The first region ranges from 0x0000 to 0x0FFF. The second address region ranges from 0x1000 to 0x1FFF. Each address region is exclusive to a single entity.

```
PRBAR0_LOW.region_base_addr = 0x00000000
PRBAR0_HIGH.region_base_addr = 0x00000000
PRLAR0_LOW.region_limit_addr = 0x0000003F
PRLAR0_HIGH.region_limit_addr = 0x00000000
PRBAR0_LOW.br = 0 PRBAR0_LOW.region_enable = 1
PRLAR0_LOW.id_valid[0] = 1 PRID0_LOW.id0 = ID0 PRID0_LOW.access_permission0 = XX
PRBAR1_LOW.region_base_addr = 0x00000040
PRBAR1_HIGH.region_base_addr = 0x00000000
```

```
PRLAR1_LOW.region_limit_addr = 0x000007F
PRLAR1_HIGH.region_limit_addr = 0x00000000
PRBAR1_LOW.br = 0 PRBAR1_LOW.region_enable = 1
PRLAR1_LOW.id_valid[0] = 1 PRID1_LOW.id0 = ID0 PRID1_LOW.access_permission0 = YY
```

For a transaction that targets an address in the first address region, ranging from 0x0000 to 0x0FFF:

- The APU checks all transactions from entity 0 against permission XX.
- The APU does not permit transactions from any other entity.

For a transaction that targets an address in the second address region, ranging 0x1000 to 0x1FFF:

- The APU checks all transactions from entity 1 against permission YY.
- The APU does not permit transactions from any other entity.

### Example 2: One address region with specific permissions for two entities and no permissions for others

This example has a single address region, ranging from 0x0000 to 0x0FFF. The address region is exclusive to two entities.

```
PRBAR0_LOW.region_base_addr = 0x00000000
PRBAR0_HIGH.region_base_addr = 0x00000000
PRLAR0_LOW.region_limit_addr = 0x000003F
PRLAR0_HIGH.region_limit_addr = 0x00000000
PRBAR0_LOW.br = 0 PRBAR0_LOW.region_enable = 1
PRLAR0_LOW.id_valid[0] = 1 PRID0_LOW.id0 = ID0 PRID0_LOW.access_permission0 = XX
PRLAR0_LOW.id_valid[1] = 1 PRID0_LOW.id1 = ID1 PRID0_LOW.access_permission1 = YY
```

For a transaction that targets an address in the address region ranging from 0x0000 to 0x0FFF:

- The APU checks all transactions from entity 0 against permission XX.
- The APU checks all transactions from entity 1 against permission YY.
- The APU does not permit transactions from any other entity.

### Example 3: The same address regions with specific permissions for two entities and generic permissions for others

This example has a single address region, ranging from 0x0000 to 0x0FFF. The address region has specific permissions for two entities and a set of permissions for any other entity.

```
PRBAR0_LOW.region_base_addr = 0x00000000
PRBAR0_HIGH.region_base_addr = 0x00000000
PRLAR0_LOW.region_limit_addr = 0x000003F
PRLAR0_HIGH.region_limit_addr = 0x00000000
PRBAR0_LOW.br = 0 PRBAR0_LOW.region_enable = 1
PRLAR0_LOW.id_valid[0] = 0 PRID0_LOW.id0 = NA PRID0_LOW.access_permission0 = XX
PRLAR0_LOW.id_valid[1] = 1 PRID0_LOW.id1 = ID0 PRID0_LOW.access_permission1 = YY
PRLAR0_LOW.id_valid[2] = 1 PRID0_HIGH.id2 = ID1 PRID0_HIGH.access_permission2 = ZZ
```

For a transaction that targets an address in the address region ranging from 0x0000 to 0x0FFF:

- The APU checks all transactions from entity 1, ID0, against permission YY.



- The APU checks all transactions from entity 2, ID1, is checked against permission ZZ.
- The APU checks all transactions from any other entity against permission XX.

#### 3.4.4.8 Address range checking

NI-710AE can also perform address range checking.

In the following examples, we use the following criteria:

- `APU_REGION_4K == 0`.
- Burst type is either FIXED, INCR, or WRAP.

##### Example 1 - Burst type is FIXED

In this example, the Burst type is FIXED:

- Burst type = FIXED
- $\text{lowest\_address} = (\text{addr\_i} / \text{burst\_size}) * \text{burst\_size}$
- $\text{start\_address} = \text{lowest\_address}$
- $\text{largest\_address} = \text{start\_address} + \text{burst\_size} - 1$

##### Example 2 - Burst type is INCR

In this example, the Burst type is INCR:

- $\text{data\_transfer\_size} = (\text{burst\_length} + 1) * \text{burst\_size}$
- $\text{lowest\_address} = (\text{addr\_i} / \text{burst\_size}) * (\text{burst\_size})$
- $\text{start\_address} = \text{lowest\_address}$
- $\text{largest\_address} = \text{start\_address} + \text{data\_transfer\_size} - 1$

##### Example 3 - Burst type is WRAP

In this example, the Burst type is WRAP:

- $\text{data\_transfer\_size} = (\text{burst\_length} + 1) * \text{burst\_size}$
- $\text{lowest\_address} = (\text{addr\_i} / \text{data\_transfer\_size}) * (\text{data\_transfer\_size})$
- $\text{start\_address} = (\text{addr\_i} / \text{burst\_size}) * (\text{burst\_size})$
- $\text{largest\_address} = \text{lowest\_address} + \text{data\_transfer\_size} - 1$

##### Example 4 - Burst type, length, and data\_transfer\_size are unused and `APU_REGION_4K == 1`

However in the following example, `APU_REGION_4K == 1` and the Burst type, length, and `data_transfer_size` are unused:

- $\text{lowest\_address} = \text{addr\_i}$
- $\text{start\_address} = \text{addr\_i}$
- $\text{largest\_address} = \text{addr\_i}$

### 3.4.4.9 Support for the APU 4K page as a child node of the endpoint node register

Extend endpoint registers to add a pointer to the APU child page if enabled. The `apu_4k_region` parameter configures the APU to support a minimum address region granularity of 4k. You can enable or disable this parameter for each endpoint.

Add an APU 4k page as a sub feature in the endpoint's page. The endpoint's page lists the number of subfeatures implemented, subfeature type, and pointers to the 4k page of the subfeature. NI-710AE implements only 1 subfeature, which is APU.

#### Subfeature registers

The subfeature registers define the number of subfeatures, the subfeature type, and the subfeature pointer.

**Table 3-2: Subfeatures register**

Bits	Name	Description	Type
[31:0]	Number of subfeatures register	Number of subfeatures register	
[31:16] [15:0]	Subfeature 0 type	Reserved Subfeature type. For more information on subfeature types, see the following table.	
[31:0]	Subfeature pointer	Subfeature pointer	

For more information on the subfeature registers, see the *Programmers model* section of the NI-710AE Technical Reference Manual.

#### Subfeature types

The following table describes the subfeature types and their value.

**Table 3-3: Subfeature types**

Subfeature type	Value
APU	0x0000
SAM	0x0001
FCU	0x0002
IDM	0x0003
RAS	0x0004

#### Address map

The address map would look like similar to the following table, where the APU subfeature page is collated with the endpoint page.

**Table 3-4: Example address map**

Offset	Content
0KB	Global register
4KB	Voltage domain 0 register

Offset	Content
8KB	Power domain 0 register
12KB	Clock domain 0 register
16KB	ASNI 0 register
20KB	ASNI 0 APU subfeature register
24KB	AMNI 0 register
28KB	AMNI 0 APU subfeature register
32KB	Clock domain 1 register
36KB	ASNI 1 register
40KB	ASNI 1 APU subfeature register
44KB	AMNI 1 register
48KB	AMNI 1 APU subfeature register
52KB	Power domain 1 register
56KB	Clock domain 2 register
60KB	ASNI 2 register
64KB	ASNI 2 APU subfeature register
68KB	AMNI 2 register
72KB	AMNI 2 APU subfeature register
76KB	HSNI 0 register
80KB	HSNI 0 APU subfeature register
	...

## 3.5 Device isolation with IDM wire interface

IDM wire interface allows functionally safe applications to isolate the safety island with as little latency as possible.

The IDM wire interface is an alternative to software intervention to trigger the soft reset functionality. With the wire interface, the FMU can immediately isolate unsafe traffic and prevent it from entering the safety island. To significantly reduce latency and ensure that soft reset mode is successfully entered, and the soft reset output is activated, an external soft reset entry request interface (HW) is required. Exiting soft reset mode is still performed using the existing mechanism, that is, over user or software programming.

### Enter soft reset mode (4-phase handshake protocol)

To enter soft reset mode the following occurs:

- The input `idm_ext_sreset_entry_req_sync_i` is asserted to request IDM to activate its output `sresetn_o` to an external device. The `idm_ext_sreset_entry_req_sync_i` remains asserted until the output `idm_ext_sreset_entry_ack_o` is asserted.
- IDM asserts the `idm_ext_sreset_entry_ack_o` to indicate that the soft reset request is granted and the IDM is in `sreset_mode`, as also reflected by the `idm_reset_control.reset`

register bit. The `idm_ext_sreset_entry_ack_o` remains asserted up to the first cycle `idm_ext_sreset_entry_req_sync_i` is deasserted.

- Deassertion cycles of request or acknowledge in between, must separate a subsequent request to enter soft reset mode from this external interface. This deassertion has no impact on the internal soft reset state, but is only present to separate the two soft reset entry requests.

## Exit soft reset mode

To exit soft reset mode use the existing IDM mechanism, that is, write `0b0` to the appropriate `idm_reset_control.reset` control register bit. For more information on the `idm_reset_control.reset` control register bit, see the appropriate `idm_reset_control` register section in the NI-710AE Technical Reference Manual:

- *ASNI register summary*
- *AMNI register summary*
- *HSNI register summary*
- *HMNI register summary*
- *PMNI register summary*
- *Power domain register summary*

Hardware requests, over the external soft reset entry request interface, and software auto requests to enter soft reset mode, can come in any order and can overlap.

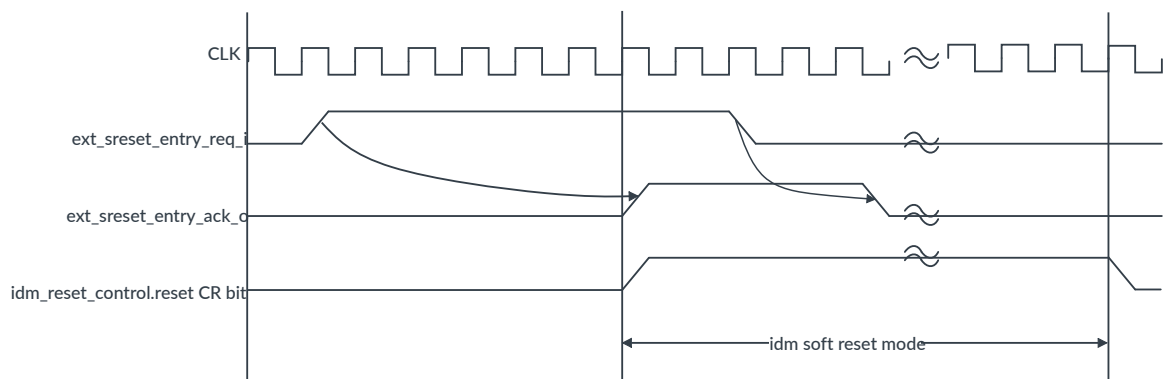
## Soft reset entry request interface

The `idm_ext_sreset_entry_req_sync_i` is a synchronized input signal.

## Soft reset entry request timing diagram

The following figure shows the soft reset entry request and acknowledgment.

**Figure 3-7: Soft reset entry request timing diagram**



### 3.5.1 Soft reset entry mode scenarios

The following examples describe some soft reset entry scenarios, using the external pin interface and software based entry overlap.

#### **Software triggered sreset entry occurs at the same time as the assertion of ext\_sreset\_entry\_req\_sync**

When the sreset entry is successful, the ext\_sreset\_entry\_ack\_o is asserted.

#### **Software triggered sreset entry occurs, no assertion of ext\_sreset\_entry\_req\_sync**

When the sreset entry is successful, the ext\_sreset\_entry\_ack\_o is not asserted.

#### **IDM is already in sreset mode, ext\_sreset\_entry\_req\_sync is asserted**

ext\_sreset\_entry\_ack\_o is asserted in the next cycle after the assertion of ext\_sreset\_entry\_req\_syn.

#### **IDM is exiting sreset mode, ext\_sreset\_entry\_req\_i is asserted**

The ext\_sreset\_entry\_req\_i (entry request) is ignored. When IDM has successfully exited soft reset mode, ext\_sreset\_entry\_req\_i (entry request) is then processed. When the soft reset entry is successful, the ext\_sreset\_entry\_ack\_o is asserted.

#### **Only ext\_sreset\_entry\_req\_sync triggers a soft reset entry. IDM successfully enters soft reset mode, ext\_sreset\_entry\_ack\_o is asserted. Software triggers the soft reset exit.**

Interconnect Device Management (IDM) successfully exits soft reset mode. IDM soft reset state machine progresses to idle state. However, ext\_sreset\_entry\_req\_sync remains asserted and therefore ext\_sreset\_entry\_ack\_o also remains asserted. IDM does not detect the asserted ext\_sreset\_entry\_req\_sync as a new soft reset entry request.

#### **The ext\_sreset\_entry\_req\_sync is asserted before the assertion of adp\_sreset\_ack**

Software or Auto HW triggers a soft reset entry. This trigger initiates the soft reset mechanism, adp\_sreset\_req is asserted to the endpoint adaptor which is then followed by the assertion of adp\_sreset\_ack from the endpoint adaptor. The ext\_sreset\_entry\_req\_sync is asserted before the assertion of adp\_sreset\_ack. IDM then successfully enters soft reset mode, q\_reset\_control[reset] bit is set. The ext\_sreset\_entry\_ack\_o is also asserted.

The ext\_sreset\_entry\_req\_sync is deasserted which is then followed by the deassertion of ext\_sreset\_entry\_ack\_o to complete the 4-phase request-acknowledge handshaking ext\_sreset\_entry request. Software triggers the soft reset exit (adp\_sreset\_req is deasserted) which is then followed by the deassertion of adp\_sreset\_ack. It is then followed by the negation of q\_reset\_control[reset]. IDM successfully exits soft reset mode.

## 3.6 Hang detection

The hang detector is a transaction timeout detection mechanism that is based on transaction tracker occupancy.

This mechanism detects transaction hang conditions and addresses systematic faults that are caused by incorrect programming or design errors. Hang detectors are duplicated for latent fault protection. One instance is connected to the primary block while the other instance is connected to the shadow block.

A transaction timeout value can be specified independently for each device by programming the corresponding FDC configuration register.

Hang detectors are present in the following units:

- ASNI
- HSNi

See the *ASNI hang\_detector\_ctrl register* and *HSNI hang\_detector\_ctrl register* sections in the NI-710AE Technical Reference Manual.

## 3.7 Clock protection

NI-710AE clocks are protected by duplication of the clock signals and the corresponding clock trees.

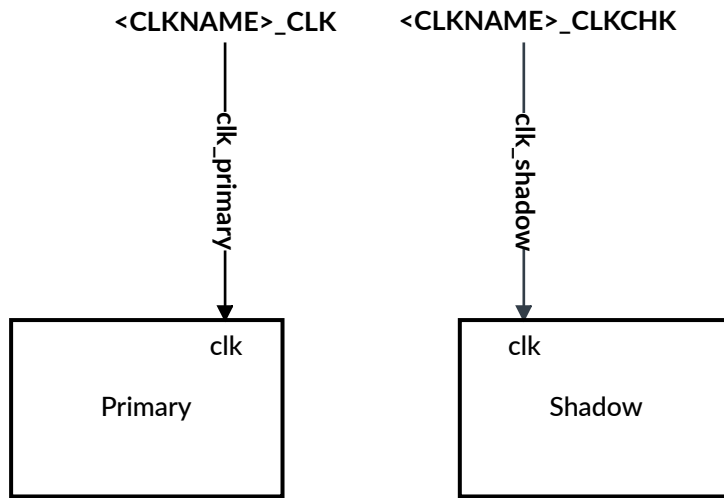
NI-710AE uses a primary CLK input and a redundant CLKCHK input in each clock domain. Both clock inputs have the same frequency and are in phase with each other.

The primary clock drives the primary clock tree to the primary block and the redundant clock drives the shadow clock tree to the shadow block.

A fault in either of the clock inputs or in either of the primary or shadow clock trees manifests itself as an output mismatch between primary and shadow blocks. The lock-step checker detects such faults as a Dual Lock-Step (DLS) fail.

The following figure shows the clock protection mechanism.

**Figure 3-8: Clock protection**



## 3.8 Reset protection

NI-710AE resets are protected by duplication of the reset signals and the corresponding reset trees.

NI-710AE uses a primary RESETn input and a redundant RESETnCHK input in each clock domain.

NI-710AE provides protection against the following types of reset faults:

- [Transient faults](#)
- [Internal reset faults](#)

For more information about the requirements of the NI-710AE resets, see [FuSa reset requirements](#).

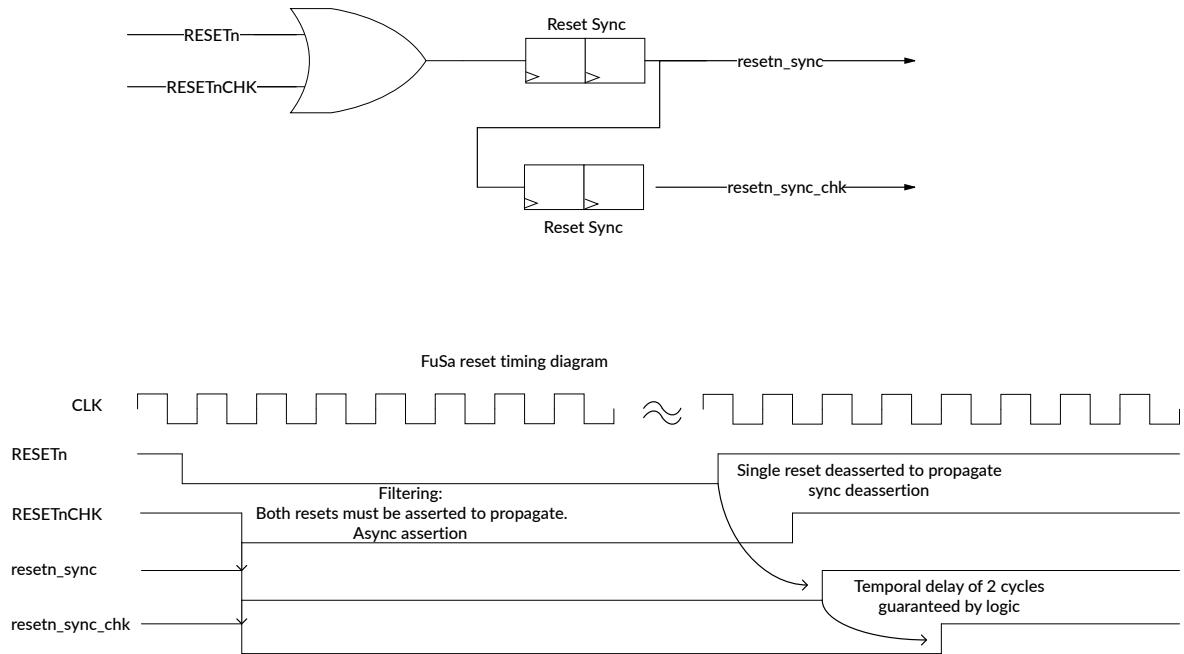
### 3.8.1 Transient faults

NI-710AE protects against transient reset faults by suppressing spurious reset assertions.

For example, resetn\_sync and resetn\_sync\_chk outputs to the functional blocks are not asserted unless RESETn and RESETnCHK inputs are both asserted. Therefore, spurious assertions of RESETn and RESETnCHK inputs are filtered out.

The following figure shows the reset transient fault protection mechanism.

**Figure 3-9: Reset transient fault protection**



## 3.8.2 Internal reset faults

NI-710AE protects against internal reset faults by duplicating the reset trees. Each of the primary block and the shadow block has its own reset tree.

A fault in either of the primary or shadow reset trees manifests itself as an output mismatch between primary and shadow blocks. The lock-step checker detects such faults as a Dual Lock-Step (DLS) fail.

## 3.8.3 FuSa reset requirements

Various requirements apply to both reset assertion and deassertion in NI-710AE.

### Reset assertion

Both reset inputs, `RESETn` and `RESETnCHK`, must be asserted before reset signals to the functional blocks, `resetn_sync` and `resetn_sync_chk`, are asserted.

When asserted, to ensure reset is propagated through all logic pipeline stages, resets must remain asserted for at least 40 clock cycles.

Reset assertion is propagated asynchronously to the NI-710AE logic.



## Reset deassertion

Either of the reset inputs, RESETn and RESETnCHK, can be deasserted before reset signals to the functional blocks, resetn\_sync and resetn\_sync\_chk, are deasserted.

Reset deassertion to the functional blocks is synchronous. Deassertion of resetn\_sync\_chk is delayed by two clock cycles from the deassertion of resetn\_sync. This delay ensures lock-step between primary and shadow blocks when coming out of reset.

## 3.9 Fault Management Unit

The NI-710AE Fault Management Unit (FMU) monitors and reports errors throughout the interconnect. Distributed interconnect components send errors to the FMU, which collates the errors into a software-accessible record and can signal error interrupts to the system for handling.

The FMU is composed of two different types of subcomponents: mini FMUs and a central FMU. In a NI-710AE configuration, there are several mini FMUs but only one central FMU. The mini FMUs are connected to the central FMU through an error network.

### Mini FMUs

These FMU subcomponents are distributed across the interconnect, one for each endpoint. Mini FMUs receive inputs from error sources, such as the end-to-end network protection Error Detection and Correction (EDC) checkers. The output from mini FMUs takes the form of error packets that are sent to the central FMU for handling. For more information about how the mini FMUs receive and transmit errors, see [FMU error logging process](#).

### Central FMU

This FMU subcomponent is located in the same voltage, power, and clock domain as the global Configuration Network Interface (CFGNI). The central FMU receives error reporting inputs from the distributed mini FMUs and stores the details in an error record table. For more information about how the central FMU logs errors, see [FMU error record table](#).

The output from the central FMU takes the form of interrupts, which the central FMU can use to stop normal system operation for error handling. The central FMU can issue fault handling interrupts, error recovery interrupts, and critical error interrupts. For more information, see [FMU interrupts](#).

### Error network

This network connects the distributed mini FMUs to the central FMU. The error network is separate from the NI-710AE data and configuration networks. Using a dedicated network for error reporting avoids interference and backpressure from other traffic, which ensures that errors are reported in a timely manner.

If at least one of the following safety mechanism parameters is enabled, NI-710AE automatically implements an FMU and an error network:

- DLS logic protection
- AMBA interface protection

- Hang detector
- Access Protection Unit (APU)

You can specify which safety mechanisms report errors to the FMU. For more information, see [Controlling which safety mechanisms report errors to the FMU](#).

The FMU provides a register that you can use to inject errors into a safety mechanism. For more information, see [FMU error injection](#).

As part of the discovery process, software must identify each node that can report errors to the FMU. For more information, see [Software initialization and the FMU](#).

A locking mechanism is used to protect the FMU registers. For more information, see [FMU register protection mechanism](#).

### 3.9.1 FMU error logging process

When the NI-710AE safety mechanisms detect errors, they send details about the error to a central Fault Management Unit (FMU) for logging. NI-710AE follows a specific process to log errors in the FMU error record table.

During normal operation, the safety mechanisms monitor interconnect traffic to detect errors. When an error is encountered, the safety mechanism signals to the local mini FMU that an error has occurred. The mini FMU then constructs an error packet containing the following information:

- The safety mechanism that detected the error
- The error type, which can be one of the following:
  - Non-critical
  - Critical
- The ID of the endpoint at which the error was detected
- The type of endpoint at which the error was detected

The mini FMU sends out the error packet on the error network that connects all the mini FMUs to the central FMU.

When an error packet arrives at the central FMU, the central FMU updates the relevant entry in the error record table with the information in the packet. If errors are already logged for an error record entry, NI-710AE follows the Arm RAS System Architecture v1.1 rules for error prioritization. For more information, see the *Prioritizing errors, RAS System Architecture v1.1* section of the [Arm® Architecture Reference Manual for A-profile architecture](#).

If the error requires handling, the central FMU can also signal to the system that an error has occurred through various level-sensitive interrupt ports.

The central FMU has a dedicated APB interface. This interface allows error recovery software to read the contents of the error record table and act on the errors.

### 3.9.2 FMU error record table

The error record table in the Fault Management Unit (FMU) contains one entry for each endpoint in a NI-710AE configuration. Individual entries in the error record table store information about the error, the endpoint that reported the error, and the transaction that caused the error.

In the error record table, entries are organized first by power domain and then by endpoint type. The following example shows how these rules are applied to arrange the first few entries in an error record table.

**Table 3-5: Example showing ordering of entries in an error record table**

Error record entry number	Associated endpoint
0	Power domain 0, ASNI 0
1	Power domain 0, HSNI 0
2	Power domain 0, CFGNI 0
3	Power domain 1, AMNI 0
4	Power domain 1, PMNI 0
...	...

The specific endpoint that is associated with an error record entry is identified by the node type and node ID. Software can determine the node type and node ID for each endpoint during the discovery process. For more information, see the *Discovery* section in the NI-710AE Technical Reference Manual.

To identify the error record entry for an endpoint, cross-check the endpoint node type and node ID values obtained during discovery against the values for each entry. The FMU stores the node type and node ID values for error record entries in the FMU\_ERR<n>\_MISCO registers.

The contents of each error record entry are held in the following registers, where <n> is the error record entry number:

- FMU\_ERR<n>\_FR, which defines the Reliability, Availability, and Serviceability (RAS) features that are implemented and enabled
- FMU\_ERR<n>\_CTLR, which controls whether error logging is enabled for the record entry and specifies the interrupts that this error record generates
- FMU\_ERR<n>\_STATUS, which contains information related to the recorded error
- FMU\_ERR<n>\_MISCO, which contains information about the endpoint that reported the error

### 3.9.3 Controlling which safety mechanisms report errors to the FMU

NI-710AE provides a configuration register, FMU\_SMEN, that controls the error reporting behavior for each individual safety mechanism. You can use this register to specify which safety mechanisms report errors to the central Fault Management Unit (FMU).

The values of the bits in the EN field of the FMU\_SMEN register determine whether FMU error reporting is enabled or disabled for the safety mechanisms. Each bit of this field corresponds to a

specific safety mechanism. To disable error reporting to the central FMU for a safety mechanism, set the corresponding bit to 0. Set the bit to 1 to enable error reporting for the associated safety mechanism. For more information about which bit controls which safety mechanism, see the *FMU FMU\_SMEN register* section in the NI-710AE Technical Reference Manual.



Once enabled, error reporting for a safety mechanism must not be disabled during mission critical operation. Reprogramming the FMU\_SMEN register during mission critical operation can cause the error reporting mechanism to behave incorrectly.

### 3.9.4 Configuring which errors are critical

Critical errors are generated by Safety Mechanisms (SMs) that are set as critical at build time by using the **Critical Error Vector** configuration parameter. By default, no SM errors are configured as critical.

All SM errors, except for those from the legacy Error Correcting Code (ECC) checker, can be configured to report as critical errors to the Fault Management Unit (FMU). Errors from the legacy ECC checker correspond to bit[14] in the FMU\_SMEN register. For a list of all the SM errors, see the *FMU FMU\_SMEN register* section in the NI-710AE Technical Reference Manual.

### 3.9.5 FMU interrupts

The NI-710AE Fault Management Unit (FMU) implements several types of external interrupt to report errors to the rest of the system. You can enable each type of interrupt signaling individually by programming the FMU configuration registers.

When a NI-710AE safety mechanism detects an error, the error is usually logged in the FMU error record. You can configure the FMU to signal interrupts externally when an error is logged, enabling the FMU to report the error state to the system. To provide this external signaling, the FMU implements the following interrupt types:

#### Error Recovery Interrupt (ERI)

Used to report Uncorrected Errors (UEs) and Deferred Errors (DEs) to the rest of the system. The corresponding signal for this interrupt is the FMU\_ERI\_INT signal. ERIs can be used for error recovery, fault handling, or both.

#### Fault Handling Interrupt (FHI)

Used to report UEs and Corrected Errors (CEs) to the rest of the system. The corresponding signal for this interrupt is the FMU\_FHI\_INT signal. FHIs can be used for fault handling.

#### Critical Error Interrupt (CRI)

Used to report critical error conditions to the rest of the system. The corresponding signal for this interrupt is the FMU\_CRI\_INT signal. CRIs can be used for error recovery.

For more information about these interrupt types, see the [Arm® Architecture Reference Manual for A-profile architecture](#).

When an FMU interrupt is received, interrupt handling software can read various FMU registers for more details about the error. For more information, see the *Handling FMU interrupts* section in the NI-710AE Technical Reference Manual.

The corresponding error interrupt signals are level-sensitive. For more information about the implemented signals, see the *Fault management unit interface signals* section in the NI-710AE Technical Reference Manual.

You can configure the FMU interrupt signaling functionality by programming the FMU FMU\_ERR\_CTLR\_0 register. This register contains controls for enabling error interrupt reporting, and also allows you to enable specific interrupt types for different types of error.

### 3.9.5.1 Handling FMU interrupts

When an Error Recovery Interrupt (ERI) or Fault Handling Interrupt (FHI) is received, interrupt handling software can read various Fault Management Unit (FMU) registers for more information.

Interrupt handling software can read the FMU FMU\_ERRGSR register to identify the error. The [N]th bit in the FMU\_ERRGSR.S field is set to 1 when error record N contains a valid error.

The ID of the safety mechanism that reported the error is specified in the IERR field of the FMU FMU\_ERR\_STATUS register. Where a node has reported more than one error of the same type to an error record, the FMU\_ERR\_STATUS.OF field is set to 1.

When the recovery procedure is complete, the errors from the error record can be acknowledged by writing an appropriate value to the FMU\_ERR\_STATUS register.

### 3.9.6 FMU error injection

To inject an error into a safety mechanism, write to the FMU\_SMINJERR register of the Fault Management Unit (FMU).

The values of the bits in the BLK field of the FMU\_SMINJERR register specify the functional block. The setting of the SMID field bits identifies the safety mechanism into which to inject the error. When you write to the FMU\_SMINJERR register, you inject a single error only. You do not need to clear the error. For more information, see the *FMU FMU\_SMINJERR register* section in the NI-710AE Technical Reference Manual.

### 3.9.7 Software initialization and the FMU

During the discovery process, software must identify which nodes in a NI-710AE configuration can report errors to the Fault Management Unit (FMU).

Software can extract the following information for each node:

- Voltage domain ID (VD\_ID)
- Power domain ID (PD\_ID)
- Clock domain ID (CD\_ID)

- Node type
- Node ID

For more information about mapping and discovery, see the *IDM and device discovery* section in the NI-710AE Technical Reference Manual. For more information about node type and node ID, see the *Node, interface, and transaction identifiers* section in the NI-710AE Technical Reference Manual.

Some nodes that report errors to the FMU, such as Configuration Network Interfaces (CFGNIs), are not explicitly discoverable by the node ID mapping process. However, these nodes are implicit and observe the following rules:

- There is only one FMU
- There is one CFGNI for each power domain
- There is one power controller for each power domain
- There is one clock controller for each clock domain

The number of error records can be used to determine the number of nodes that can report errors to the FMU. That is, the number of error records is the sum of the number of ASNIs, AMNIs, CFGNIs, clock controllers, and power controllers, plus the Performance Monitoring Unit (PMU) and the FMU. Software must use the FMU FMU\_ERRDEVID register to identify the number of error records that are implemented.

Error recovery software can obtain information from each node that can report a functional safety error by reading the FMU FMU\_ERR\_MISC0 register.

## Initialization process

As part of the initialization procedure, software must determine the number of error records and extract information about the nodes that can report errors to the FMU.

The following example shows the pseudocode for the process:

```
num_records = ERRDEVID.NUM
for (i = 0 ; i < num_records; i = i + 1) {

    //read lower 32 bits
    rd_data = apb_read(ERR<i>MISC0);
    NODES[i].NodeID = ERR<i>MISC0[15:0];
    NODES[i].NodeType = ERR<i>MISC0[29:16];

    //read upper 32 bits
    rd_data = apb_read(ERR<i>MISC0 + 4);
    NODES[i].VD_ID = ERR<i>MISC0[29:20];
    NODES[i].PD_ID = ERR<i>MISC0[19:10];
    NODES[i].CD_ID = ERR<i>MISC0[9:0];
}
```

After initialization, software must retain a table for each node similar to the following example. In the table, N/A indicates that the parameter is not applicable to that type of node.

**Table 3-6: Example node information table**

Block	Node type	Node ID	VD_ID	PD_ID	CD_ID
asniO	0x0004	0x0000	0x000	0x000	0x000
hsniO	0x0007	0x0001	0x000	0x000	0x001
amniO	0x0005	0x0000	0x000	0x001	0x000
...					
cfgniO	0x0060	0x0000 (N/A)	0x000	0x000	0x000 (N/A)
cfgni1	0x0060	0x0000 (N/A)	0x000	0x001	0x000 (N/A)
clkctrlO	0x0040	0x0000 (N/A)	0x000	0x000	0x000
clkctrl1	0x0040	0x0000 (N/A)	0x000	0x000	0x001
pwrctrlO	0x0041	0x0000 (N/A)	0x000	0x000	0x000 (N/A)
pmu	0x0006	0x0000 (N/A)	0x000	0x000	0x000
fmu	0x0061	0x0000 (N/A)	0x000 (N/A)	0x000 (N/A)	0x000 (N/A)

The initialization routine must iterate through each node and enable error reporting for all safety mechanisms. For more information, see [Controlling which safety mechanisms report errors to the FMU](#).

The following example shows the pseudocode for the process.

```

num_mechanisms = 18;
num_records = ERRDEVID.NUM
for (i = 0 ; i < num_records; i = i + 1) {

    FMU_SMINFO.VD_ID    = NODES[i].VD_ID;
    FMU_SMINFO.PD_ID    = NODES[i].PD_ID;
    FMU_SMINFO.CD_ID    = NODES[i].CD_ID;
    FMU_SMINFO.NodeID   = NODES[i].NodeID;
    FMU_SMINFO.NodeType = NODES[i].NodeType;

    apb_write(0xE210,    FMU_SMINFO[31:0]);
    apb_write(0xE210 + 4, FMU_SMINFO[63:32]);

    for (j = 0; j < num_mechanisms; j = j + 1) {
        //perform this sequence for a valid mechanism in the node
        if (notValidSM(j,i) next;

        FMU_SMEN.SMID = j;
        FMU_SMEN.EN = 1;
        apb_write(0xE204, FMU_SMEN);
    }
}

```

### 3.9.8 FMU register protection mechanism

The FMU registers are protected against inadvertent writes by a locking mechanism. To write to an FMU register, the registers must first be unlocked by sending a specific write transaction to the FMU\_KEY register.

After a reset, the FMU registers are in a locked state. When the registers are in this state, writes to any register other than the FMU FMU\_KEY register are ignored.

Before writing to any other FMU register, a write must first be made to the FMU\_KEY register that meets all the following requirements:

- The transaction is a Secure write
- The write is for 32 bits, that is, all write strobes
- The bottom 8 bits are 0xBE

When such a transaction is received, all the FMU registers are unlocked for writes. In this state, the FMU\_KEY register reads as 0x00000BE. The registers are locked again when a Secure write of any width and with any write strobes is made to any FMU register except FMU\_KEY. When the registers are locked, the FMU\_KEY register reads as 0x00000000.

Writing to the FMU\_KEY register when in the unlocked state only leaves the FMU registers unlocked if the write satisfies all the unlocking criteria. Any other type of write to the FMU\_KEY register locks the registers.

Non-secure accesses to FMU registers are always unsuccessful and never affect the locked state of a register.

### Writing to 64-bit FMU registers

Some FMU registers are 64-bit registers, but the FMU APB interface is 32 bits wide, which is the maximum data width for the APB protocol. Therefore, when the FMU registers are unlocked, NI-710AE allows two consecutive writes to update the same 64-bit FMU register without requiring an intermediate unlocking write to FMU\_KEY. Both the transactions must be Secure writes with all write strobes to the same register. However, the writes must target different halves of that register. This behavior is permitted to enable the interconnect to split a single 64-bit register access and present the halves to the FMU in any order.

Using the 64-bit FMU\_ERR\_CTLR\_0 register as an example, the following sequence successfully updates the contents of this register:

1. Secure write to the FMU\_KEY register with data 0xBE and all write strobes asserted.
2. 32-bit Secure write to FMU\_ERR\_CTLR\_0 register bits[63:32] address 0x0C and all write strobes asserted
3. 32-bit Secure write to FMU\_ERR\_CTLR\_0 register bits[31:0] address 0x08 and all write strobes asserted

## 3.9.9 Fault Management Unit resets

To clear the Fault Management Unit (FMU) error records, NI-710AE provides has a dedicated set of reset signals, FMURESETn and FMURESETnCHK.

The FMU reset signals only clear the FMU error records. The functional reset signals, RESETn and RESETnCHK, reset the rest of the FMU functional logic.

The FMU reset signals and the functional reset signals can be used to perform two types of reset operations. For more information, see the following sections:



- [Perform a Cold reset](#)
- [Perform a Warm reset](#)

### 3.9.9.1 Perform a Cold reset

The NI-710AE Cold reset operation clears both the functional state and the Fault Management Unit (FMU) error record by asserting both functional and FMU resets. A Cold reset is typically performed at powerup.

#### About this task

There are some requirements related to FuSa resets which must be followed. These requirements are reflected in the following procedure. For more information, see [FuSa reset requirements](#).

Do not use this procedure if you require the information in the FMU error record to be retained for diagnostic purposes. Instead, see [Perform a Warm reset](#).

#### Procedure

1. Assert the RESETn, RESETnCHK, FMURESETn, and FMURESETnCHK reset signals simultaneously.
2. Keep the reset signals asserted for at least 40 clock cycles.
3. Deassert the reset signals.

### 3.9.9.2 Perform a Warm reset

The NI-710AE Warm reset clears the functional state but preserves the Fault Management Unit (FMU) error record by asserting only the functional resets. This operation is useful for diagnosing FuSa errors because you can reset NI-710AE while preserving FMU error record information for later analysis.

#### About this task

There are some requirements related to FuSa resets which must be followed. These requirements are reflected in the following procedure. For more information, see [FuSa reset requirements](#).

Do not use this procedure to clear both the functional state and the FMU error record. Instead, see [Perform a Cold reset](#).

#### Procedure

1. Assert the RESETn and RESETnCHK reset signals simultaneously.
2. Keep the reset signals asserted for at least 40 clock cycles.
3. Deassert the reset signals.

## 4. Power, clock, and reset management

NI-710AE supports a configurable number of power, voltage, and clock domains, with reset signals for each clock domain. Because NI-710AE is highly flexible, the interconnect can occupy various power states and operating modes.

An external P-Channel controls each power domain and defines the power state into which the power domain can enter. An external Q-Channel connects to each clock domain, and indicates whether the clock can be externally gated.

The following clock, power, and voltage domain restrictions apply to NI-710AE:

- Each clock domain must only be associated with a single power domain
- Each power domain must only be associated with a single voltage domain
- Each power domain must support one or more clock domains
- Each voltage domain must support one or more power domains

If multiple power domains are used, the power domains must be configured at the same level in the domain hierarchy. For more information, see [Power](#).

NI-710AE provides different types of clocks that can be arranged hierarchically to allow for different power scenarios. For more information, see [Clocks](#).

Separate blocks are used for power and clock control. For more information, see [Power control](#) and [Clock and reset control](#), respectively.

### 4.1 Power

NI-710AE supports configuration of multiple power and voltage domains across the design. Each power domain can be separately gated.

Up to 32 separate power domains and up to 32 separate voltage domains can be configured within an NI-710AE design. In designs with multiple power domains, all the power domains must exist at the same level in the domain hierarchy. NI-710AE does not support designs with power domains at different hierarchical levels.

Each power domain can be separately powered down or placed into retention. An external P-Channel LPI requests changes to the power domain state through the Power Domain Controller. For more information, see [P-Channel low-power interface](#).

The following asserted P-Channel PACTIVE bits indicate the minimum power state that the power domain requires to guarantee progress. For more information, see [Power state requirements and characteristics](#).

#### **PACTIVE[16]**

CONFIG. Enables restricted xSNI access for the power domain.

## PACTIVE[8]

ON. Fully powered state for all logic in the power domain.

## PACTIVE[5]

FULL\_RET. Static retention state for all logic within the power domain.

## PACTIVE[0]

OFF. Fully unpowered state for the power domain.

### 4.1.1 Power state requirements and characteristics

NI-710AE has specific signaling requirements for the different power states that are supported. Only specific power state transitions are permitted, which depend on the starting state.

The P-Channel manages the transition between the different power states.

Out of reset, the PSTATE that is presented to NI-710AE must be one of the supported values in the following table. If any other value is presented, the behavior is **UNPREDICTABLE**. The highest of the asserted P-Channel PACTIVE bits indicates the minimum power state that the power domain requires to guarantee progress.

**Table 4-1: Valid power states for power domains and the requirements of those power states**

Power state	DEVPACTIVE bit	PSTATE[7:4]	PSTATE[3:0]
CONFIG	Bit[16]	0b0001	0b1000
ON	Bit[8]	0b0000	0b1000
FULL_RET	Bit[5]	0b0000	0b0101
OFF	Bit[0]	0b0000	0b0000

#### CONFIG power state

The CONFIG state restricts access to xSNIs in the power domain. In this state, only xSNIs with <INTF>\_CONFIG\_ACCESS set HIGH at their reset input pins permit ingress of external transactions.

When PACTIVE[16] is HIGH, the CONFIG power state is the lowest power state that is required for the system. For example, this scenario can occur when the only transaction that requires access to fully powered logic is from a CONFIG-defined interface.

If PACTIVE[16] is LOW, then PACTIVE[8] must be checked to determine the required power state. In this case, PACTIVE[8] determines whether the system must transition to ON or whether the system can enter the FULL\_RET or OFF states to save power.

State transitions from CONFIG to ON, FULL\_RET, or OFF are permitted. The highest PACTIVE bit that is HIGH determines the transition.

#### ON power state

ON is the fully powered state for all logic in the power domain. The power domain must be in the ON state for all interfaces to progress.

When PACTIVE[8] is HIGH, the ON power state is required, such as when a transaction requires access to fully powered logic.

If PACTIVE[8] is LOW, then it might be possible to transition to the FULL\_RET state to save power.

State transitions from ON to CONFIG, FULL\_RET, or OFF are permitted. The lowest PACTIVE bit that is HIGH determines the transition. However, we only recommend transitioning to CONFIG if system reconfiguration is required. Otherwise, we recommend a transition to OFF.

### FULL\_RET power state

FULL\_RET is the static retention state for all logic instances within the power domain. In the FULL\_RET state, all external flow control signals are held in a state that prevents propagation of any transactions.

State transitions from FULL\_RET to ON or CONFIG are permitted. Transitioning from the FULL\_RET state to the OFF state is not permitted.

### OFF power state

OFF is the fully off state for the power domain. In the OFF state, all external flow control signals are held in a state that prevents propagation of any transactions.

State transitions from OFF to ON or CONFIG are permitted. Transitioning from the OFF state to the FULL\_RET state is not permitted.

## 4.1.2 P-Channel low-power interface

Each power domain in NI-710AE is connected to a standard P-Channel LPI that communicates external power state information. The P-Channel that is connected to each power domain determines whether the interconnect can be powered off or placed into retention.

The PACTIVE signal indicates the highest permitted power state of the power domain.

The P-Channel uses the following LPI signals to indicate the external power state and to specify the power state into which NI-710AE is required to transition.

**Table 4-2: P-Channel LPI signals**

Name	Direction	Purpose
PACTIVE[16:0]	Output	Vector indicator of the power states that NI-710AE is eligible to enter
PSTATE[7:0]	Input	Binary value of the power state into which an external requester requires NI-710AE to transition
PREQ	Input	Request signal to initiate a power state transition
PACCEPT	Output	Handshake signal to indicate that the power state transition is complete
PDENY	Output	Handshake signal to indicate that the power state transition cannot be completed

## 4.2 Clocks

NI-710AE provides configurable clock domains and supports hierarchical clock gating.

You can configure up to 32 separate clock domains within your NI-710AE design, which can be arranged in a hierarchy. For more information, see [Levels of clock gating](#).

Each of the configured clock domains can be separately gated. For more information, see [Hierarchical clock gating](#).

The clock domains are gated by Q-Channel LPIs. For more information, see [Q-Channel low-power interface](#).

The clock gating process is managed by the External Clock Controller. For more information, see [External clock controller](#).

NI-710AE requires that connected interfaces support a specific wake up signal. When this signal is asserted, NI-710AE requests activation of the relevant clock domain to ensure that the appropriate system components are ready to receive transactions. For more information, see [Clock domain wake up](#).

The NI-710AE network can be configured as a bridge that crosses between different clock frequencies. For more information, see [Network FIFO and clocking function](#).

Every clock domain has a single clock pin input, which is labeled <CLKNAME>\_CLK.

### 4.2.1 Levels of clock gating

NI-710AE contains different clock types that are arranged in a hierarchy. This hierarchy includes clocks supplying clock domains through to local clocks that are created by the RTL.

The following clock types are included in NI-710AE:

#### Top-level clock

The clock input to the clock domain <CLKNAME>\_CLK.

#### Regional clocks

Created as an output of regional clock gaters that include a coarse enable for coarse-grained clock gating under idle or mostly idle conditions. Regional clock gaters can shutdown the clock network between regional and local gaters. Therefore, this level of hierarchy enables greater power reduction than is possible using local clock gating. The regional clock gaters are instantiated in and controlled by the NI-710AE RTL.

#### Local clocks

Created according to the following hierarchy:

1. RTL creates fine-grained enable signals.
2. Fine-grained enable signals control local clock gaters.
3. Local clock gaters output local clock signals.

Local clock signals are used to clock sequential elements directly in NI-710AE. The exact set of local clocks is internal to NI-710AE and is not described here.

## 4.2.2 Hierarchical clock gating

NI-710AE supports hierarchical clock gating. During periods of low activity, the system can use hierarchical clock gating to transition to a low-power state.

Transitioning to a low-power state enables the system to save the power that the active clock tree would normally consume. Control over individual clock domains allows for flexible system design and therefore flexible power state design.

Hierarchical clock gating can gate the following regions:

- xSNIs, for example ASNIs
- xMNIs, for example AMNIs
- Routers
- PCDC blocks
- SERDES blocks
- Register blocks

The Q-Channel LPI enables hierarchical clock gating by communicating with the clock domain controller to request that the clock domain becomes quiescent. External clock controllers can use the Q-Channel LPI to request gating of individual clock domains in the interconnect.

On receipt of a request, the interconnect waits until there are no outstanding transactions within the clock domain and then blocks new transactions from entering. When this process is complete, the clock domain sends an acknowledgment to indicate that the clock controller can remove the clock.

## 4.2.3 Q-Channel low-power interface

Each clock domain in NI-710AE is connected to a standard Q-Channel LPI that gates the clock domain. A Q-Channel is present for every clock domain.

The Q-Channel LPI contains signals to control hierarchical clock gating in NI-710AE. Hierarchical clock gating is always present in the NI-710AE configuration. For more information on the function of the Q-Channel LPI signals, see [AMBA® Low Power Interface Specification](#).

**Table 4-3: Q-Channel LPI signals**

Signal	Direction	Description	Source	Destination
QACTIVE	Output, input	Interconnect active	Interconnect	Controller
QREQn	Output, input	System low-power request	Controller	Interconnect

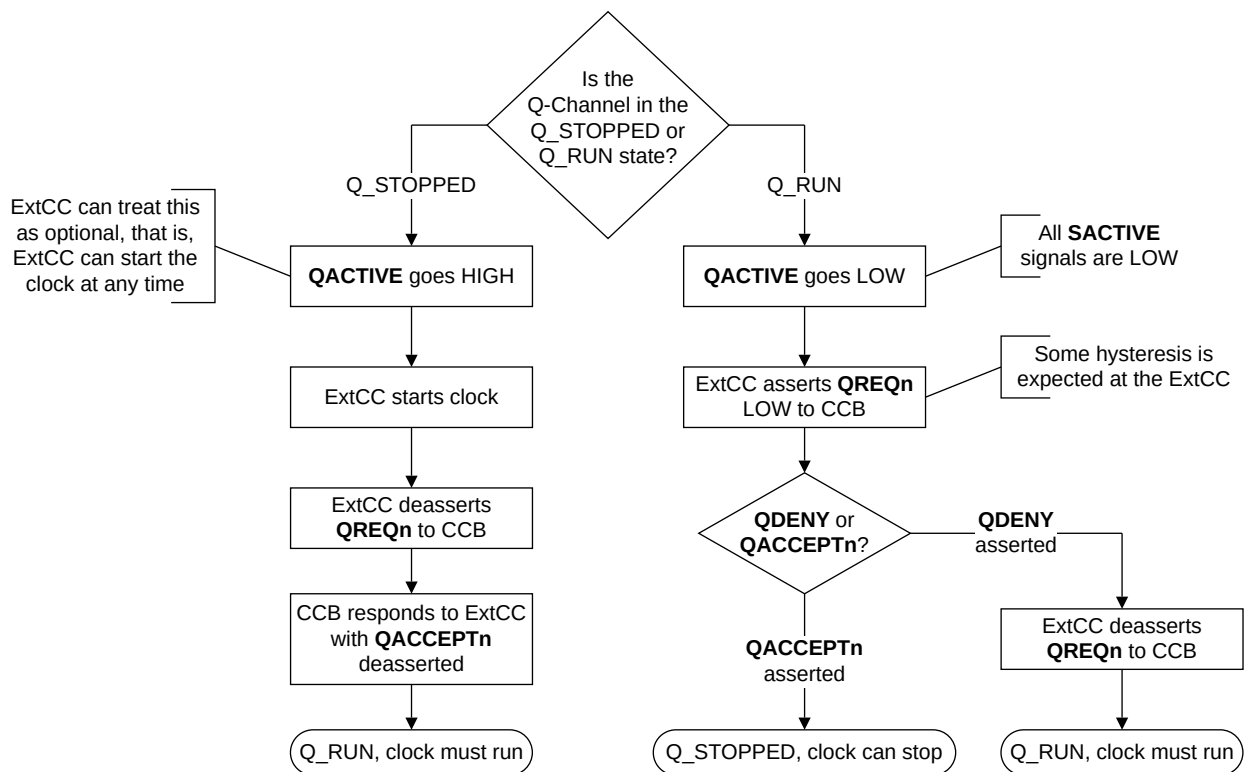
Signal	Direction	Description	Source	Destination
QACCEPTn	Output, input	Low-power request acknowledgment	Interconnect	Controller
QDENY	Output, input	Negative acknowledgment after receiving a QREQn assertion, indicating NI-710AE has refused the request from the controller to prepare to stop the clocks	Interconnect	Controller

## 4.2.4 External clock controller

The external clock controller controls the clock gating flow.

The following figure shows an example clock gating flow and how the external clock controller controls that flow.

**Figure 4-1: Example External Clock Controller (ExtCC) clock gating control flow**



This example clock gating sequence begins and ends with the Q-Channel in either of the following states:

### Q\_STOPPED

Quiescent state, where QREQn and QACCEPTn are asserted.

### Q\_RUN

Active state, where QREQn and QACCEPTn are deasserted.

The following requirements apply to the external clock controller:

- The external clock controller must supply a clock to NI-710AE when the Q-Channel is in any state other than Q\_STOPPED.
- The external clock controller can either:
  - Choose to gate the clock to NI-710AE when the Q-Channel is in the Q\_STOPPED state.
  - Choose to run the clock at any time.
- The external clock controller is responsible for bringing the Q-Channel to the Q\_RUN state after reset deassertion.
- The exact behavior of the external clock controller and its usage of QREQn in response to QACTIVE deassertion is not described here. However, the design of the external clock controller is likely to include a control loop with some hysteresis. This feature ensures that hierarchical clock gating is enabled when the system is inactive for long periods. Hierarchical clock gating is not enabled for short periods of inactivity. If the clocks are stopped in response to short periods of inactivity, the performance of NI-710AE can be negatively affected.
- It is the responsibility of the SoC designer to fully control the clock management Q-Channel. If a control or configuration bit is required to completely enable or disable hierarchical clock gating, that register or bit must exist outside of NI-710AE. There is no internal means of disabling hierarchical clock gating in NI-710AE.

## 4.2.5 Clock domain wake up

Wake up signals are present on the requester device side of the ASNI and the completer device side of the AMNI. These signals indicate incoming or outgoing network traffic, so that the relevant system components are activated and available to receive traffic.

NI-710AE requires that upstream requesters support AWAKEUP when connecting those requesters to the interconnect. Similarly, NI-710AE drives AWAKEUP from the AXI requester interfaces. If AXI4 requesters support AWAKEUP, they can connect to NI-710AE.

Each ASNI has an input signal, AWAKEUP, that must be asserted when the AXI or ACE-Lite AxVALID signal is HIGH. AWAKEUP must remain asserted until the associated ARVALID-ARREADY handshake, or the AWVALID-AWREADY handshake completes. When the address handshake is completed, NI-710AE keeps the clock active until the transaction completes. When AWAKEUP is asserted, NI-710AE drives the QACTIVE signal of the corresponding clock domain HIGH to request activation of the clock signal. For more information, see the *WAKEUP signals* section of the NI-710AE Configuration and Integration Manual.

## 4.2.6 Network FIFO and clocking function

If you configure the network as a clock frequency crossing bridge, then non-blocking Resource Plane (RP) FIFO functions are also configured.

You can configure the FIFO to implement both buffering and clock domain crossing functionality. You can define the FIFO as:

- SYNC 1:1, see [Clock synchronization modes](#).
- SYNC 1:n, see [Clock synchronization modes](#).



- SYNC m:1, see [Clock synchronization modes](#).
- ASYNC, see [Clock synchronization modes](#).

You can configure the depth value of the FIFO to be 1–8.



You can configure the buffering for multiple flits even if you are using a 1:1 clocking ratio.

All clock boundary crossings are implemented using a FIFO structure with appropriate synchronization for the mode of operation.

#### 4.2.6.1 Clock synchronization modes

Socrates automatically calculates the mode of synchronization in accordance with the clock relationships that are defined at design entry.

The following options are available:

##### **Asynchronous**

Select asynchronous if the two clocks bear no relationship to one another.

##### **Synchronous (1:1)**

Select synchronous (1:1) if the two clocks are the same.

##### **Synchronous (1:N)**

Select synchronous (1:N) if both of the following are true:

- The first clock has a lower frequency than the second clock.
- The positive edge of the first clock always coincides with a positive edge of the second clock.

##### **Synchronous (M:1)**

Select synchronous (M:1) if both of the following are true:

- The first clock has a higher frequency than the second clock.
- The positive edge of the second clock always coincides with a positive edge of the first clock.

## 4.3 Power control

The NI-710AE power control network consists of power control blocks, clock control blocks, and several power control signals.

An NI-710AE power controller must be in a power domain that is Relatively Always ON (RAON) compared to the power domain that it manages. This requirement enables assertion of the internal wake up signal and the PACTIVE signal on the external P-Channel as they are in the \*\_RAON

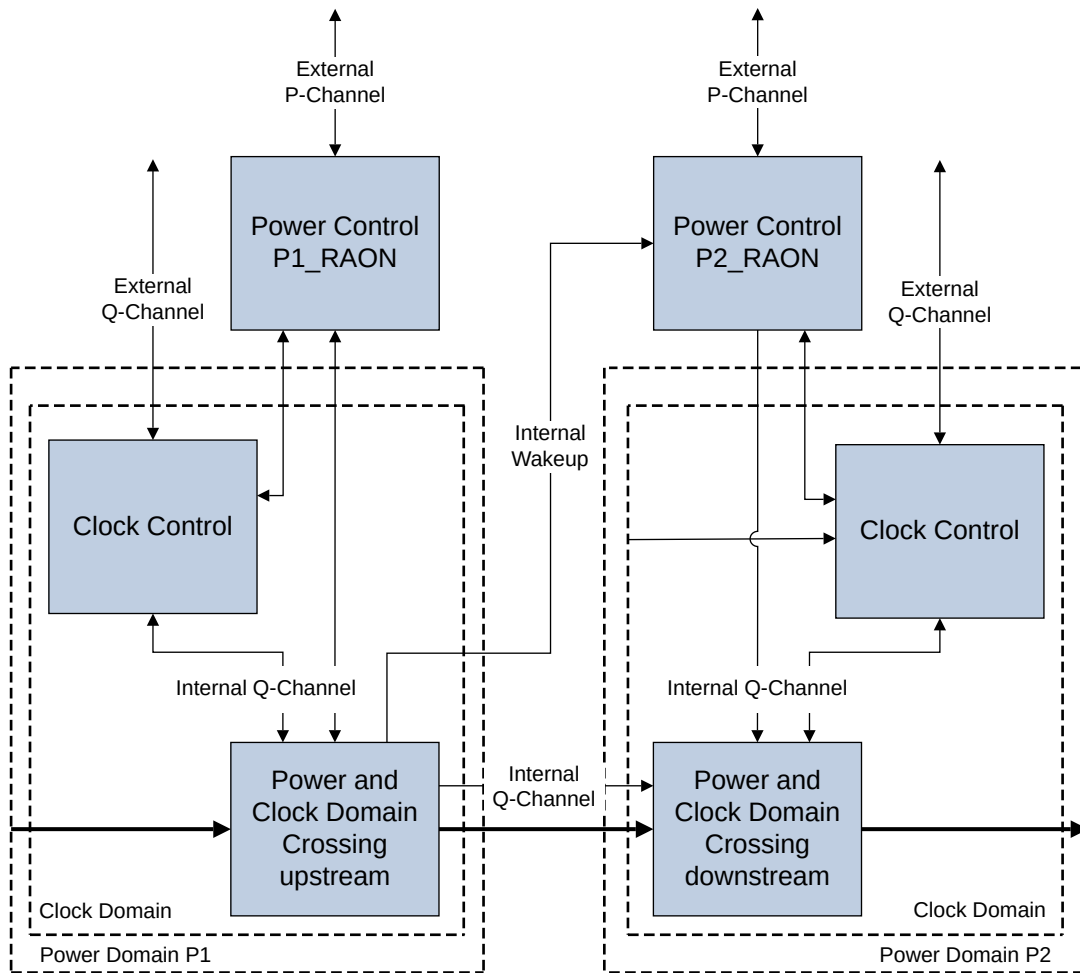
power domain. When asserted, these signals indicate that the corresponding power domain must be turned ON.

In the following diagram, P1\_RAON and P1, and P2\_RAON and P2 are corresponding power domains. For example, if P2\_RAON asserts the external PACTIVE signal, then the SoC power controller is expected to turn on the power for the P2 power domain. Similarly, before requesting the power state transition through the \*\_AON power controller, the SoC power controller must also ensure that the corresponding P1 or P2 power domain is already ON.

The clock and reset to the power controller comes from the clock controller in the corresponding power domain. For example, P1 could contain multiple clock domains. However, the power controller in P1\_RAON is considered to be in the same clock domain as one of the clock domains in P1. Therefore, the clock and reset to the P1\_RAON power controller comes from the corresponding clock controller in the same clock domain in P1. Before P1 or P2 powers down, the signals crossing between each power domain and the corresponding \*\_RAON power domain are all isolated.

The following figure shows the various elements in the power control network for NI-710AE.

**Figure 4-2: NI-710AE power control network**



Specific power control steps are required to enable managed power domains to move between the ON and OFF states. For more information, see [Power control sequences](#).

NI-710AE includes a feature that enables attached devices to transition between power states while the interconnect remains powered up. For more information, see [External power domain boundaries](#).

HSNIs include logic to ensure that the AHB address phase can be sampled even when the unit is clock gated. For more information, see [AHB address phase buffering in HSNIs](#).

### 4.3.1 Power control sequences

The NI-710AE power control network must perform specific sequences of actions to allow downstream power domains to transition between power states.

The following sections list the steps involved in ON to OFF and OFF to ON power transitions and the order in which they must be performed.

#### Upstream power domain ON, downstream power domain ON→OFF

The following sequence describes how a downstream power domain transitions from ON to OFF when the upstream power domain is ON.

1. The downstream external PACTIVE[16:1] signal is driven LOW, indicating that all activity within the power domain is complete.
2. The external P-Channel requests that the downstream power domain enters the P\_OFF state.
3. The internal power QREQn signal, which targets the downstream PCDC, goes LOW.
4. If there is no activity in the downstream PCDC:
  - a. The downstream PCDC performs logical isolation of the boundary and indicates to the upstream PCDC a requirement to enter the P\_OFF state.
  - b. The upstream PCDC acknowledges the P\_OFF state request from downstream PCDC, performs logical isolation, and resets the PCDC FIFO pointers to the reset value.
  - c. The downstream PCDC receives the acknowledgment from the upstream PCDC, resets the PCDC FIFO pointers to the reset value, and issues QACCEPT to the power control block.
  - d. The power control block issues a P-Channel accept to the external interface.
5. The external clock controller requests that all clock Q-Channels enter the Q\_STOPPED state.
6. When all P-Channels and Q-Channels are in the P\_OFF or Q\_STOPPED states, all power domain pins are physically isolated, if necessary.

In NI-710AE, all isolation values are inactive values of the corresponding signals. That is, 0 for active-HIGH polarity and 1 for active-LOW polarity.

#### Upstream power domain ON, downstream power domain OFF→ON

The following sequence describes how a downstream power domain transitions from OFF to ON when the upstream power domain is ON.

1. A new upstream transaction arrives in the CDC.
2. The upstream PCDC, in the RAON domain, asserts an internal wake up signal to the downstream power controller.
3. The downstream power controller asserts the external higher power state PACTIVE asynchronously.
4. The external power control:
  - a. Restores power to the downstream power domain.
  - b. Performs a Cold reset on the domain using the general reset and Fault Management Unit (FMU) reset signals. For more information, see [Perform a Cold reset](#).

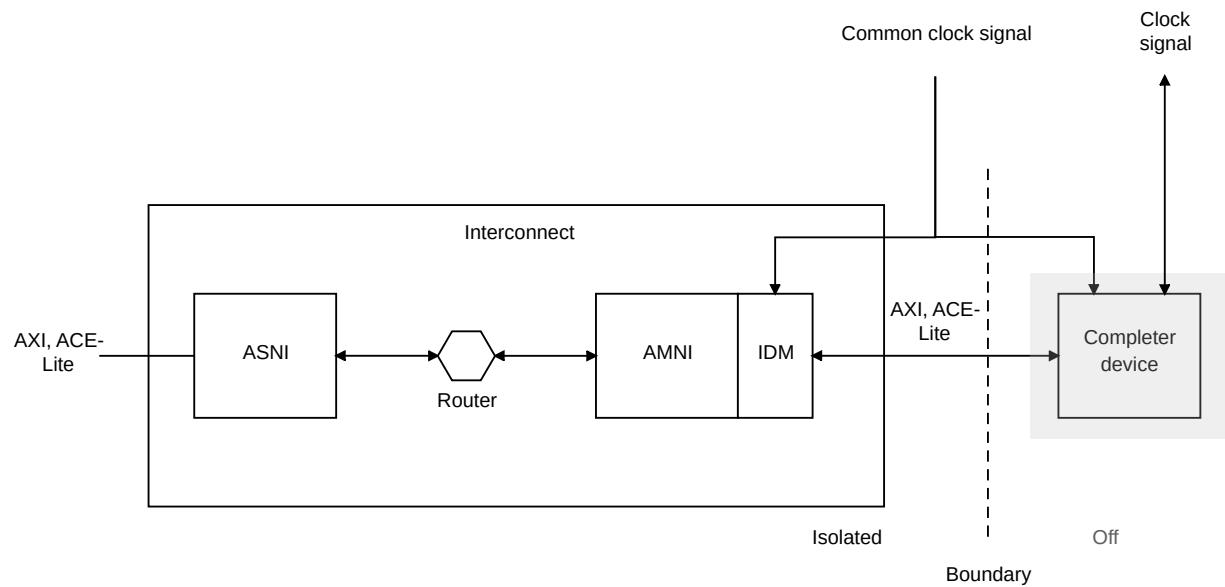
- c. Removes physical isolation.
  - d. Removes resets to the domain.
5. The external P-Channel requests to enter the P\_ON state and the clock Q-Channel requests to enter Q\_ON.
6. The internal QREQn signal, which targets the downstream PCDC, goes HIGH.
- a. The downstream PCDC removes logical isolation of the boundary and issues a QACCEPTn transition to the clock control and power control blocks.
  - b. The clock control and power control blocks forward the QACCEPTn transition to the external interface.
  - c. The downstream PCDC indicates to the upstream PCDC that power is restored and that the downstream PCDC is in the P\_ON state.
  - d. The upstream PCDC acknowledges the downstream PCDC and removes logical isolation.

### 4.3.2 External power domain boundaries

External power domain boundaries are used in NI-710AE to enable attached devices to switch power state independently of the interconnect.

NI-710AE provides power isolation on AXI signals at the boundary of the interconnect and integrated IP. This feature can be used when the attached IP is in a switchable power domain, and the interconnect must be in a RAON power domain. For example, power isolation can be applied at the interconnect boundary between an AMNI and its attached AXI completer device, as the following figure shows:

**Figure 4-3: NI-710AE external power domain boundary**



When applying this feature, the interconnect must use IDM isolation to prevent cross-boundary accesses. For more information, see [IDM access control](#).

The clock domain crossing is within the IP block. Arm assumes that the same clock feeds the interconnect network interface and the IP interface.

When the interconnect is powered up, the IDM functionality is in the isolate state, and the attached device is OFF. At this point, software can still access enumeration values in IDM registers. For more information, see [IDM and device discovery](#).

A specific sequence of events must occur in the system to power up or power down a device in an external power domain. The following sections list the steps involved and the order in which they must occur.

### External power domain powerup sequence

The following events must occur in the system to power up a device in an external power domain.

1. The system applies power to the IP domain.
2. The system removes isolation cell clamp values on the AXI boundary.
3. The system applies the IP reset sequence, either through a full system reset or by an IDM soft reset.

For more information about the IDM soft reset feature, see [IDM soft reset mode](#).

4. The system releases IDM isolation.
5. Configuration or mission access to the IP occurs.

## External power domain power down sequence

The following events must occur in the system to power down a device in an external power domain.

1. The IDM functionality is placed into the isolation state.
2. The system applies isolation cell clamp values on the IP boundary.
3. The system removes the power to the IP power domain.

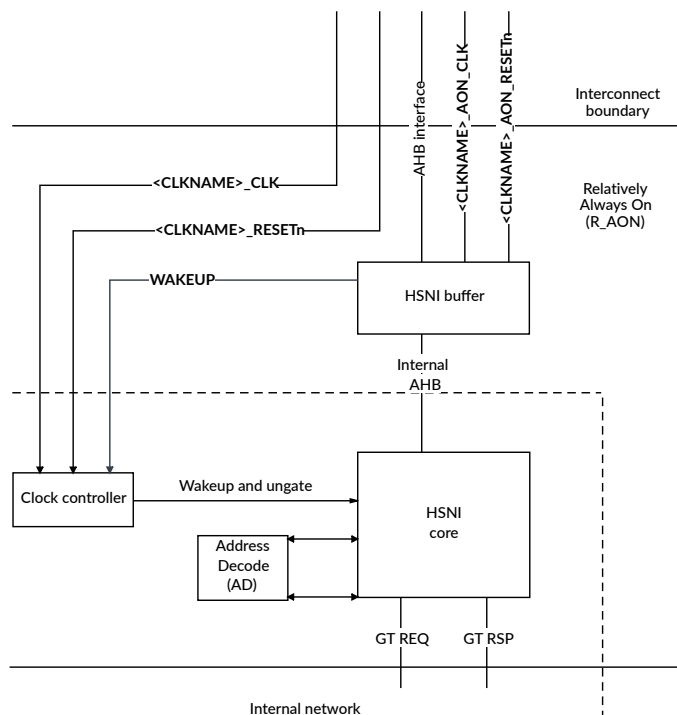
### 4.3.3 AHB address phase buffering in HSNIs

Extra buffering logic and signals in NI-710AE HSNIs enable AHB address phase sampling when the unit is clock gated.

In the AHB protocol, a completer cannot request that the address phase of a transaction is extended. Therefore, all HSNIs must be able to sample the address phase, even when clock gated. The HSNi block adds an extra buffer stage to accept the address phase of a transaction when the HSNi is clock gated.

The following figure shows how the HSNi buffer works.

**Figure 4-4: HSNi clock gating buffer mechanism**



The standard <CLKNAME>\_CLK and <CLKNAME>\_RESETn signals behave normally and connect to the clock and power architecture. These signals must follow the same rules that are described in [External clock controller](#). So, the clock input can only be removed when the Q-Channel is in the Q\_STOPPED state.

NI-710AE adds extra <CLKNAME>\_AON\_CLK and <CLKNAME>\_AON\_RESETn signals for the buffer stage. These signals must be ON before an initial transaction enters the device. If the network does not follow this constraint, the transaction is lost. The wake up signal is routed to the clock controller of the respective clock domain. The clock controller can then wake up and ungate the core component so that the HSNI can start to accept transactions.

The clock for the HSNI buffer and the HSNI core must be driven from the same source clock. There is no synchronization and the buffer and core are assumed to be in the same clock domain. If the buffer and core are not in the same clock domain, then transactions are likely to be lost.

The AHB requester, HSNI buffer, and HSNI core must all be in the same power domain. This arrangement provides improved power saving. When the AHB requester and HSNI buffer are powered OFF, the HSNI core is also powered OFF, which saves power.

## 4.4 Clock and reset control

The NI-710AE clock and reset control network consists of clock control blocks and several clock control signals.

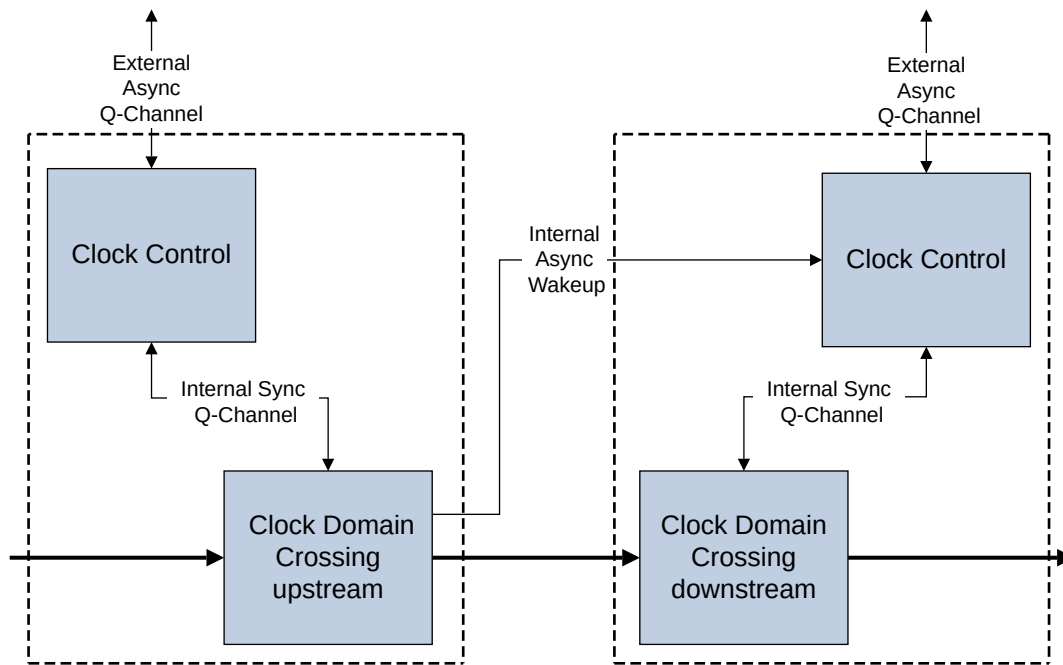
NI-710AE contains one external Q-Channel and reset signal for each clock domain. When the Q-Channel is in the Q\_STOPPED state, there is logical isolation between clock domains, and all transactions are stalled at the domain boundary.

Clock domains exit reset in the Q\_STOPPED state when the domains are logically isolated. Therefore, requests cannot be lost. The full Q-Channel sequence transitioning from Q\_STOPPED to Q\_RUN must be completed before requests can enter a clock domain. All clock domains within a single power domain must be reset together.

The following figure shows an example clock and reset control network within the interconnect.



**Figure 4-5: Example NI-710AE clock and reset control network**



Specific clock control steps are required to enable managed clock domains to move between the ON and OFF states, and to exit from the reset state. For more information, see [Clock control sequences](#) and [Reset control sequences](#).

#### 4.4.1 Clock control sequences

The NI-710AE clock control network must perform specific sequences of actions to allow downstream clock domains to transition between states.

The following sections list the steps involved in ON to OFF and OFF to ON clock transitions and the order in which they must be performed.



NI-710AE does not deny requests to enter a higher clock state, such as a transition from OFF to ON.

##### Upstream clock domain ON, downstream clock domain ON→OFF

The following sequence describes how a downstream clock domain transitions from ON to OFF when the upstream clock domain is ON.

1. The downstream external QACTIVE signal is driven LOW, indicating that all activity within the clock domain is complete.
2. The external QREQn signal is driven LOW by the external clock controller.

3. The internal QREQn signal to the CDC goes LOW.
4. If there is any activity or no activity in the CDC:
  - There is activity within the CDC:
    - a. The CDC asserts the internal QACTIVE signal.
    - b. The CDC issues the internal QDENY signal.
    - c. The top-level Q-Channel sends an external QDENY handshake.
    - d. The external clock controller must complete the Q-Channel QDENY by reasserting QREQn.
  - There is no activity in the CDC:
    - a. The CDC performs logical isolation of the boundary and issues the QACCEPTn signal to the clock control block.
    - b. The clock controller forwards the QACCEPTn signal to the external interface.
    - c. The clock is gated externally, if necessary.

#### **Upstream clock domain ON, downstream clock domain OFF→ON**

The following sequence describes how a downstream clock domain transitions from OFF to ON when the upstream clock domain is ON.

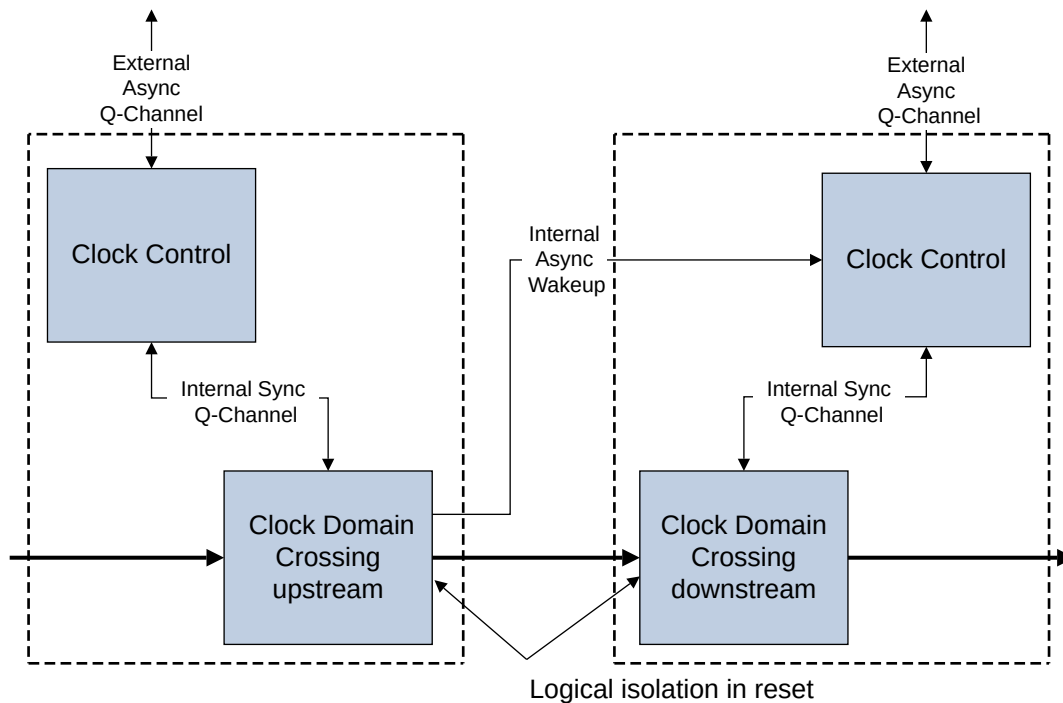
1. A new upstream transaction arrives in the CDC.
2. The upstream CDC asserts an internal wake up signal to the downstream clock controller.
3. The downstream clock controller asserts the external QACTIVE signal asynchronously.
4. The clock signal is restored externally to the downstream clock domain.
5. The external QREQn signal is driven HIGH by the external clock controller.
6. The internal QREQn signal to the CDC goes HIGH.
  - a. The CDC removes logical isolation of the clock domain boundary and issues a QACCEPTn transition to the clock controller.
  - b. The clock controller forwards the QACCEPTn transition to the external interface.

#### **4.4.2 Reset control sequences**

A specific sequence of actions must occur to permit a clock domain to exit from the reset state. The sequence differs depending on whether the upstream or downstream clock domain exits reset first.

The following figure shows the logical isolation between clock domains in reset within an example clock and reset control network.

**Figure 4-6: Logical isolation between example clock domains in reset**



### Both domains in reset state, upstream domain exits reset first

The following sequence describes how an upstream clock domain transitions out of reset when both clock domains are in reset.

1. The upstream clock domain completes a clock and power handshake to permit operation.
2. A new transaction arrives at the upstream CDC.
3. The downstream clock domain is in reset (Q\_STOPPED state). So, it now follows the same flow as when the upstream clock domain is ON and the downstream clock domain transitions from OFF to ON. For more information, see [Clock control sequences](#). However, the downstream clock domain must first exit reset.

### Both domains in reset state, downstream domain exits reset first

The following sequence describes how a downstream clock domain transitions out of reset when both clock domains are in reset.

1. The downstream clock domain completes a clock and power handshake to permit operation.
2. The downstream clock domain now functions as if in normal operation. It awaits transactions, which can be forwarded after the upstream clock domain exits reset and completes the external clock and power handshake. The upstream clock domain does not issue transactions until it is out of reset.

## 5. Node, interface, and transaction identifiers

NI-710AE uses various types of identifiers to identify domains, components, interfaces, and transactions in the interconnect. Each type of NI-710AE identifier has a specific function.

### Node ID

When you build an NI-710AE configuration, NI-710AE assigns each xSNI and xMNI node a node ID. It also defines node IDs for the voltage, power, and clock domains, and the Performance Monitoring Units (PMUs). The software discovery process uses node IDs to detect the programming region for each node.

The node ID of an xSNI is unique within the group of all xSNIs, and similarly no two xMNIs can have the same node ID. However, each node is also identified by its node type, so the node ID spaces of xSNIs and xMNIs might overlap.

For more information about calculating node IDs, see [Node ID calculation](#).

### Interface ID

NI-710AE assigns each external interface in your configuration a unique interface ID. Generic Transport (GT) packets include a Target ID (TgtID) that indicates the target interface for the packet and a Source ID (SrcID) that indicates the source interface. NI-710AE populates interface IDs in these fields to ensure that each packet is routed to the correct external interface destination or Configuration Network Interface (CFGNI).

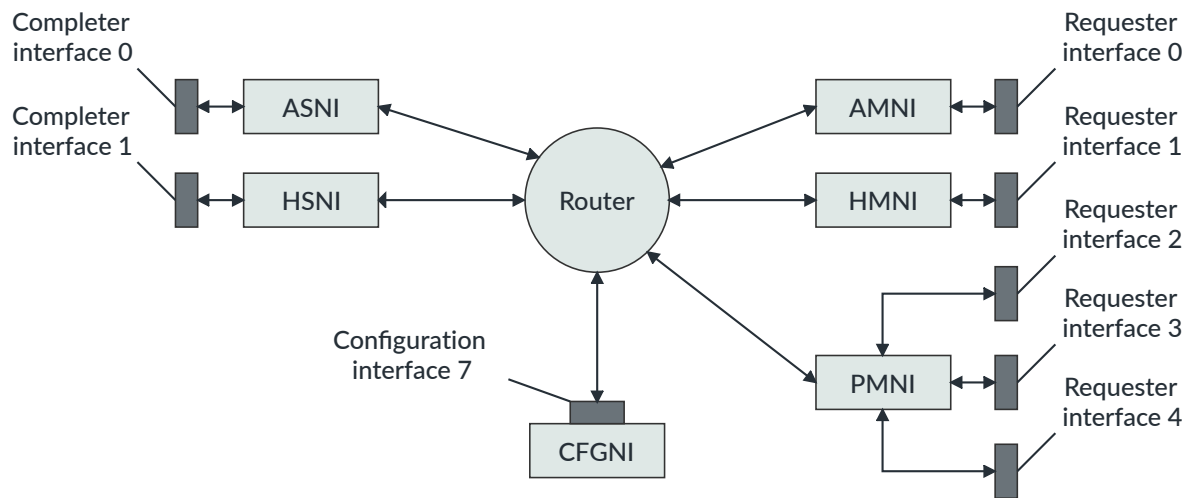
Interface ID values are assigned in two unique pools:

- Completer interfaces, on xSNIs
- Requester interfaces, on xMNIs

NI-710AE also assigns an interface ID to the CFGNI, for use in the address map.

The following diagram shows example interface IDs in NI-710AE and is for illustrative purposes only.

**Figure 5-1: Example unique interface IDs for requester and completer interfaces**



For more information about calculating interface IDs, see [Interface ID calculation](#).

### Input transaction ID

The requesters connected to NI-710AE provide transaction identifiers for each transaction they send into the interconnect. For ASNIs, the value of the AxID input signal provides the input transaction ID, whereas for HSNIs, the value of the HMASTER input signal provides this ID. NI-710AE uses the input transaction IDs when determining the ordering requirements for transactions.

### Internal transaction ID

Each NI-710AE GT packet has an internal transaction ID, which takes the value of the input transaction ID.

### Output transaction ID

When NI-710AE sends AXI and AHB transactions to the downstream completer, those transactions must have an output AxID or HMASTER signal value. AMNIs and HMNIs derive the output ID to pass downstream by concatenating the xSNI input ID and the SrcID width. If the maximum width is not required to represent all the possible SrcID values, AMNIs and HMNIs can also reduce the output ID width. For more information, see [Output ID calculation](#).

## 5.1 Node ID calculation

The method of calculating a node ID differs depending on whether the node ID is for an endpoint, domain, or Performance Monitoring Unit (PMU). The NI-710AE Access Protection Units (APUs) and Fault Management Unit (FMU) do not have node ID values, so there is no calculation for them.

### Calculating endpoint node IDs

Each endpoint in NI-710AE has its own node ID. NI-710AE assigns endpoint node IDs in two pools, one for the xSNIs and one for the xMNIs. To calculate the node IDs for the specific endpoints in your configuration:

- List in alphabetical order all ASNIs and then all HSNIs in your configuration. NI-710AE assigns node IDs to these nodes sequentially from 0 to x, where  $x = \text{number of xSNIs} - 1$ .
- List in alphabetical order all AMNIs, then all HMNIs, and finally all PMNIs in your configuration. NI-710AE assigns node IDs to these nodes sequentially from 0 to y, where  $y = \text{number of xMNIs} - 1$ .

For example, consider a configuration with two ASNIs, an HSNi, four AMNIs, an HMNI, and a PMNI. NI-710AE assigns node IDs for this configuration in the following way:

- The two ASNIs get xSNI node IDs 0 and 1.
- The HSNi gets xSNI node ID 2.
- The four AMNIs get xMNI node IDs 0–3.
- The HMNI gets xMNI node ID 4.
- The PMNI gets xMNI node ID 5.

It is likely that the two node ID pools overlap. If an xSNI and xMNI have the same node ID in your configuration, you can also use the node type to distinguish them.

### Calculating domain node IDs

Each voltage, power, and clock domain in NI-710AE has its own node ID. NI-710AE assigns node IDs for domains in three pools, one for the voltage domains, one for the power domains, and one for the clock domains. To calculate the node IDs for the specific domains in your configuration:

- List in alphabetical order all voltage domains in your configuration. NI-710AE assigns node IDs to these domains sequentially from 0 to x, where  $x = \text{number of voltage domains} - 1$ .
- List in alphabetical order all power domains in your configuration in order of the voltage domains to which they belong. NI-710AE assigns node IDs to these domains sequentially from 0 to y, where  $y = \text{number of power domains} - 1$ .
- List in alphabetical order all clock domains in your configuration in order of the power domains to which they belong. NI-710AE assigns node IDs to these domains sequentially from 0 to z, where  $z = \text{number of clock domains} - 1$ .

For example, consider a configuration with one voltage domain, two power domains, and five clock domains. The first power domain contains three clock domains, and the second power domain contains two clock domains. NI-710AE assigns node IDs for this configuration in the following way:

- The voltage domain gets voltage domain node ID 0.
- The power domains get power domain node IDs 0 and 1.
- The three clock domains in the first power domain get clock domain node IDs 0–2.
- The two clock domains in the second power domain get clock domain node IDs 3 and 4.

### Calculating Performance Monitoring Unit node IDs

In NI-710AE, there is a single PMU for each clock domain. To calculate node IDs for the PMUs in your configuration:

1. Count the number of CFGNIs in your configuration. For each power domain, add the number of endpoints in that power domain to the number of CFGNIs. This value gives the starting node ID for the PMUs in that power domain. NI-710AE assigns each PMU in the power domain subsequent node IDs from the starting node ID.
2. Repeat the preceding step for each power domain.

For example, consider a configuration with two power domains, and each power domain contains two clock domains. Each clock domain contains two endpoints, meaning that there are four endpoints in each power domain. NI-710AE assigns PMU node IDs for this configuration in the following way:

- The starting PMU node ID for both power domains is 6, because there are two CFGNIs in the configuration and four endpoints for each power domain.
- The first PMU in each power domain gets PMU node ID 6, and the second PMU in each power domain gets PMU node ID 7.

## 5.2 Interface ID calculation

The method of calculating an interface ID depends on whether the interface ID is for an external interface or for the Configuration Network Interface (CFGNI).

### Interface ID calculation for external interfaces

Each external interface in NI-710AE has its own interface ID. NI-710AE assigns interface IDs in two pools, one for completer interfaces and one for requester interfaces. PMNIs can have multiple external APB requester interfaces, so the number of APB interfaces for each PMNI affects the assignment of IDs. To calculate the interface IDs for the interfaces in your configuration:

- List in alphabetical order all the ASNIs and then all the HSNIs in your configuration. NI-710AE assigns interface IDs to the external interfaces on these endpoints from 0 to x, where  $x = \text{number of xSNIs} - 1$ .
- List in alphabetical order all the AMNIs, then all the HMNIs, and then all the PMNIs in your configuration. Next, for each PMNI in turn, list all the APB interfaces in alphabetical order. NI-710AE assigns interface IDs to the external interfaces on the AMNIs, HMNIs, and PMNIs sequentially from 0 to y, where  $y = (\text{number of AMNIs} + \text{number of HMNIs} + \text{number of external APB interfaces}) - 1$ .

For example, consider a configuration with an ASNI, an HSNI, an AMNI, an HMNI, and two PMNIs. Each PMNI has four external interfaces. NI-710AE assigns interface IDs for this configuration in the following way:

- The AXI interface on the ASNI gets completer interface ID 0.
- The AHB interface on the HSNI gets completer interface ID 1.
- The AXI interface on the AMNI gets requester interface ID 0.
- The AHB interface on the HMNI gets requester interface ID 1.
- The APB interfaces on the first PMNI get requester interface IDs 2–5.
- The APB interfaces on the second PMNI get requester interface IDs 6–9.

### Calculating the interface ID for the Configuration Network Interface

The CFGNI interface ID is used as the target for the configuration address region in the NI-710AE address map. The value of this ID depends on the number of external interfaces in your configuration. To determine this value for your configuration:

1. Determine `GT_TGTID_WIDTH`, which is the larger of the following values:
  - $\text{ceil}(\log_2(\text{number of xSNI interfaces}))$
  - $\text{ceil}(\log_2(\text{number of xMNI interfaces} + 1))$
2. Calculate the CFGNI interface ID, where  $\text{CFGNI interface ID} = 2^{\text{GT\_TGTID\_WIDTH}} - 1$ .

The CFGNI interface ID value is the same regardless of the number of CFGNIs in your configuration.

## 5.3 Output ID calculation

The width of the NI-710AE AMNI and HMNI output ID is a function of the xSNI input ID signal width and the Source ID (SrcID) width.

The width of the NI-710AE SrcID field depends on the number of completer interfaces in the system. Each NI-710AE completer interface is assigned a SrcID to identify the interface that originated a transaction. For example, a system with 32 xSNIs has 32 completer interfaces, so requires a 5-bit SrcID field. The SrcID of the incoming request is captured in the `node_id` field of the [ASNI node\\_type register](#) or [HSNI node\\_type register](#).

The input ID signal is the AxID signal for an ASNI or the HMASTER signal for an HSNI. The width of the input ID signals might be different for each completer interface. NI-710AE does not modify the incoming ID signal value, so the interconnect must be able to transport the widest ID values in the system.

AMNIs and HMNIs concatenate the SrcID and the input ID value to provide the output ID to pass downstream. The SrcID comprises the lower part of the output ID. Therefore, the maximum ID width at the AMNI or HMNI output is a product of the SrcID width and the maximum ID signal width. However, AMNIs and HMNIs make some optimizations in the likely case that paths are not



present between every xSNI and AMNI or HMNI. These optimizations let you reduce the required ID width at AMNI and HMNI outputs where the maximum width is not required.

## ID reduction

To reduce the ID width, the AMNIs and HMNIs use the following values to determine the output width:

- The largest input ID signal width for all the completer interfaces that have a valid path to this AMNI or HMNI
- The width required to represent the largest SrcID that has a valid path to this AMNI or HMNI

Socrates determines these values and sets the output ID width accordingly.

For example, consider an AMNI with paths to two ASNIs, which have SrcIDs 0 and 1 in the system. The AxID signal width for SrcID 0 is 3 bits and the AxID signal width for SrcID 1 is 5 bits. The AMNI only requires a 1-bit SrcID because the AMNI can only receive transactions from two ASNIs. The AMNI also only requires a 5-bit maximum AxID. The calculated output ID is a concatenation of the 1-bit SrcID and the 5-bit AxID signal width. Therefore, the AMNI has a 6-bit output ID.

## 6. Data width conversion

NI-710AE implements functionality to support a wide variety of external devices. NI-710AE can connect requester and completer devices that support different data widths, and which have different clocking and power requirements.

NI-710AE AMNIs include data width upsizing and downsizing functions to support AXI transfers between devices with different data widths. For more information, see [Upsizing AXI and ACE-Lite data width function](#) and [Downsizing AXI and ACE-Lite data width function](#).

AXI and AHB interfaces can include an optional set of user-defined signals. These User signals can be employed to provide extra information about transactions that is not defined in the AMBA specifications. For more information, see [User signals](#).

The NI-710AE SERDES unit resizes, splits, and collates flits to enable flits to move between regions with different link widths. For more information, see [Flit resizing and collating](#).

### 6.1 Upsizing AXI and ACE-Lite data width function

NI-710AE supports transactions from AXI requester and completer devices with different data widths. AMNIs are responsible for upsizing data that is sent from a device with a smaller data width than the transaction target.

The AMNI upsizing function can expand the data width in ratios of 1:2, 1:4, 1:8, 1:16, or 1:32.

Upsizing only optimizes the transaction size and length for write or read transactions that:

- Are modifiable, in other words  $AxCACHE[1] = 1$
- Use the full data width as signaled by  $AxSIZE$  on the input interface

There are several packing rules for different burst types and acceptance capabilities. Aligned and unaligned input bursts are defined as follows:

#### Aligned input burst

The network first aligns the address to the transfer size and then the address is aligned to the output data width boundary.

#### Unaligned input burst

The network aligns the address to the transfer size but does not align the address to the output data width boundary.

The following transaction rules apply to upsizing:

- If a transaction passes through the network, the upsizing function does not change the input transaction size and type.
- If the network splits input exclusive transactions into more than one output bus transaction, the network removes exclusive information the transactions that it creates.

- If multiple responses from created transactions are combined into one response, then the order of priority is:
  - DECERR is the highest priority
  - SLVERR is the next highest priority
  - OKAY is the lowest priority

The network upsizes different bursts as follows:

- The network converts INCR bursts into the optimum size based on the output data width. For more information, see [Upsizing INCR bursts](#).
- The network either passes WRAP bursts through unconverted or converts WRAP bursts into INCR bursts. For more information, see [Upsizing WRAP bursts](#).
- The network passes all FIXED bursts through unconverted.

### 6.1.1 Upsizing INCR bursts

The network converts all input INCR bursts that complete within a single output data width into an INCR1 of the minimum size possible. It packs all other INCR bursts into INCR bursts of the optimum size possible.

INCR<n> indicates an incrementing burst with n data beats. Bursts are never merged.

The following table shows how the network converts INCR bursts when it upsizes them. In this example, the input data width is 64 bits and the output data width is 128 bits.

**Table 6-1: Conversion of INCR bursts by the upsizing function**

INCR burst type	Converted to
64-bit INCR1	Passes through unconverted
64-bit aligned INCR2	INCR1
64-bit unaligned INCR2	Passes through unconverted
64-bit aligned INCR4	INCR2
64-bit unaligned INCR4	Sparse INCR3

### 6.1.2 Upsizing WRAP bursts

The network either passes WRAP bursts through unconverted, or converts WRAP bursts to one or two INCR bursts on the output bus.

Input WRAP bursts with a total payload that is less than the output data width are converted to single INCRs.

The following table shows how the network converts WRAP bursts when it upsizes them from 64 bits to 128 bits, that is, a ratio of 1:2. In this example, the input data width is 64 bits and the output data width is 128 bits.

**Table 6-2: Conversion of WRAP bursts by the upsizing function**

WRAP burst type	Converted to
128-bit aligned WRAP2	INCR1
128-bit aligned WRAP4	WRAP2
128-bit unaligned WRAP4	Depending on the address: <ul style="list-style-type: none"> <li>INCR2 + INCR1</li> <li>INCR1 + INCR2</li> </ul>

## 6.2 Downsizing AXI and ACE-Lite data width function

NI-710AE supports transactions from AMNI and ASNI devices with different data widths. The AMNI is responsible for downsizing data that is sent from a device with a larger data width than the transaction target.

The AMNI downsizing function can reduce the data width by ratios of 2:1, 4:1, 8:1, 16:1, and 32:1.

If the transaction is marked as a Non-cacheable transaction, the downsizing function does not merge data that is narrower than the destination bus.

### 6.2.1 Downsizing INCR bursts

NI-710AE converts INCR bursts that fall within the maximum payload size of the output data bus to a single INCR burst. It converts INCR bursts that are greater than the maximum payload size of the output data bus to multiple INCR bursts.

The following table shows how the network converts INCR bursts when it downsizes them, using a 2:1 downsizing ratio as an example.



Note

The INCR7 output example is only valid if the address is aligned to the destination width, and is not aligned to the source width. For example, if the address is 0x4 for a 64–32 bit downsizer, then an INCR7 output is generated. If the address is 0x1 for a 64–32 bit downsizer, an INCR8 output is generated.

**Table 6-3: Conversion of INCR bursts by the downsizing function**

INCR burst type	Converted to
Aligned INCR4	INCR8
Unaligned INCR4	INCR7 The INCR7 output example is only valid if the address is aligned to the destination width, and is not aligned to the source width. For example, if the address is 0x4 for a 64–32 bit downsizer, then an INCR7 output is generated. If the address is 0x1 for a 64–32 bit downsizer, an INCR8 output is generated.

INCR burst type	Converted to
Aligned INCR129	INCR256 + INCR2

INCR bursts with a size that matches the output data width pass through unconverted.

NI-710AE packs INCR bursts with a SIZE smaller than the output data width to match the output width whenever possible. NI-710AE uses the upsizing function to pack the INCR bursts.

## 6.2.2 Downsizing WRAP bursts

NI-710AE always converts WRAP bursts to WRAP bursts of twice the length, up to a maximum size of WRAP16. At the maximum size of WRAP16, NI-710AE treats the WRAP burst as two INCR bursts that can each map onto one or more INCR bursts.



If a WRAP transaction is aligned to the WRAP boundary, it is converted into an INCR transaction.

## 6.2.3 Downsizing FIXED bursts

NI-710AE converts FIXED bursts to one or more INCR1 or INCRn bursts, depending on the downsizing ratio.

The following table shows how the network converts FIXED bursts when it downsizes them.

**Table 6-4: Conversion of FIXED bursts by the downsizing function**

FIXED burst type	Converted to
FIXED1	INCR2
FIXED2	INCR2 + INCR2 + ...

NI-710AE optimizes unaligned FIXED bursts. If an unaligned input FIXED burst maps onto a single output beat, then the output is a FIXED burst of the optimal size.

## 6.3 User signals

NI-710AE supports User signal widths for different interface types and supports two different user modes.

The following table describes the supported User signal mode.

**Table 6-5: User signal mode description**

Mode	Description
User signal mode	Global mode that determines how user data signals, RUSER data portion, WUSER, HRUSER, and HWUSER, are handled across all AXI and AHB interfaces. This mode impacts the behavior with upsizing and downsizing.

The following table describes how the two different modes work and which parameters it impacts.

**Table 6-6: User signal mode behavior**

User signal mode	Upsizing or downsizing	Behavior	Comments
Legacy mode	Downsizing	<p>The interface width of the source is larger than the interface width of the destination.</p> <p><b>Note:</b> The user bits which accompanied the original data beat repeat for each of the downsized data beats the original data beat is split into.</p>	<p>This user data mode works if the user bits are for each transaction, that is, if they are identical across all beats of the same transaction. If the user bits are different for each data beat, then the scheme is lossy. This difference is clear for the upsizing case where only the bits for the last data beat of the user are retained and the others are lost.</p> <p>In this mode, the user data width is identical across all AXI and AHB interfaces.</p>
Legacy mode	Upsizing	<p>The interface width of the source is smaller than the interface width of the destination.</p> <p><b>Note:</b> The user bits which accompanied the last data beat from the source are sent with the upsized data beat. The combination of the smaller data beats creates the upsized data beat.</p>	<p>This user data mode works if the user bits are for each transaction, that is, if they are identical across all beats of the same transaction. If the user bits are different for each data beat, then the scheme is lossy. This difference is clear for the upsizing case where only the bits for the last data beat of the user are retained and the others are lost.</p> <p>In this mode, the user data width is identical across all AXI and AHB interfaces.</p>
Per Byte	Downsizing	<p>The interface width of the source is larger than the interface width of the destination.</p> <p><b>Note:</b> The user bits which accompanied the original data beat are appropriately split into corresponding portions. Each portion accompanies each downsized data beat and the original data beat is split into.</p>	<p>This User data mode is suited for use cases where the user bits that accompany the data are expected to scale appropriately with upsizing and downsizing.</p> <p>In this mode, the number of user data bits for each byte is identical across all network interfaces, that is, ASNIs, AMNIs, HSNIs, and HMNIs. This identical number enables the user data bits to be scaled up and down along with the <code>DATA_WIDTH</code> of each interface without it being lossy.</p> <p>Since the <code>DATA_WIDTH</code> of each interface can be different, the <code>USER_DATA_WIDTH</code> of different interfaces can be different and is computed as: <math>(DATA\_WIDTH / 8) * (\text{number of user data bits for each byte})</math></p>

User signal mode	Upsizing or downsizing	Behavior	Comments
Per Byte	Upsizing	<p>The interface width of the source is smaller than the interface width of the destination.</p> <p><b>Note:</b> The combination of the smaller data beats creates the upsized data beat. Similarly, the user bits which accompanied the individual incoming data beats from the source are combined into a single wider user data bus. The combined user bits accompany the upsized data beat at the destination.</p>	<p>This User data mode is suited for use cases where the user bits that accompany the data are expected to scale appropriately with upsizing and downsizing.</p> <p>In this mode, the number of user data bits for each byte is identical across all network interfaces, that is, ASNIs, AMNIs, HSNIs, and HMNIs. This identical number enables the user data bits to be scaled up and down along with the <code>DATA_WIDTH</code> of each interface without it being lossy.</p> <p>Since the <code>DATA_WIDTH</code> of each interface can be different, the <code>USER_DATA_WIDTH</code> of different interfaces can be different and is computed as: <math>(DATA\_WIDTH / 8) * (\text{number of user data bits for each byte})</math></p>

## User signal widths

Specify User signal widths for different interface types in NI-710AE:

**Table 6-7: Supported User signal widths**

Interface type	User signal	Signal width parameter	Comments
AXI	ARUSER	<code>USER_REQ_WIDTH</code>	This parameter is a single global parameter across all AXI and AHB interfaces. The parameter applies to ARUSER, AWUSER, and HAUSER.
AXI	AWUSER	<code>USER_REQ_WIDTH</code>	This parameter is a single global parameter across all AXI and AHB interfaces. The parameter applies to ARUSER, AWUSER, and HAUSER.
AXI	RUSER	<code>USER_DATA_WIDTH</code> + <code>RUSER_RESP_WIDTH</code>	<p>Issue H of the AXI specification introduces an extra user parameter for the read response to capture the per-transaction user information. This component of RUSER (present in bits <code>RUSER_RESP_WIDTH</code>) is the same for every beat of that transaction. However the <code>USER_DATA_WIDTH</code> component of RUSER can be different for every beat</p> <p><code>RUSER_RESP_WIDTH</code> is expected to be 0 when <code>USER_DATA_MODE</code> is 0.</p>
AXI	WUSER	<code>USER_DATA_WIDTH</code>	See the note in the preceding User signal mode behavior table for constraints on <code>USER_DATA_WIDTH</code> .
AXI	BUSER	<code>BUSER_RESP_WIDTH</code>	This parameter applies to the AXI write response width.
AHB	HAUSER	<code>USER_REQ_WIDTH</code>	This parameter is a single global parameter across all AXI and AHB interfaces and applies to ARUSER, AWUSER, and HAUSER.
AHB	HRUSER	<code>USER_DATA_WIDTH</code>	See the note in the preceding User signal mode behavior table for constraints on <code>USER_DATA_WIDTH</code> .
AHB	HWUSER	<code>USER_DATA_WIDTH</code>	See the note in the preceding User signal mode behavior table for constraints on <code>USER_DATA_WIDTH</code> .

The following table shows the supported User signal parameters and their ranges:

**Table 6-8: Parameters and supported range**

Parameter	Supported range
<code>USER_REQ_WIDTH</code>	0-256 bits
<code>USER_DATA_WIDTH</code>	<code>USER_DATA_MODE</code> = 0 → 0-64 bits

Parameter	Supported range
USER_DATA_WIDTH	<p>USER_DATA_MODE = 1</p> <p>Supports between 1-4 bits for each byte</p> <p>Max DATA_WIDTH = 1024 bits</p> <p>Max USER_DATA_WIDTH = <math>(1024 / 8) \times 4 = 512</math> bits</p>
BUSER_RESP_WIDTH	0-64 bits
RUSER_RESP_WIDTH	0-64 bits

## 6.4 Flit resizing and collating

NI-710AE includes a configurable SERDES unit that allows flits to move between interconnect regions with different link widths. The SERDES unit resizes flits by collating or dividing them.

You can configure the SERDES unit to resize flits according to different width ratios:

### Upsizing (N:M)

Multiple input flits are collated together to form a single large output flit.

### Downsizing (M:N)

A single input flit is read into multiple smaller output flits.

After resizing, output flits are aggregated in a FIFO until one of the following conditions is met:

- The FIFO tidemark threshold is reached
- The last flit is received
- The aggregating FIFO is full

When one of these conditions is met, flit aggregation stops, and the flits exit the block.

At build time, you can configure the number of flits to aggregate and store until the packet is released.



## 7. Transaction handling

NI-710AE can connect requester and completer devices that support different AMBA protocols.

The NI-710AE AXI5 to AHB5 bridge translates AXI exclusive bursts and exclusive transactions into transfer types that are supported by the AHB protocol. For more information, see [Exclusive and locked accesses](#).

According to the [AMBA® AHB Protocol Specification](#), requesters that require locked transfers must assert HMASTLOCK, while there is no requirement for completers to implement this signal. So, NI-710AE HMNIs and HSNIs respond differently to locked transfers. For more information, see [AHB locked transfers](#).

NI-710AE AXI network interfaces include the option to support the Memory Tagging Extension (MTE). For more information, see [Memory tagging support](#).

### 7.1 Exclusive and locked accesses

The AXI protocol supports exclusive bursts, but the AHB protocol only supports single (length 1) exclusive transfers. To account for this difference, the AXI5 to AHB5 bridge handles AXI exclusive bursts and single AXI exclusive transactions differently.

AXI exclusive accesses and AHB exclusive transfers are a read transaction followed by a write transaction to the same address range. AXI exclusive bursts are similar, except that the read and write transactions comprise sequences of transfers. Exclusive accesses and bursts allow for semaphore-type operations without requiring the bus to remain dedicated to a particular requester throughout the operation.

Unlike AXI exclusive accesses, AHB exclusive transfers must be single-beat transfers. So, if the AXI5 to AHB5 bridge receives an AXI exclusive burst, it translates the burst to normal (non-exclusive) AHB transfers. When the bridge receives a single AXI exclusive transaction, it translates the transaction into an exclusive AHB transfer.

The AXI5 to AHB5 bridge does not support single sparse exclusive writes because splitting the write transaction would create an exclusive AHB burst. As the preceding exclusive read might have been answered with HEXOKAY, the bridge always responds with SLVERR for a single sparse exclusive write. The bridge returns SLVERR because although OKAY is a valid exclusive response, an OKAY response could cause the AXI requester to repeat the exclusive write indefinitely.

The bridge uses the AxID values to identify the AXI requester that is issuing an exclusive access. For the AHB transfer, the bridge copies the AxID value to HMASTER.

The following table shows the AHB transfer types to which the AXI5 to AHB5 bridge maps different AXI exclusive accesses.

**Table 7-1: AXI5 to AHB5 bridge exclusive access mapping**

Received AXI access type	Received AXI transaction type	Translated AHB transfer type
AXI exclusive read	Single	Exclusive AHB transfer
AXI exclusive read	Burst	Normal AHB transfers
AXI non-sparse exclusive write	Single	Exclusive AHB transfer
AXI non-sparse exclusive write	Burst	Normal AHB transfers
AXI sparse exclusive write	Single	Normal AHB transfer (SLVERR)
AXI sparse exclusive write	Burst	Normal AHB transfers

## 7.2 AHB locked transfers

HSNIs and HMNI behave differently when handling AHB locked transfers.

In the AHB protocol, locked transfers are sequences that are indivisible and must be processed before any other transfers are processed. Typically, locked transfers are used to ensure that a completer does not perform other operations between the read and write phases of an instruction. Requesters that require locked accesses must assert the HMASTLOCK signal.

At an HSNI, HMASTLOCK is ignored for the HSNI. At an HMNI, any non-modifiable read or write request is mapped to a locked sequence. HMASTLOCK is asserted for AHB transfers belonging to the original non-modifiable read or write request. No arbitration is permitted for the length of the burst.



Note

Although an AXI burst can cross a 1KB address range, the AHB protocol requires that all transfers in a locked sequence go to the same completer address region. If an HMNI receives a non-modifiable burst with a size of more than 1KB, the burst is sent as a non-modifiable AHB burst. However, HMASTLOCK is not asserted and the response is sent with SLVERR.

## 7.3 Memory tagging support

You can enable memory tagging on any AXI network interface in NI-710AE by using the **Memory Tagging Extension** option.

NI-710AE only transports the Memory Tagging Extension (MTE) tags. There is no tag splitter or tag cache within the interconnect. The AMBA AXI specification describes two MTE configurations, that is, basic and standard. All combinations of these MTE configurations on the ASNI and AMNI are supported except when the ASNI is configured as standard and the AMNI is configured as basic.

The following table describes the AMNI behavior depending on the level of MTE support in the interconnect and at the AXI interface.

**Table 7-2: AMNI behavior with different MTE support within NI-710AE**

MTE support in the interconnect	Memory Tagging Extension value in AMNI AXI interface	Behavior
False	<b>Disabled</b>	Not supported.
False	<b>Basic</b>	Tie off AxtAGOP to 0 from the AMNI.
False	<b>Standard</b>	Tie off AxtAGOP to 0 from the AMNI. BTAGMATCH is not present on the AMNI AXI interface.
Basic	<b>Disabled</b>	Ignore tag operation but pass the transactions through. BTAGMATCH is not present on the AMNI AXI interface.
Basic	<b>Basic</b>	Propagate AxtAGOP. BTAGMATCH is not present on the AMNI AXI interface.
Basic	<b>Standard</b>	Propagate AxtAGOP. BTAGMATCH on the AMNI AXI interface is not used.
Standard	<b>Disabled</b>	Ignore tag operation and pass the transactions through. For setting BTAGMATCH in the response upstream from the AMNI, if incoming request is Match then return BTAGMATCH as 0b10, Fail. Otherwise return BTAGMATCH as 0b00.
Standard	<b>Basic</b>	Propagate AxtAGOP.  For setting BTAGMATCH in the response upstream from the AMNI, if incoming request is Match then return BTAGMATCH as 0b10, Fail. Otherwise return BTAGMATCH as 0b00.
Standard	<b>Standard</b>	Propagate AxtAGOP and BTAGMATCH.

Where MTE support in the interconnect is based on the least common support across all the ASNs, then:

- If none of the ASNs support MTE, then MTE support in the interconnect is false.
- If at least one of the ASNs is configured for MTE basic, and none of the ASNs are configured for MTE standard, then MTE support in the interconnect is basic.
- If at least one of the ASNs is configured for MTE standard, then MTE support in the interconnect is standard.

## 8. Reliability, Availability, and Serviceability

NI-710AE supports Reliability, Availability, and Serviceability (RAS) features.

NI-710AE AXI and AHB network interfaces can be configured to transport data parity, Error Correcting Code (ECC), and poison information. For more information, see [Support for transporting data parity, ECC, and poison information](#).

### 8.1 Support for transporting data parity, ECC, and poison information

NI-710AE supports transporting data parity, ECC, or poison information through the interconnect.

This support only applies to AXI RDATA and WDATA signals, and AHB HRDATA and HWDATA signals. NI-710AE only transports the parity, ECC, or poison information. There is no support for generating or checking parity or ECC within the interconnect. NI-710AE uses the RUSER, WUSER, HRUSER, and HWUSER user data bits to receive and transmit parity, ECC, or poison information. For more information, see the *Data parity, ECC, and poison information* section of the NI-710AE Configuration and Integration Manual.

For this feature, the system builder must configure the user data mode by setting the **User Signal Width Mode** option in Socrates to **perByte**. This setting lets NI-710AE support upsizing and downsizing the parity or ECC information in the user data bits appropriately.

For more information on the user data mode, see [User signals](#).

## 9. Secure and Non-secure accesses

NI-710AE supports Secure and Non-secure accesses from request and response sources. The mechanisms that NI-710AE uses to handle these accesses depend on the AMBA protocol that the source supports.

NI-710AE also supports a full Access Protection Unit (APU), which you can use to restrict access according to these security attributes. For more information, see [Memory access protection and the Access Protection Unit](#).

### 9.1 Security access permissions of AXI requests

Security access permissions are signaled by NI-710AE on incoming AXI requests through AxPROT[1]. Depending on the value of AxPROT[1], a request can target specific register types within the interconnect.

NI-710AE transports AxPROT[1] on each request, which encodes whether the request is Secure or Non-secure. The incoming AxPROT[1] value at the ASNI is conveyed on the outgoing interface from the AMNI.

### 9.2 Security access permissions of AHB requests

You can configure whether each HSNI and HMNI in your design supports Secure transfers. Depending on the type of device that is attached, each functional unit also has configurable registers that define how the interface handles request security.

The AHB5 SECURE\_TRANSFERS field defines whether the interface supports Secure transfers. When an interface supports Secure transfers, HNONSEC is asserted for a Non-secure transfer and deasserted for a Secure transfer.

There are four security configuration options for AHB completer interfaces. The following table describes each option.

**Table 9-1: AHB completer interface security configuration options**

Configuration option	Description
Pin	The HNONSEC pin exists and passes the security attribute.
Programmable	The HSNI contains a software programmable register to set the security attribute for requests from this completer interface. If the register bit is set to 1, then the request is Non-secure and if the bit is set to 0, then the request is Secure. See <a href="#">HSNI node_control register</a> .
Always Secure	At build time, all requests which originate from this completer interface are marked as Secure.
Always Non-secure	At build time, all requests which originate from this completer interface are marked as Non-secure.

There are also four security configuration options for AHB requester interfaces. The following table describes these security configuration options.

**Table 9-2: AHB requester interface security configuration options**

Configuration option	Description
Pin	The HNONSEC pin exists and passes the security attribute to the downstream completer.
Programmable	The HMNI contains a software programmable register to set the security attribute of the assets in the downstream completer. If the register bit is set to 1, then the downstream completer is Non-secure. If the register bit is set to 0, then the downstream completer is Secure. See <a href="#">HMNI node_control register</a> .
Always Secure	Only Secure transactions can access components that are attached to this requester interface.
Always Non-secure	Both Secure and Non-secure transactions can access components that are attached to this requester interface.

The following table describes the reset values for the HSNi and HMNI programmable security register.

**Table 9-3: HSNi and HMNI programmable security register reset values**

Interface	Reset value	Description
HSNi	1	Out of reset, all requests from HSNi are Non-secure
HMNI	0	Out of reset, all assets in the downstream AHB completer are considered to be Secure

If a Non-secure transaction targets a requester interface which is either programmed as Secure, or is set to always Secure, the HMNI does not forward the transaction. Instead, the HMNI provides the following responses, with no error indication:

#### Read request

The HMNI responds with zeroed data.

#### Write request

The HMNI drops all write data and issues a protocol-compliant write response without error indication.

If a HSNi is set to always Non-secure or programmed to be Non-secure, then it is not permitted access to Secure registers within NI-710AE. This constraint is defined in [Register security attribute and security classification](#). If the Secure access attribute is overridden as defined in [Secure access register](#), no access to Secure registers occurs.

## 9.3 Security access permissions of APB requests

Each PMNI can have up to 16 APB interfaces attached. Some interfaces can be configured for APB3, APB4, or APB5. You can configure whether each interface supports Secure transfers.

You can independently configure the security behavior of each of the APB interfaces. The following table describes the APB configuration option.

**Table 9-4: APB security configuration options**

Configuration option	Description
Pin	When this option is selected for an APB4 interface, the PPROT pin communicates the security attribute to the downstream completer.  The pin option is not available for APB3 interfaces as PPROT is not supported on APB3.
Programmable	PMNIs contain a software-programmable register to set the security attribute of the assets in the downstream completer. If the register bit is set to 1, the downstream completer is Non-secure. If the register bit is set to 0, then the downstream completer is Secure.  Where the protocol is APB4: <ul style="list-style-type: none"> <li>If the register is configured to indicate a Non-secure completer, the security attribute is passed on the PPROT pin.</li> <li>If the register is configured to indicate a Secure completer, the Non-secure requests are not passed downstream. Instead, they are terminated at the PMNI with a protocol-compliant response and SLVERR. Incoming Secure requests are passed downstream to the completer with the security attribute communicated on the PPROT pin.</li> </ul>
Always Secure	Only Secure transactions can access components that are attached to this specific APB requester interface.  If the interface supports APB4, the security attribute is communicated on the PPROT pin. Non-secure requests are not passed downstream. Instead, they are terminated at the PMNI with a protocol-compliant response and SLVERR.
Always Non-secure	Both Secure and Non-secure transactions can access components that are attached to this specific APB requester interface. If the interface supports APB4, the security attribute is communicated on the PPROT pin.

When an interface is configured as programmable, use the PMNI node\_control software programmable register to indicate the security attribute for each downstream APB interface. For more information on Secure and Non-secure APB interfaces, see [PMNI node\\_control register](#).

The following table contains the reset value and description for the PMNI programmable security register.

**Table 9-5: PMNI programmable security register reset value**

Interface	Reset value	Description
PMNI	0	Out of reset all assets in the downstream AHB completer are considered to be Secure

If the interface is configured as programmable, always Secure or always Non-secure, The PMNI is responsible for completing the Secure access permission check. If a Non-secure transaction targets a requester APB interface that is either programmed as Secure or set to be Secure at build time, the PMNI does not forward the transaction to the downstream APB completer. Instead, the PMNI provides the following responses with no error indication:

#### Read request

The PMNI responds with zeroed data.

#### Write request

The PMNI drops all write data and issues a protocol-compliant write response without error indication.

## 9.4 Register security attribute and security classification

Each NI-710AE register is classified according to its security attribute value. The classification affects the register access permissions.

For requests targeting internal NI-710AE registers, the security attribute determines whether the request can access a specific register. For more information, see [TrustZone technology and security](#).

The NI-710AE registers are classified according to the following types:

### Secure

Accessible only by Secure requests, but this access permission can be overridden. For more information about overriding this access permission, see [Secure access register](#).

### Secure debug

Includes PMU registers and Silicon Debug registers. These registers are only accessible by Secure accesses. This access permission can be overridden, but the way that this override is performed is different to standard Secure registers. For more information about overriding this access permission, see [Secure access register](#).

### Secure only

Accessible only by Secure requests, and this access permission cannot be overridden.

### Non-secure

Accessible by Secure and Non-secure requests.

## 9.5 Secure access register

The NI-710AE Secure access register is a Secure only register that is used to modify the security access permissions of the other Secure registers.

This register is present in all register regions including:

- Global configuration register regions
- Voltage, power, and clock domain register regions
- xMNI and xSNI register regions
- PMU register region

Software can program this register to override the Secure access permissions of any specific register region instance. These register region instances include the xMNI and xSNI regions.

The Secure access register has two bits:

### Bit[0]

Non-secure access override bit. If this bit is set, Non-secure accesses can access all Secure registers within that register region, including the PMU registers and Silicon Debug registers.



### Bit[1]

Non-secure debug monitor override bit. If this bit is set, Non-secure accesses can access the PMU registers and Silicon Debug registers within that region. If bit[0] is not set, but bit[1] is set, then the security access permissions are only overridden for the PMU and Silicon Debug registers.

For more information, see [Programmers model](#).

## 9.6 Secure debug

NI-710AE supports Secure debug through the SPNIDEN, SPIDEN, and DBGGEN signals.

The performance monitoring events corresponding to each upstream and downstream interface are specified in [Performance monitoring](#). The SPNIDEN, SPIDEN, and DBGGEN inputs determine the conditions for permitting Secure events to be captured in the PMU event counters or Silicon Debug registers. The SPNIDEN, SPIDEN, and DBGGEN inputs are described in [Debug and Performance Monitoring Unit interface signals](#).

The following equation determines whether Secure debug is permitted:

Secure debug = ((SPIDEN & DBGGEN) | SPNIDEN) & (DBGGEN | NIDEN)

If Secure debug is not enabled, then the PMU event counters and Silicon Debug registers can only capture Non-secure events.

## 9.7 Interrupt and error logging register security

NI-710AE has separate registers for Secure and Non-secure interrupt status and error logging. NI-710AE also has separate Secure and Non-secure interrupt pins for each power domain.

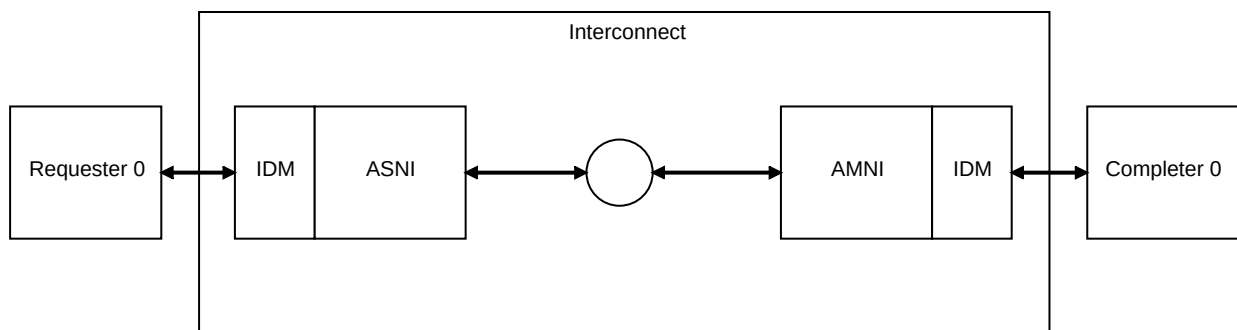
For more information, see [Error handling and interrupt security](#).

## 10. Interconnect Device Management

Interconnect Device Management (IDM) lets the interconnect configure, manage, and reset individual or groups of system components in isolation, without affecting other components. The IDM functionality is an optional feature and integrates with all NI-710AE network interface blocks.

If enabled on a network interface, the IDM block is instantiated between the network interface and the device to which it connects. For example, if enabled on an ASNI, the IDM block is instantiated on the requester device to ASNI connection. The following figure shows an example system with IDM blocks integrated with ASNI and AMNI components:

**Figure 10-1: IDM integration with network interface blocks**



The IDM block on a requester device to xSNI connection provides control and status information about transactions that the requester device issues. IDM blocks on xMNI to completer device connections provide control of and status information about transactions that are issued to the completer device.

Each IDM block has its own set of software-accessible registers. These registers are in the same 4KB NI-710AE memory region as all other registers belonging to the IDM-enabled network interface. For more information on the relevant IDM registers, see:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

All network interface IDM blocks include the following key features:

- Software access to configuration, control, and status of the attached device through the NI-710AE programmers view
- [Timeout detection through the IDM block](#)
- [Error logging through the IDM block](#)

- [IDM soft reset mode](#)
- [IDM access control](#)

This NI-710AE release has some constraints on specific AXI5 properties when IDM is enabled:

- Some aspects of AXI5 atomics, for example AtomicLoad, AtomicSwap, and AtomicCompare, have both a read and a write response. If you enable IDM, NI-710AE does not track a timeout on the read response for the atomic request on the AW channel.
- AXI-G cache maintenance for persistence operations on the write channel can have a persist response that arrives separately from the completion response. You cannot enable IDM on an AMNI that has the `persist_cmo` property set to true.
- AXI-H adds two types of support for the Memory Tagging Extension (MTE), basic and standard.
  - Standard support means that memory tagging is supported on the interface, all MTE signals are present. You cannot enable IDM on an AMNI that has MTE support set to standard.
  - Basic support means that memory tagging is supported on the interface at a basic level. A limited set of tag operations is permitted. BTAGMATCH is not present and BCOMP is not required.

## IDM and read data chunking features enabled together

When you enable the IDM and read data chunking features together, protocol violations can exist. Violations occur under real error scenarios where IDM logic must synthesize read and write data beats with SLVERR.

The violations also occur if IDM soft reset or isolation entry occurs in the middle of an outstanding request. The IDM logic does not monitor the individual CHUNKNUMs and CHUNKSTRBs that have already arrived for each outstanding request. The monitoring process is very expensive, however the synthesized data beats carry an SLVERR response anyway.

## Completer interface enters soft reset mode during a write transaction

When a completer interface enters soft reset in the middle of a write transaction, the ASNI synthesizes any remaining write data beats. All the write data beats that are required by the transaction are completed in a protocol-compliant manner. The synthesized data beats have zero write data and zero write strobes. Therefore, no memory location is updated or corrupted with this synthesized data beat. For an example, see [xSNI write data transaction timeout leading to soft reset in Soft reset use case examples for xSNIs and xMNIs](#).

However, AXI protocol violations can occur. For example, a WriteUniqueFull, which implies a full cacheline write, requires all write strobes to be set. Similarly, WriteUniqueFull with MTE tag update must have all associated WTAGUPDATE bits asserted. For synthesized data beats,WSTRB, WTAGUPDATE, and WTRACE are driven to zero. Therefore, the synthesized data beats do not update and corrupt the memory.

## Considerations when setting the timeout

You must program an adequate timeout value to account for functional scenarios that can lead to delays because of network contention or backpressure from external interfaces. For example, an ASNI has accepted numerous incoming write requests, AWVALID, where each request is a very large burst. It can take a significant amount of time for the ASNI to accept all the incoming write

data, WVALID. As a result, the most recent requests experience a large delay between accepting AVALID and receiving WVALID corresponding to the first data beat.

Furthermore, contention within the interconnect can lead to longer delays. Set the timeout value so these functional scenarios are not falsely triggered as misbehaving requesters or completers.

## 10.1 IDM and device discovery

The IDM functionality extends and facilitates the NI-710AE discovery process by providing a value that is configurable by the system designer to identify devices that are attached to the interface.

The discovery mechanism that is described in the [Discovery](#) section enables software to discover the voltage, power, and clock domain of any interface in the interconnect. When IDM is enabled on an interface, NI-710AE adds a corresponding `idm_device_id` register. This register contains a 32-bit `device_id` value that is configured by the designer. The `device_id` value is accessible through the programmers' view, and facilitates identification of devices that are attached to the interface and overall system discovery.

For more information about `idm_device_id` configuration, see:

- [ASNI `idm\_device\_id` register](#)
- [AMNI `idm\_device\_id` register](#)
- [HSNI `idm\_device\_id` register](#)
- [HMNI `idm\_device\_id` register](#)
- [PMNI `idm\_device\_id` register](#)

## 10.2 Timeout detection through the IDM block

The IDM block timeout detection feature uses an interrupt to indicate when transactions from or to the attached device have stalled or failed to progress. This feature is available at both xMNIs and xSNIs.

Example cases where an xSNI IDM block indicates stalled or failed transactions from a requester device include:

- The external device fails to send complete write data for a received write address phase transaction. The interconnect can indicate failure at any point in the write data beat count.
- The external device fails to send a write address phase for a write transaction with leading write data.
- The external device fails to accept a write response for an issued write transaction.
- The external device fails to accept all read data beats for an issued read transaction.

Example cases where an xMNI IDM block indicates stalled or failed transactions to a completer device include:

- The external device fails to accept a read address or write address phase transaction.
- The external device fails to accept a write data beat for a write transaction. The interconnect can indicate failure at any point in the write data beat count.
- The external device fails to send a write response for an issued write transaction.
- The external device fails to send all read data beats for an issued read transaction.

When the timeout detection feature is enabled, the block produces a level-based interrupt and stores various transaction details for software-based investigation and debug. For more information on the transaction details, see the relevant IDM registers in the:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

Once the IDM block detects a timeout, and if `idm_reset_control.reset_control_auto` register field is asserted, the network interface raises a timeout interrupt and enters soft reset mode. At this point:

- The interface gates new transactions from the external device. For example, the interface gates any incoming responses from the downstream completer and prevents them from entering the interconnect at the requester interface.
- All outstanding requests are completed in a protocol-compliant manner.

The timeout does not cause the traffic on the interconnect to start backing up as the network interface completes all outstanding transactions. The network interface remains in soft reset mode until the software requests an exit from this mode using a write to the `idm_reset_control.reset_control` register field. For more information see, [IDM soft reset mode](#).

## 10.3 Error logging through the IDM block

When you configure a network interface to include IDM functionality, extra logic is enabled to trigger error detection and interrupt generation in the IDM block.

IDM block error logging uses an interrupt to indicate when an IDM-enabled xMNI or xSNI signals an AMBA protocol bus error. This feature is available at both xMNIs and xSNIs.

When this feature is enabled, the block produces a level-based interrupt and stores various transaction details for software-based investigation and debug. For more information on the transaction details, see the:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)

- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

## 10.4 IDM soft reset mode

The IDM soft reset feature permits software to isolate an endpoint and reset attached erroneous devices without affecting other endpoints or devices. This feature is available at both xMNIs and xSNIs.

Use soft reset together with either or both of the error logging or timeout detection features. For more information, see [Timeout detection through the IDM block](#) and [Error logging through the IDM block](#).

IDM soft reset mode consists of two distinct stages:

### Recovery

The network interface gates its external interfaces. Any transactions that were outstanding when entering soft reset mode are completed in a protocol-compliant fashion by synthesizing the remaining transfers. The endpoint synthesizes transfers to complete any new transactions when in recovery mode.

### Soft reset assertion

The external soft reset pin associated with the device connected to the timed-out interface is asserted.

A write of 1 to the `idm_reset_control.reset_control` register field causes the network interface to enter soft reset mode. The external soft reset pin is activated, unless soft reset mode was triggered by an auto entry. A write of 0 to the `idm_reset_control.reset_control` register field when it is already set to 1 causes an exit from soft reset mode and the deassertion of the external soft reset pin. The recovery stage is entered either when a timeout occurs and `idm_reset_control.reset_control_auto` register is already set to 1, or when 1 is written to the `idm_reset_control.reset_control` register field.

For more information on the IDM registers, see:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

To enter soft reset assertion, write 1 to the `idm_reset_control.reset_control` register field.

## 10.4.1 Hardware-initiated entry based on timeout detection

If a timeout is detected, NI-710AE enters soft reset mode.

In soft reset mode, the relevant network interface immediately asserts the timeout if enabled, and logs errors into the IDM registers. If `idm_reset_control.reset_control_auto` is set to 1, then the network interface enters the recovery stage. If there are also outstanding requests at the time, the soft reset request is received, the network interface block completes the transactions in a protocol-compliant manner, and the external interface is gated.

When a timeout is detected at an xMNI or an xSNI, the network interface enters soft reset mode if `idm_reset_control.reset_control_auto` is set to 1. In this case, the soft reset pin is not asserted. Therefore, for xMNIs:

- No further transactions are sent downstream
- Any incoming responses from downstream are gated and are not permitted to enter the interconnect
- Any required responses are synthesized with SLVERR and sent upstream to complete transactions in a protocol-compliant manner

However, for xSNIs:

- No further incoming transactions are accepted
- Any required responses are synthesized with OK and sent upstream to complete transactions in a protocol-compliant manner

This hardware-initiated entry into soft reset mode does not affect the soft reset pin. To toggle the external soft reset pin, soft reset mode must be requested by writing 1 to the `idm_reset_control.reset_control` field. If the endpoint is already in soft reset mode, writing 1 to the `idm_reset_control.reset_control` field asserts the external soft reset pin to the device.

Network interfaces remain in soft reset mode until 0 is written to the `idm_reset_control.reset_control` field. In addition to causing the interface to exit from soft reset mode, writing 0 to the `idm_reset_control.reset_control` field also deasserts the external soft reset pin.

For more information on exiting soft reset mode, see the `idm_reset_control` register in the:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

For more information on timeouts, see [Timeout detection through the IDM block](#).

## 10.4.2 Software-initiated entry

IDM-enabled NI-710AE endpoints can reset attached requester or completer devices under software control.

This reset can occur independently of the rest of the interconnect and other external devices. The `idm_reset_control` register associated with the endpoint provides the functionality to request that the attached device is placed into soft reset. The endpoint ensures that there are no incomplete transactions at either the requester or completer on reset.

When NI-710AE receives a soft reset request, the relevant xSNI or xMNI immediately isolates the external interface. If there are outstanding requests at the time NI-710AE receives the soft reset request, the network interface block completes the transactions in a protocol-compliant manner. With software initiated entry, the relevant network interface also toggles the external reset pin to the attached device. The synthesized responses have an SLVERR indication. For information on the transaction flows, see [Soft reset use case examples for xSNIs and xMNIs](#).

The network interface remains in soft reset mode until software writes 0 to the `idm_reset_control` register.`reset_control` field to exit soft reset mode. Writing 0 to the `idm_reset_control`.`reset_control` field causes the interface to exit soft reset mode and also deasserts the external soft reset pin. For more information on exiting soft reset mode, see the `idm_reset_control` register in the:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

## 10.4.3 Reset initialization input pin

IDM enabled endpoints receive an external input pin, `device_sreset_strap_i`, that is connected to the `idm_reset_control`.`reset_control` field. The external input pin only controls the `idm_reset_control`.`reset_control` field.

If the pin value is set to 0, then there is no change in endpoint behavior out of reset. However, if the pin value is set to 1, when the endpoint exits soft reset it behaves as if it is already in soft reset mode. That is, the endpoint behaves as if software had written 1 to the `idm_reset_control`.`reset_control` field to request entry to soft reset mode. Therefore:

- The asserted external soft reset pin is asserted immediately out of reset
- The external interface is isolated
- Any incoming requests are terminated at the endpoint and completed in a protocol-compliant manner with SLVERR responses

For more information on the `idm_reset_control` field names and bit assignments, see:



- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

## 10.5 IDM access control

Specific scenarios might require software to isolate attached devices from the interconnect in a controlled way. The IDM access control feature enables such device isolation and is available at both xSNIs and xMNIs.

IDM access control is useful in various situations, for example device power management or disabling of malfunctioning devices. Using this feature, software can set individual endpoints to prevent transactions from progressing through the interface. By preventing movement of transactions to or from the interconnect, the attached device is isolated from the rest of the interconnect.

The `idm_access_control` register that is associated with the endpoint is used to request that the attached device is isolated. IDM access control ensures that there are no incomplete transactions at either the requester or completer when the device is isolated.

When an isolation request is received, the corresponding xSNI or xMNI waits for the current outstanding transactions to complete normally before entering isolation. This wait is the primary difference between isolation and soft reset.

To reach a clean point where outstanding transactions are completed and then the external device is reset, the following must occur:

1. First, software must request isolation entry using the `idm_access_control` register.
2. When isolation entry is successful, software requests a soft reset entry using the `idm_reset_control` register.

For more information about the `idm_access_control` register and the `idm_reset_control` register, see:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

For an xSNI, isolation means the network interface does not send incoming requests into the interconnect. For an xMNI, isolation means the network interface does not send incoming requests to the downstream completer. Any new requests that are received are marked for loopback even while there are still outstanding transactions to complete. That is, they are isolated. The xSNI or xMNI generates internally looped back responses with an SLVERR indication for the new requests. These looped back responses happen when all current outstanding transactions have completed normally.

For example, for an AMNI:

- Requests that are already outstanding complete normally.
- The AMNI implements burst splitting for cases related to downsizing. If there is an original incoming request that is in the process of being burst split, then the AMNI issues all burst split requests corresponding to the original request downstream. AMNI completes those requests normally.
- Any transaction waiting for acceptance downstream, AxValid\_o without axready\_i, is sent downstream and completes normally.
- Subsequent requests are marked for loopback.
- Requests that are marked for loopback:
  - Wait for the channel to enter loopback mode.
  - Wait for currently outstanding transactions and all transactions sent downstream to complete normally.
  - Send SLVERR responses at this point only.
- Any new incoming requests are sent with SLVERR and follow the same sequence.

The AXI protocol has independent read and write address channels. Therefore, the read and write channels can enter isolation or loopback mode at different times after receiving an isolation entry request. However, the `idm_access_control` register indicates isolation entry only after both the read and write channels have entered isolation mode. As a result, either channel can receive loopback responses with SLVERR even before the `idm_access_control` register shows successful isolation entry. Software must either quiesce both the channels before requesting isolation entry, or must be able to handle the preceding behavior.

For more information about the transaction flow, see [Access control use case example for xMNIs and xSNIs](#).

## 10.6 Soft reset use case examples for xSNIs and xMNIs

These examples show the expected use case for the IDM soft reset functionality for a stalled read transaction and a write data transaction at an xSNI. The examples also show an xMNI write transaction timeout and an xMNI read transaction timeout, both leading to a soft reset.

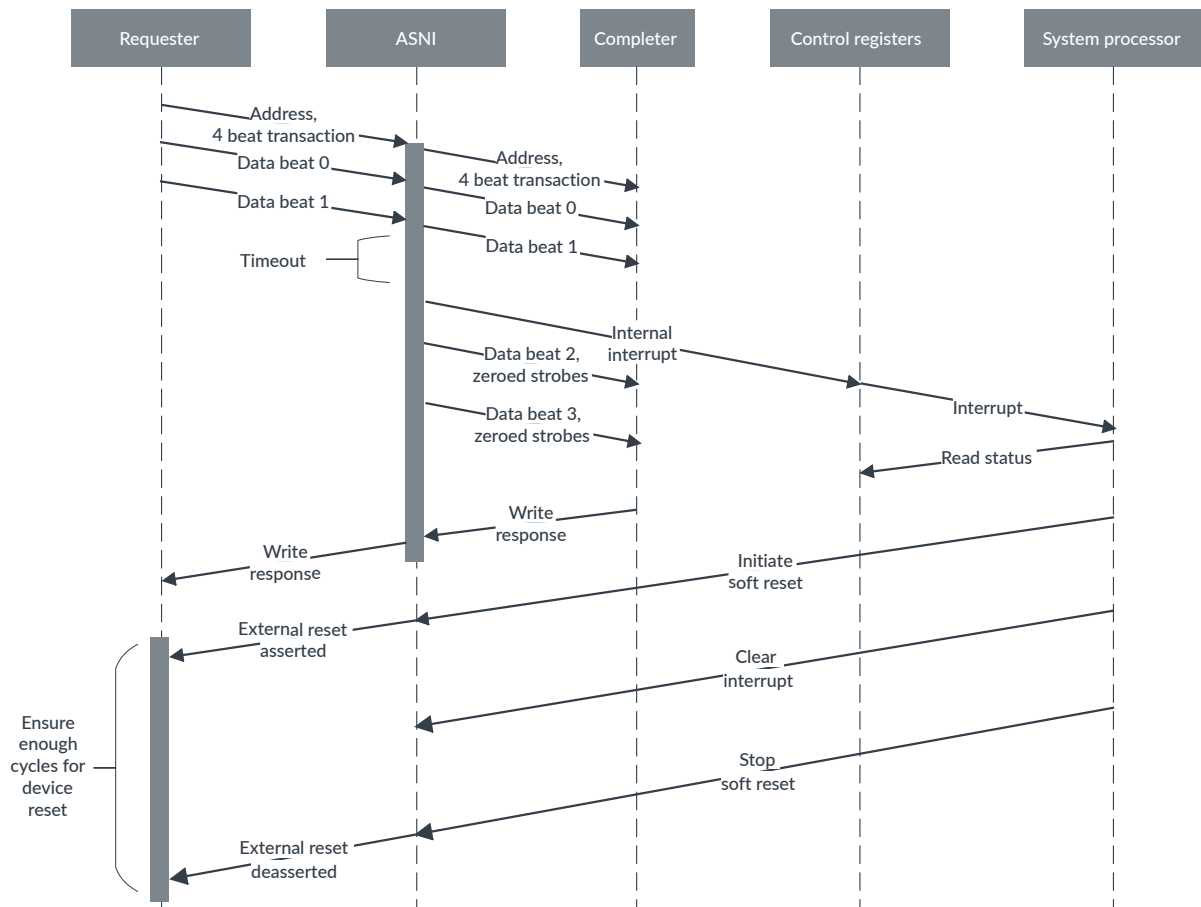
### xSNI write data transaction timeout leading to soft reset

In this example, the requester device has stalled after issuing the second of four write data beats.

If the `idm_reset_control.reset_control_auto` field is set to 1, on detection of the timeout, hardware automatically enters soft reset mode to gate the external interface. The hardware also synthesizes the outstanding write data beats with zeroed write strobes. The write response indication is sent upstream when the ASNI receives it. After the software writes 1 to the `idm_reset_control.reset_control` field to initiate the soft reset sequence for the endpoint, the external reset pin is also asserted. This assertion resets the attached stalled completer device.

The following figure shows the sequence of events for an ASNI write data transaction timeout leading to a soft reset.

**Figure 10-2: ASNI write data transaction timeout leading to a soft reset**



### xSNI stalled read transaction leading to soft reset

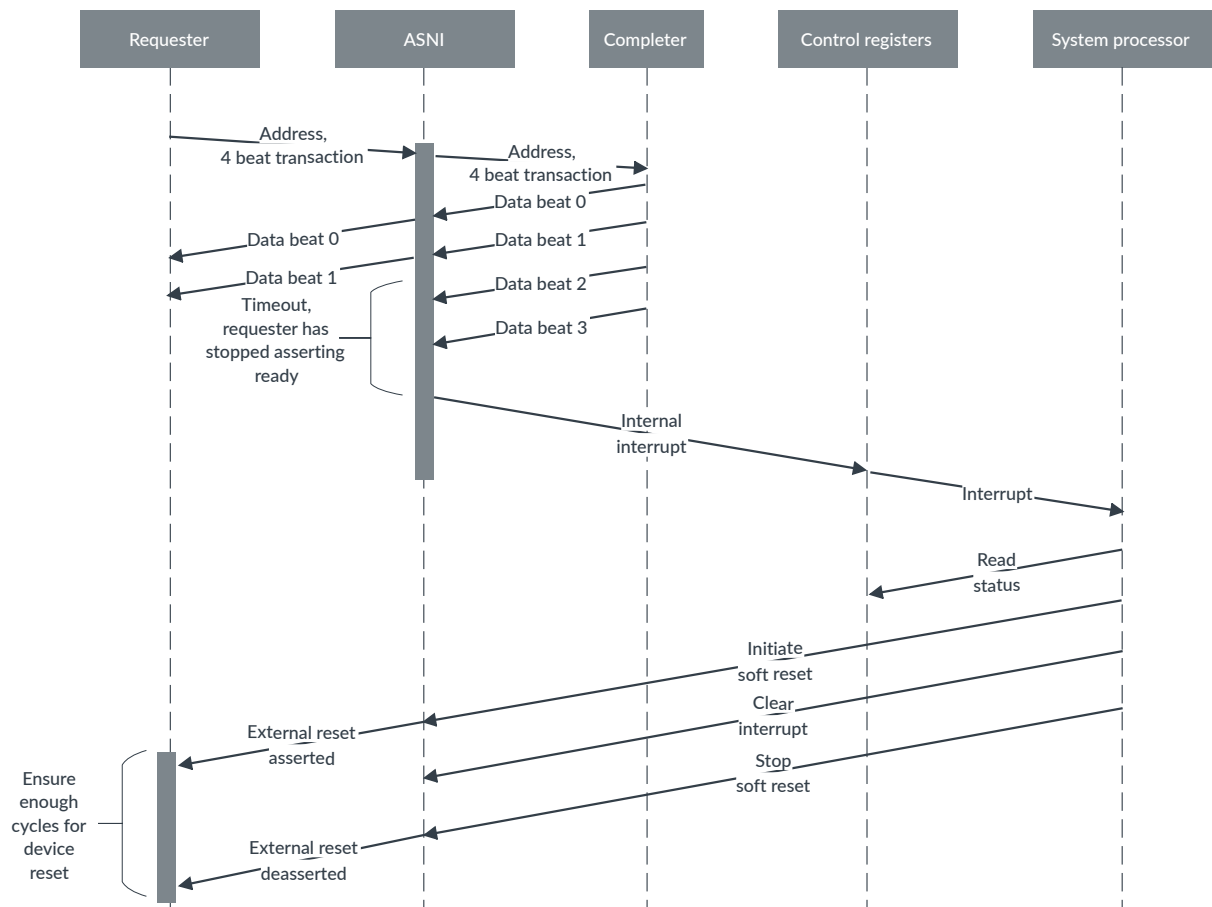
This example demonstrates the expected use case for the IDM soft reset functionality for a read transaction at an xSNI. In this example, a requester device stops accepting read data beats following the second data beat.

After the programmed timeout period has elapsed, an interrupt is asserted for software to handle. If the `idm_reset_control.reset_control_auto` field is set to 1, on detection of the timeout,

hardware automatically enters soft reset mode to gate the external interface. The outstanding read data beats are sunk internally and the request is completed. After the software writes 1 to the `idm_reset_control.reset_control_auto` field to initiate the soft reset sequence for the endpoint, the external reset pin is also asserted. This assertion resets the attached stalled requester device.

The following figure shows the sequence of events for an ASNI stalled read transaction leading to a soft reset.

**Figure 10-3: ASNI stalled read transaction leading to a soft reset**



### xMNI write transaction timeout leading to soft reset

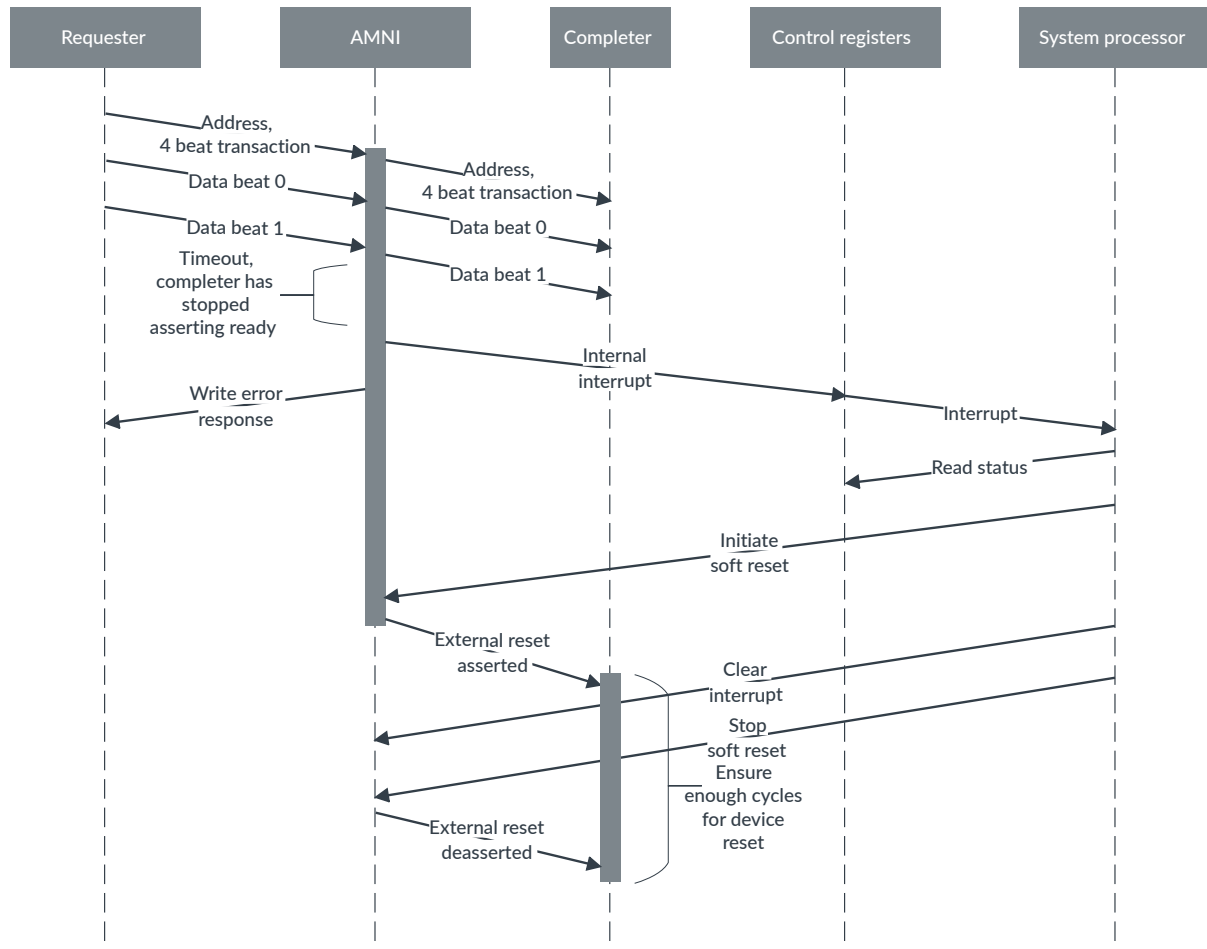
The next example demonstrates an expected use case for the IDM soft reset functionality for a write transaction at an AMNI. In this example, a completer device stops accepting write data beats after the second data beat.

After the programmed timeout value, an interrupt is asserted for software to handle. If the `idm_reset_control.reset_control_auto` field is set to 1, on detection of the timeout, hardware automatically enters soft reset mode to gate the external interface. The outstanding write data beats are sunk internally, a write response with error is generated, and the request is completed.

After the software writes 1 to the `idm_reset_control.reset_control_auto` field to initiate the soft reset sequence for the endpoint, an external reset pin is also asserted. This assertion resets the attached stalled completer device.

The following figure shows the sequence of events for an AMNI write transaction timeout leading to a soft reset.

**Figure 10-4: AMNI write transaction timeout leading to a soft reset**



### xMNI read transaction timeout leading to soft reset

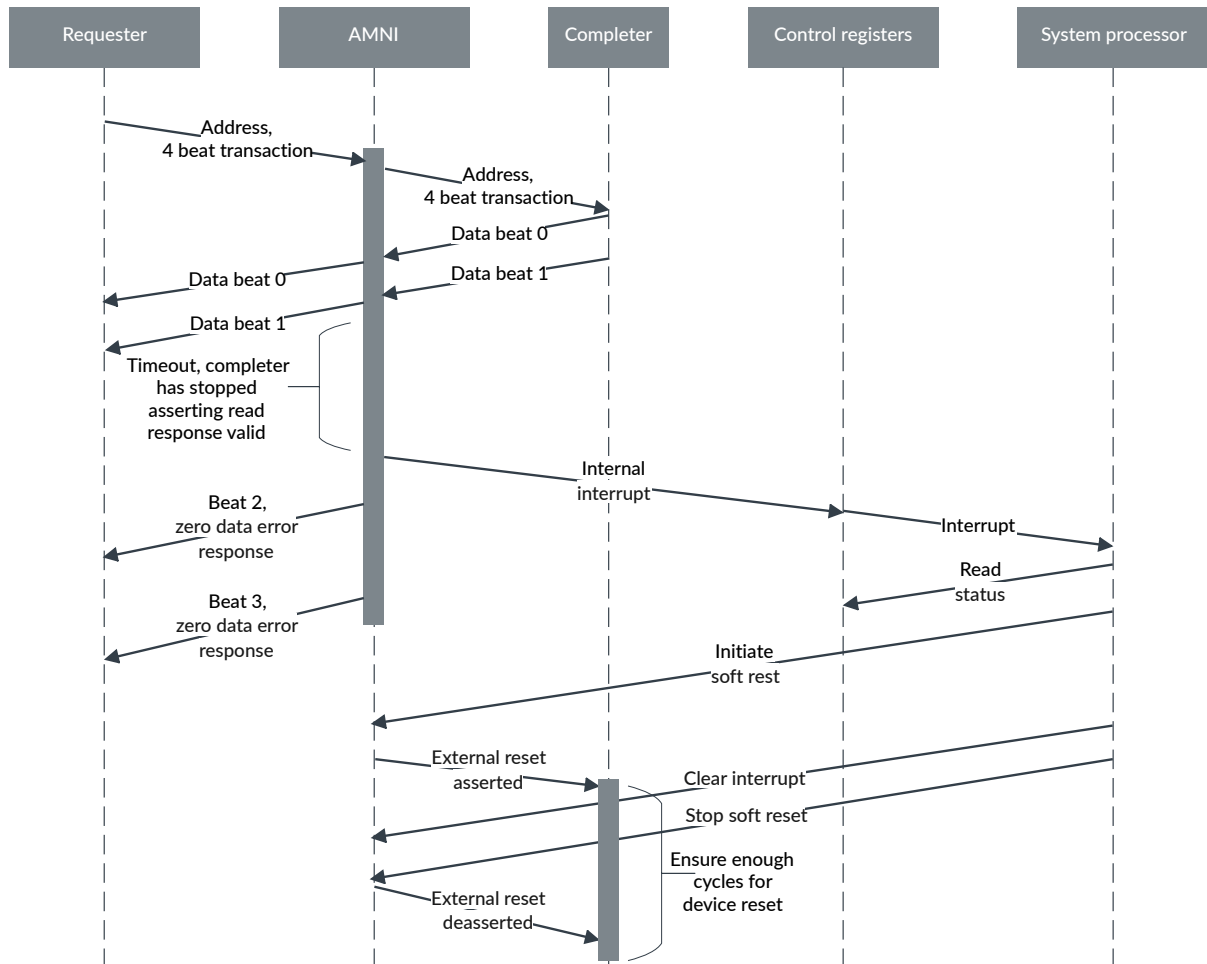
This example demonstrates an expected use case for the IDM soft reset functionality for a read transaction at an AMNI. In this example, a completer device stops issuing read data beats after the second data beat.

After the programmed timeout value, an interrupt is asserted for software to handle. If the `idm_reset_control.reset_control_auto` field is set to 1, on detection of the timeout, hardware automatically enters soft reset mode to gate the external interface. The outstanding read data

beats are synthesized with zero data and with an error response. After the software writes 1 to the `idm_reset_control.reset_control_auto` field to initiate the soft reset sequence for the endpoint, the external reset pin is also asserted. This assertion resets the attached stalled completer device.

The following figure shows the sequence of events for an AMNI read transaction timeout leading to a soft reset.

**Figure 10-5: AMNI read transaction timeout leading to soft reset**



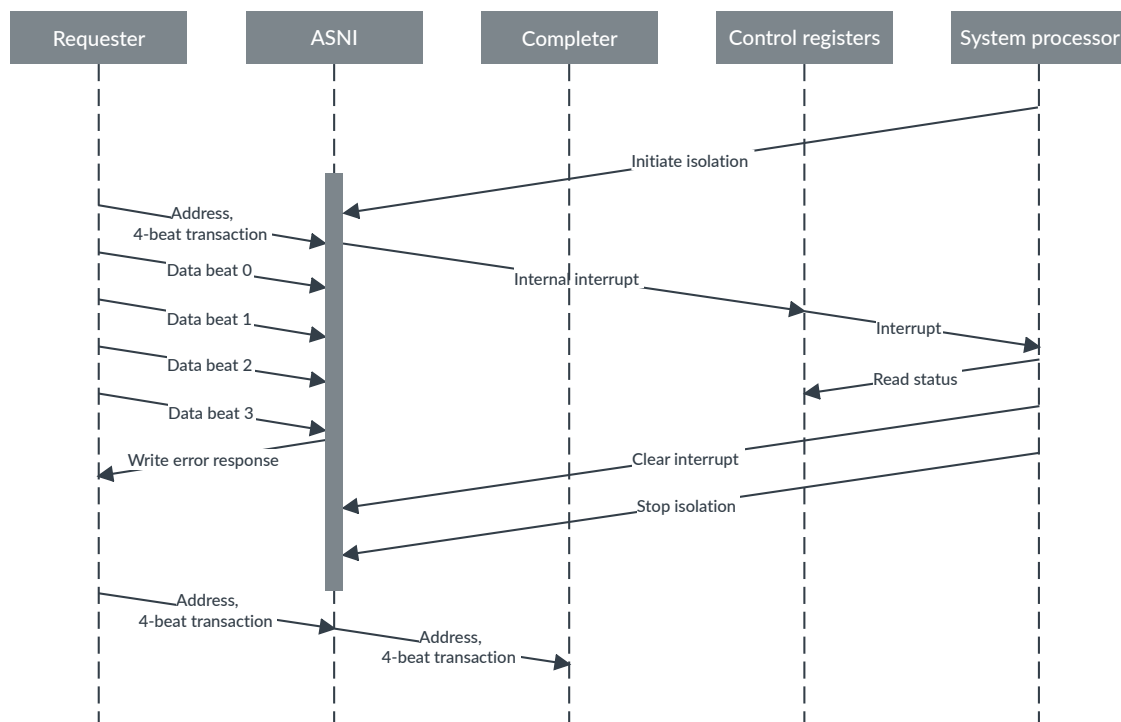
## 10.7 Access control use case example for xMNIs and xSNIs

These examples demonstrate the expected use cases for the IDM access control functionality, when a write transaction arrives at the interconnect from an isolated device.

In the first example, a requester device has been isolated from the interconnect and issues a new transaction. After the transaction arrives at the interconnect, an error response is generated and an interrupt is asserted for software to handle. The software then removes the isolation state and allows the transaction to complete later.

The following figure shows a write transaction arriving at a completer interface while the requester device is in the isolation state:

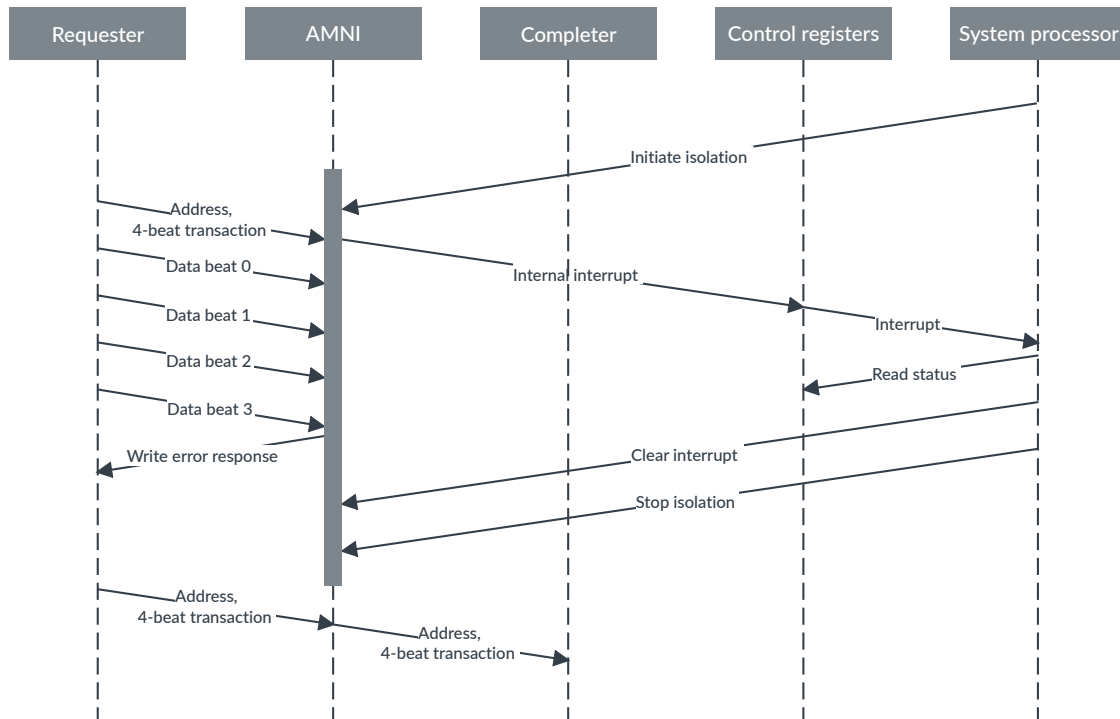
**Figure 10-6: ASNI write transaction arriving from an isolated requester**



The next example demonstrates the expected use case for the access control functionality for a write transaction at a requester interface. In this example, a completer device has been isolated from the interconnect and the AMNI connected to the completer device receives a new transaction. After the transaction arrives at an interconnect, an error response is generated and an interrupt is asserted for software to handle. The software then removes the isolation state and permits the transaction to complete later.

The following figure shows a write transaction arriving at a requester interface while the completer device is in the isolation state:

**Figure 10-7: AMNI write transaction arriving from an isolated completer**



## 10.8 Example interrupt handling sequence

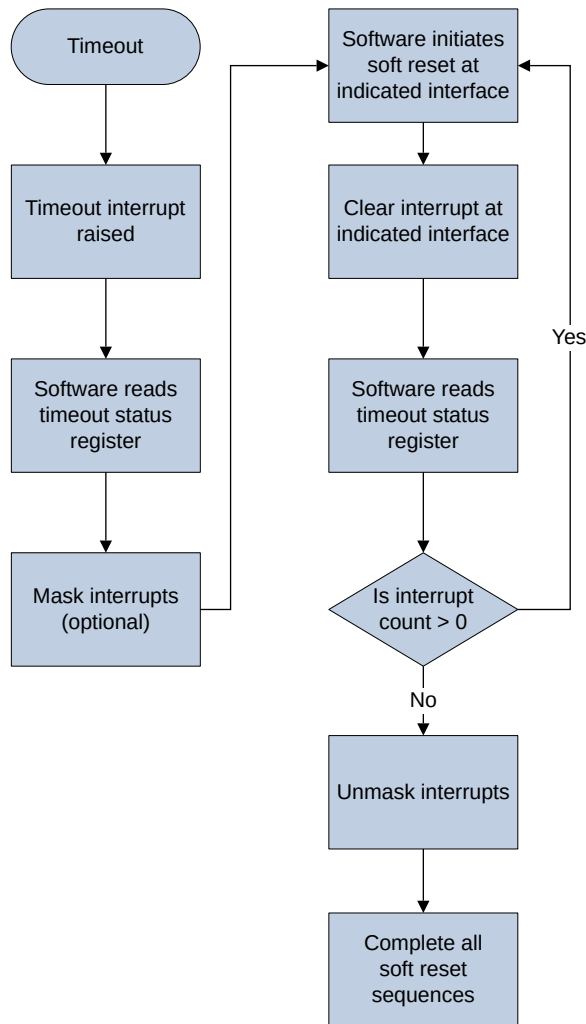
This example demonstrates the interrupt handling sequence when an interface timeout interrupt is asserted.

The software can read the status register to determine the interface that has asserted the interrupt and if there are multiple assertions. If necessary, all further interrupts can be masked. For this example, the interface indicating a timeout is placed in a soft reset state while its interrupt is cleared. The timeout interrupt status register is checked again to determine if any more interface interrupts require servicing. Once all interrupts have been serviced, the software brings all interfaces out of soft reset.

The following figure demonstrates a sample interrupt handling sequence.



**Figure 10-8: Interrupt handling sequence**



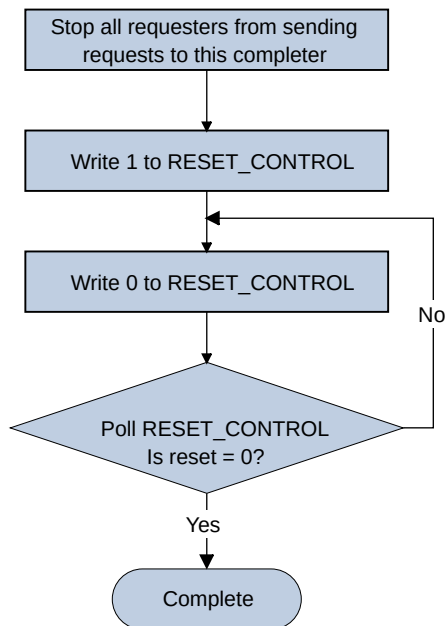
## 10.9 Soft reset sequence

This example shows a fast sequence for placing a completer device into soft reset.

The software is expected to first stop all the requesters that you expect are accessing that completer device. If this stop does not happen, it might not be straightforward for the software to leave soft reset at a later stage.

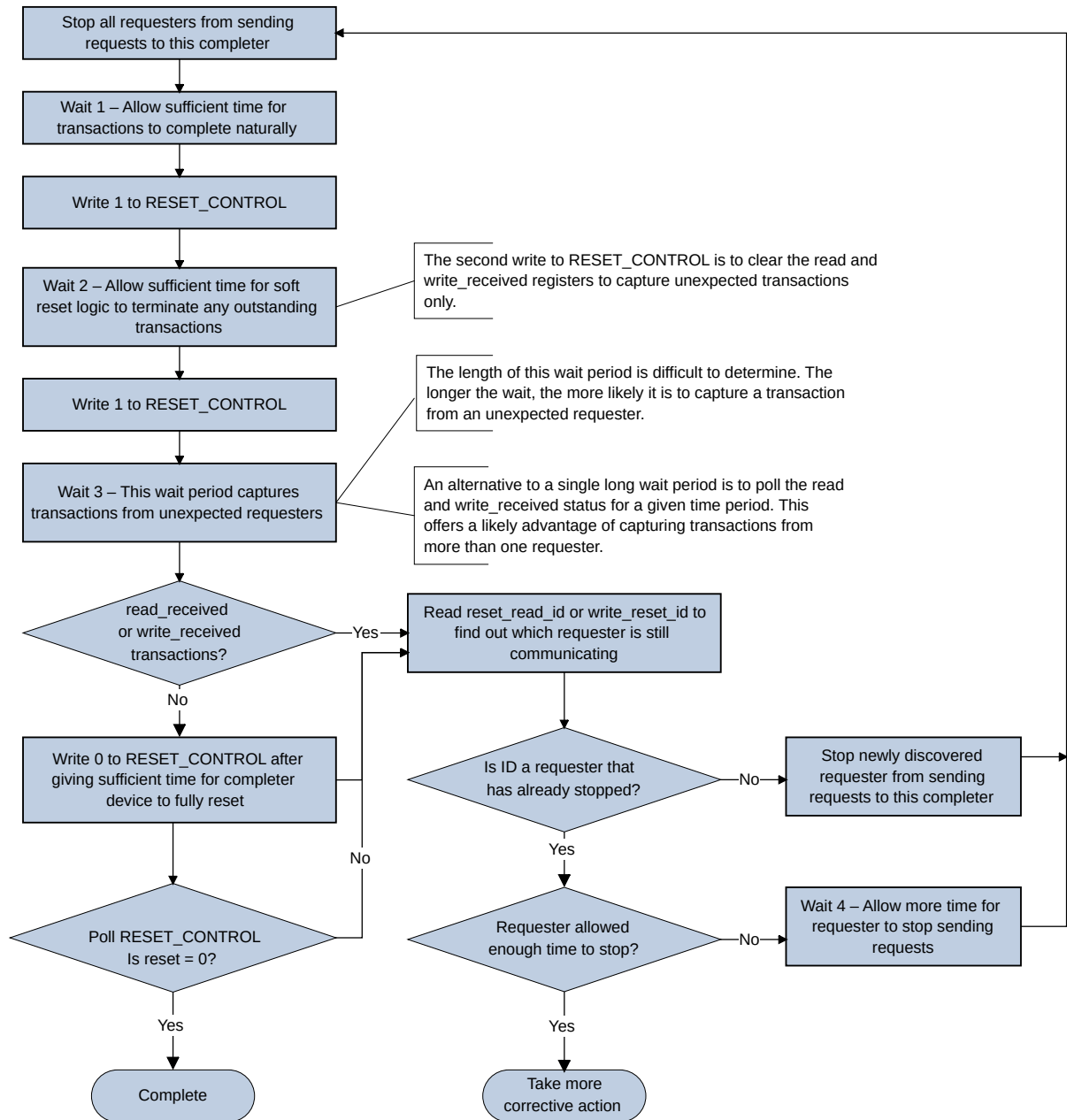
The following figure shows the soft reset sequence to place a completer device into soft reset.

**Figure 10-9: Completer device soft reset sequence**



Although the preceding sequence is the expected primary use case model, software can use a more cautious sequence. The following figure shows this more cautious sequence.

**Figure 10-10: Software using a more cautious interrupt handling sequence**



Similar software sequences can apply to IDM access control.

# 11. Address decode and mapping

When a controller device generates a request, the address is presented to the connected xSNI. The xSNIs have address decoders, which decode the address and maps the request to a target ID for correct transaction routing.

## 11.1 ASNI address decode

When an AXI requester device generates a request, the connected ASNI receives the transaction address through the request channel. The ASNI decodes the address and calculates the target ID for that address region.

The ASNI address decoders are generated when you configure the ASNI through Socrates. Separate address decoders exist in the ASNI for the read and write request channel, enabling parallel lookup. If an address pointing to an unmapped region of memory is presented to the address decoder, an address DECERR response is generated.

## 11.2 HSNI address decode

When an AHB requester device generates a request, the connected HSNI receives the transaction address through the request channel. The HSNI decodes the address and calculates the target ID for that address region.

When you use Socrates to configure the HSNI, you generate the HSNI address decoders at the same time. A single address decoder exists in the HSNI as read and write requests come on the same channel. If an address pointing to an unmapped region of memory is presented to the address decoder, it generates an address DECERR response.

## 11.3 PMNI address decode

When an AXI or AHB requester generates a request, the connected ASNI or HSNI receives the transaction address through the request channel. The ASNI or HSNI decodes the address and calculates the target ID for that address region.

As described in [PMNI](#), each PMNI can have up to 16 APB interfaces behind it. If the target ID from the address decode corresponds to a PMNI, the target ID includes information that encodes the exact APB interface behind the PMNI. Correspondingly, the address map in the ASNI and HSNI has address regions defined for each APB interface behind every PMNI instance.

## 11.4 Address striping

NI-710AE supports transaction address striping. The ASNI handles address striping as it decodes a transaction address.

An NI-710AE configuration must obey the following constraints for address striping:

- You define a stripe group by the number of stripe targets that are part of it and the striping granularity.
- NI-710AE supports the following address stripe granule sizes:
  - 128 bytes
  - 256 bytes
  - 512 bytes
  - 1024 bytes
  - 2048 bytes
  - 4096 bytes
- NI-710AE supports stripe groups which have one, two, or four target interfaces. When there is a stripe group with a single target, all requests to that striped region are sent to the same target. However, the requests are split based on the specified stripe granularity.
- The target interfaces that are part of a stripe group must all be AMNIs or HMNIs.
- All AXI and ACE-Lite properties must be the same for all the AMNIs that are part of the same stripe group.
- All AHB properties must be the same for all the HMNIs that are part of the same stripe group.
- PMNIs cannot be part of a stripe group.
- It is the responsibility of the SoC integrator and system builder to set up the address maps and stripe groups consistently.

There are several address map restrictions regarding stripe groups:

- Two different stripe groups can have different striping granularity or number of stripe targets or both.
- Two different address regions in an address map can point to one or both of the following options:
  - Two different stripe groups with different stripe granularities
  - A different number of stripe targets
- The default and remap target, or two different remap targets of an address region in an address map can point to two different stripe groups. The two different stripe groups can have two different stripe granularities or different number of stripe targets or both.

### Address Hash Function

Two stripe targets:

- Mask off the lower bits based on the stripe granularity.

- XOR all the remaining address bits to generate a single bit. 0 or 1 determines the stripe target.

Four stripe targets:

- Mask off the lower bits based on the stripe granularity.
- Generate a 2-bit stripe select to cover four targets:
  - Even stripe select. XOR all the remaining even address bits to generate a single bit.
    - Even stripe select drives bit Select[0].
  - Odd stripe select. XOR all the remaining odd address bits to generate a single bit.
    - Odd stripe select drives bit Select[1].

## 11.5 Remap

Registers in the programmers model control the remap functionality.

The address decoder supports up to eight remap states, which are programmed using the address remap vector register. The system must be in a quiesced state before programming the address remap vector register. The BRESP response for the configuration writes to the address remap vector register confirms that the register write is complete.

After a write to the address remap vector register occurs, further transactions must only be issued after receiving BRESP. This constraint ensures that the interconnect maps transactions correctly.

For more information, see the [Programmers model](#). You can define the remap states using 8 bits of the remap register. A bit in the remap register controls each remap state.



You can use each remap state to control the address decoding for one or more target interfaces. If two remap states that are both asserted affect a target interface, the remap state with the lowest number takes precedence.

---

You can configure each target interface independently so that a remap state can perform different functions for different controllers.

A remap state can:

- Change the target controller interface for an address region. The target can change from:
  - One target to a different target
  - A single target to a stripe group
  - One stripe group to a different stripe group
  - A stripe group to a single target
  - Point to no target, that is, provide a DECERR
- Remove an address region

- Add an address region

Because of the nature of the distributed register subsystem, the controllers receive the updated remap bit states in sequence, and not simultaneously.

The following figures show examples of how different remap states interact with each other. These examples represent the two bottom address ranges of the memory map. The remap bits correspond to these ranges.

While NI-710AE can support up to eight remaps, consider an example configuration that uses three remap bits. The following figure shows the memory map when you set the remap value to 0b000, representing no remap.

**Figure 11-1: No remap, remap set to**

Target 2
Target 1
Target 0 region 1
Target 0 region 0
Target 3 region 1
Target 0 region 0

In the following figure, there is a default memory map that divides target 0 and target 3 into two separate regions. In this example, you can choose to set up a remap value whereby target 3 is aliased over target 0, using the remap code 001. At powerup, target 0 region 0 is aliased over target 3 region 0. After powerup, the target 0 region 0 alias is removed as shown.

**Figure 11-2: Remap set to**

Target 2
Target 1
Target 0 region 1
Target 0 region 0
Target 3 region 1
Target 3 region 0

Alternatively, you might decide to move target 1 to the bottom of the address range by setting remap to 010 as the following figure shows.

Figure 11-3: Remap set to

Target 2
Target 0 region 1
Target 0 region 0
Target 1

You can choose to remove entire target regions. The following figure shows that if you set remap to 100, target 3 is removed.

Figure 11-4: Remap set to

Target 2
Target 0 region 1
Target 0 region 0

Remap bit 0 still takes precedence if you set it as the following figure shows.

Figure 11-5: Remap set to

Target 2
Target 0 region 1
Target 0 region 0
Target 1
Target 3 region 0

In addition, you can choose to remove entire memory regions. The following figure shows that if you set remap to 101, target 3 and target 1 are removed.



Figure 11-6: Remap set to

Target 2
Target 0 region 1
Target 0 region 0
Target 3 region 0



Caution

When you define the address map and remap, ensure you maintain access to the NI-710AE programmers model space from at least one ASNI or HSNI. If you do not maintain access, you cannot access the NI-710AE programmers model to change the address remap option or access any other configuration register.

Therefore, the default target of the configuration address region from at least one ASNI or HSNI, must point to the configuration target. No address region can map to the configuration target except the configuration address region aligned with PERIPHBASE. The default and remap targets of the configuration address region can only be one of two values, configuration target or no target. To program the address\_remap register in ASNI or HSNI to choose a specific remap, see [ASNI address\\_remap register](#) and [HSNI address\\_remap register](#).

## 12. Transaction tracking and ordering

A transaction deadlock can occur when routing multiple transactions concurrently to multiple completers from a point of ingress to the interconnect, such as a completer interface. To prevent such a deadlock, each NI-710AE ASNI uses either or both of two mechanisms.

ASNI uses the following mechanisms to prevent transaction deadlocks:

- A configurable [transaction reorder buffer](#)
- A single completer for each ID [Cyclic Dependency Avoidance Scheme \(CDAS\)](#) mechanism

### 12.1 Transaction reorder buffers

To prevent issues when reordering AXI transactions, the ASNI supports transaction reorder buffers. NI-710AE contains a write response reorder buffer, which is always present, and an optional configurable read data reorder buffer.

#### Write response reorder buffer

To improve performance, a write response reorder buffer is always present for write transactions.

#### Read data reorder buffer

You can configure an optional read data reorder buffer of 1–256 data beats. This option enables a limited number of outstanding requests with the same ID to different destinations.

Responses that are received out-of-order are buffered internally to the ASNI until correct response ordering can be guaranteed. If there is insufficient capacity in the reorder buffer for the total number of read data beats of a transaction, the ASNI uses a single completer for each ID.

A read reorder buffer entry is allocated on a per transaction basis. This allocation only occurs when required, because of a change in destination for the same traffic ID. In this case, the number of entries reserved is equal to the length of the transaction that reuses the ID.

Read reorder buffer slots are also used to merge partial read responses. This process occurs when read response data beats come from xMNIs that have a data width less than the data width of the ASNI. In this case, NI-710AE merges the partial read responses in the same entry of the read reorder buffer. Merging these responses creates a full sized data beat at the ASNI. One read reorder buffer slot is reserved for each transaction outstanding at a xMNI with a smaller data width.

### 12.2 Cyclic Dependency Avoidance Scheme

The AXI protocol permits reordering of transactions, therefore it can be necessary for NI-710AE to enforce rules to prevent a transaction deadlock when routing transactions. NI-710AE uses a Cyclic Dependency Avoidance Scheme (CDAS) to prevent transaction deadlock.

The same CDAS mechanism operates independently for read and write transactions.

### 12.2.1 Single completer for each ID

A single completer for each ID ensures that for an ASNI, the following transactions go to the same destination:

- All outstanding read transactions with the same ID.
- All outstanding write transactions with the same ID, when there are non-modifiable accesses to striped regions and when Ordered Write Observation (OWO) is enabled. Otherwise outstanding write transactions with the same ID do not follow a single completer for each ID.

When the ASNI receives a read transaction that has an ID that:

- Does not match any outstanding transactions, it passes the CDAS
- Matches the ID of an outstanding transaction, and the destinations also match, it passes the CDAS
- Matches the ID of an outstanding transaction, and the destinations do not match, it fails the CDAS check, and is stalled

A stalled transaction remains stalled until one of the rules passes.

AXI non-modifiable transactions which access a striped region, must follow a single completer for each ID. That is, if another outstanding transaction has the same ID then it waits for its response to return before it sends out the next one.

### 12.2.2 Ordered Write Observation

If all other agents in NI-710AE observe two write transactions with the same ID and in the same order that the transactions are issued, then an interface can be declared as providing Ordered Write Observation (OWO).

NI-710AE contains its own logic to check for any outstanding transactions with the same ID for write. If there are any outstanding transactions with the same ID for write and OWO is enabled, then the interface works in single completer for each ID mode.

If consecutive writes happen to go to the same target, then the interface sends the requests back to back with full throughput.

If the next write has the same ID, and there is a previous outstanding write to a different destination with the same ID, then the interface waits to receive the BRESP signal before it sends the write.

## 13. Traffic arbitration schemes

To ensure optimal service is provided to all requesters and completers in a system configuration with a shared interconnect, NI-710AE provides configurable traffic arbitration schemes. These schemes arbitrate between different traffic sources and traffic types and help to manage access to shared system components and paths.

NI-710AE supports assignment of traffic generators to different Resource Planes (RPs). The network manages the access of each RP to shared system resources so that the RPs do not block each other. For more information, see [Resource planes](#).

NI-710AE also uses Quality of Service (QoS) values to arbitrate between traffic of different priority values. For more information, see [Quality of Service](#).

Optionally, NI-710AE supports propagation of Memory System Resource Partitioning and Monitoring (MPAM) signals through the interconnect. For more information, see [Memory System Resource Partitioning and Monitoring](#).

### 13.1 Resource Planes

Resource Planes (RPs) help reduce congestion and blocking between traffic streams that share resources.

Use RPs to help distribute and prioritize traffic flows across connections and end to end paths. Most network components support up to four resource planes, and provide time-multiplexed Virtual Channels (VCs) on a connection link. Each downstream interface, or network initiator, can be assigned to a specific RP. The RP is then applied to all routes originating from the initiator.

Use RPs to help prioritize certain traffic paths through shared resources. For example, two downstream interfaces share routes. One handles high-bandwidth traffic, for example, graphics data and one handles latency sensitive traffic, for example, CPU data. By default both are assigned the same RP, so they compete for bandwidth. However, assigning one of the interfaces to a different RP potentially prevents congestion between different traffic classes in the system. RPs provide the system designer with a solution to support non-blocking flows between different traffic classes and fix potential deadlock situations.

You can configure up to four RPs.

## 13.2 Quality of Service

Throughout NI-710AE, arbitration nodes decide the order of progression for transactions according to priority. These nodes use the Quality of Service (QoS) value of a transaction as it passes through the interconnect to inform arbitration decisions.

QoS regulation features are also available at traffic injection points in the network. You can use these features to program the behavior of NI-710AE during periods of high network traffic.

NI-710AE QoS includes the following features:

- Configurable QoS options for ASNIs.
- Regulation of read and write requests.
- Programmable QoS facilities for attached upstream devices with AMBA interfaces.
- VAXQOSACCEPT[3:0] signaling. These requests stall transactions with a QoS priority that is less than the current qosaccept value.

At any arbitration node, there is a fixed priority for transactions with a different QoS. Transactions with the highest QoS value have the highest priority. If there are coincident transactions with the same QoS value that require arbitration at a node, then the network uses a Least Recently Used (LRU) algorithm.

As a side-effect of QoS, starvation can occur when streams of traffic with high QoS priority block low QoS priority transactions from progressing. To avoid starvation, NI-710AE arbitration nodes can be configured to ignore the QoS priority of a set proportion of arbitration decisions. For example, the network can be configured to permit one in every 16 transactions to pass regardless of the QoS priority.

NI-710AE supports two types of QoS bandwidth regulation:

### Hard bandwidth regulation

You can program NI-710AE to block new traffic based on the number of outstanding transactions or based on an acceptable traffic profile that you define. For more information, see [Hard bandwidth regulation](#).

### Soft bandwidth regulation

You can program NI-710AE to reduce the QoS value of new traffic when the bandwidth limit is exceeded rather than just blocking new transactions. For more information, see [Soft bandwidth regulation](#).

### 13.2.1 Hard bandwidth regulation

You can apply hard bandwidth regulation to the points of network traffic injection in the NI-710AE, such as the ASNIs.

The NI-710AE supports the following types of QoS hard bandwidth regulators:

- Outstanding Transaction (OT) regulators

- Traffic Specification (TSPEC) regulators

Hard bandwidth QoS regulators block new network traffic according to two constraints:

- The number of transactions that are awaiting a network response
- An upper bandwidth limit that is applied to the request channels on the downstream interface

### 13.2.1.1 Outstanding transaction regulation

The NI-710AE ASNI tracks outstanding read and write requests that are submitted at its completer interfaces.

You can program the tracking logic to constrain the maximum number of outstanding requests. This feature is known as Outstanding Transaction Quality of Service (OT QoS) regulation.

The ASNI tracks all outstanding requests that it receives until it has received the correct number of response GT packets. To reduce congestion within the interconnect, you can constrain request numbers by programming OT QoS regulators.

There are three types of OT QoS regulators that you can configure at the ASNI:

#### **Read regulator**

Tracks outstanding read requests

#### **Write regulator**

Tracks outstanding write requests

#### **Combined regulator**

Tracks both read and write requests

Each OT regulator has an 8-bit programmable register value. If the number of outstanding requests equals the programmed regulator value, then new requests from the corresponding channel are stalled. Requests also stall if the number of outstanding requests exceeds the regulator value because of reprogramming. Split Bursts count as multiple entries.

The minimum programmable value for each regulator to maintain OT regulation is 1. Programming an OT regulator to zero or more than issuing capability results in no OT regulation.

The OT regulator registers are visible to system software to help performance debug.

### 13.2.1.2 Traffic specification regulation

The NI-710AE supports hard bandwidth regulation through TSPEC QoS regulators. This regulator is configured in the ASNI and HSNI units.

Multiple TSPEC QoS regulators are present within the ASNI unit. You can configure the ASNI to include TSPEC regulators for the individual AXI read and write request channels, and a combined TSPEC regulator. You can only configure the HSNI to include a combined TSPEC regulator, as AHB only has a single channel.

When programming the TSPEC regulators, you define a limit on the acceptable network traffic profile. The following table describes each parameter you can configure for the TSPEC regulators.



Transfers in the following parameter descriptions correspond to data beats in read and write transactions.

**Table 13-1: Acceptable network traffic profile limit**

Parameter	Description
r_value	Average number of transfers for each cycle
p_value	Peak number of transfers for each cycle
b_value	Burstiness allowance (the amount of data bandwidth more than the average data bandwidth)

The r\_value and the p\_value, represent the rate of transfers as a fraction of the maximum bandwidth of the port. The r\_value and the p\_value are programmed as fractions represented in binary values, see the following examples.

The b\_value represents the total number of transfers that are allowed to be sent above the average rate (r\_value). The value is loaded to a counter. Once the counter of permitted transfers is zero, the regulator restricts bandwidth to the exact average rate (r\_value). If the port bandwidth drops below the average rate (r\_value), then this drop permits all or part of the burstiness allowance to be accepted in addition to the average rate. However this scenario depends on the duration of the low bandwidth or idle window.



The port bandwidth is limited by the peak rate (p\_value) at any time.

The regulator measures the incoming channel transfer rates and compares them against programmed parameters. Outputs from the TSPEC regulator are used to enforce hard regulation by gating address channel handshake signals. Therefore, incoming requests are blocked until the channel is within specification.

Transactions are stalled if one of the following conditions is met:

1. The total number of transfers exceeds the average number of transfers, plus the burstiness allowance.
2. The data rate exceeds the peak number of transfers for each cycle.

### 13.2.1.2.1 Calculating TSPEC parameters for traffic

NI-710AE supports several Traffic Specification (TSPEC) parameters for tracking and regulating traffic. These parameters are *r\_value* (average number of transfers for each cycle), *b\_value* (burstiness allowance), and *p\_value* (peak number of transfers for each cycle).

#### Calculating the average number of transfers for each cycle (*r\_value*)

The following example shows how to program the TSPEC regulator to restrict an upstream device from issuing no more than 40MB/s traffic. If the upstream device has a data width of 64 bits and a clock frequency of 100MHz, the max bandwidth of the port is:

$$100 \times 8 = 800\text{MB/s}$$

To restrict the bandwidth to 40MB/s, the limit on this interface must correspond to an average transfer rate of:

$$40 \text{ MBs} / (100 \times 8) = 0.05$$

In other words, 5% of the maximum bandwidth of the 64-bit interface at 100MHz. The value of the 0.05 fraction in binary is 0b00001100110011001101. However, since the average rate register field is six bits, that corresponds to setting the *qosrdavg* register to 0b000011 (six most significant bits). Effectively, the *r\_value* parameter is programmed as:

$$(8 \times 100 / 2^6) \times \text{qosrdavg register value} = 37.5\text{MB/s}$$

So, a rounding error is introduced because of the six bits of granularity of the register.

#### Calculating the burstiness allowance (*b\_value*)

The burstiness allowance is useful when the traffic from the upstream device follows a uniform pattern, but contains some repeating patterns, burstiness, or both. The *b\_value* parameter allows you to set some level of burstiness variability over the average rate from that device. This parameter specifies an allowance of transfers that can be sent in addition to the average rate.

For example, if the incoming transfer rate matches the average rate, extra transfers can be sent until all these extra transfers are used. However, burstiness is not a one-time allowance. If the incoming transfer rate falls below the average rate, then all or part of the burstiness allowance can be accepted in addition to the average rate. The number of extra transfers that are accepted depends on the duration of the low bandwidth or idle window.

For the 64-bit 100MHz upstream device in the previous example, the *r\_value* setting of 37.5MB/s allows the device to send one transfer approximately every 21 cycles:

$$8 \times 100 / 37.5 = 21.3$$

When the burstiness allowance is set to 0, the regulator enforces the injection rate given in the preceding equation. But with a nonzero *b\_value* parameter, the regulator allows extra transfers to be sent in the monitoring window, according to:

$$(\text{r\_value} \times \text{total cycles}) + \text{b\_value}$$



The `b_value` register is 14 bits wide, so the burstiness allowance can be up to `0x3fff` more transfers than the average rate over the monitoring window.

### Calculating the peak number of transfers for each cycle (`p_value`)

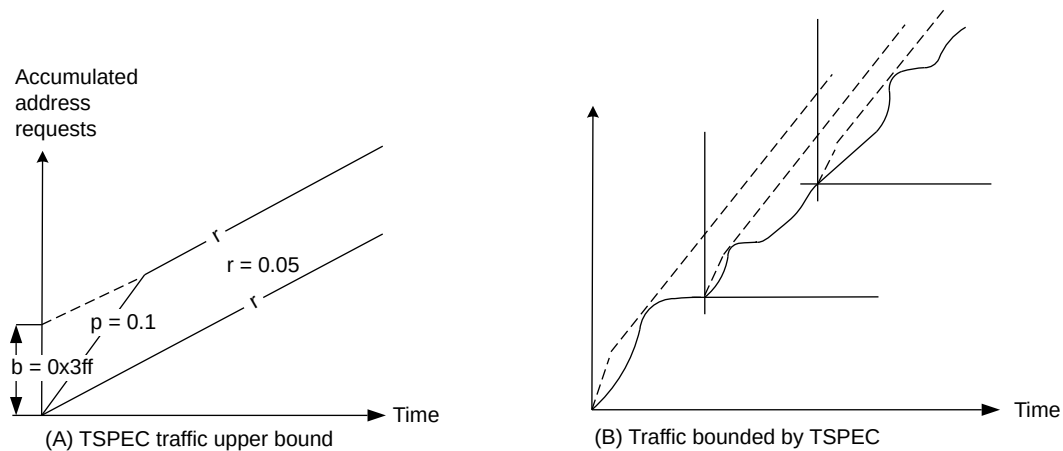
The peak rate setting defines an upper limit on bandwidth from the upstream device and is set as a fraction, similarly to the `r_value` parameter. Imposing an upper limit by using the `p_value` setting is useful when there is considerable burstiness in the traffic and a large burstiness allowance is set.

Again using the 64-bit 100MHz upstream device example, assume the `b_value` parameter is set to the maximum value of `0x3fff` and the `p_value` setting is 80MB/s. With a maximum bandwidth of 800MB/s, this peak rate setting corresponds to 10% of the maximum bandwidth, which is 0.1 times the channel width. Represented as a binary fraction, this value is `0b000110`. With this `p_value` setting, the regulator allows transactions to be issued at a peak rate of 80MB/s until the burstiness allowance of 16383 transfers have been sent. After the burstiness allowance transfers have all been used, the regulator enforces a transfer rate of exactly 40MB/s until the burstiness allowance is available again.

#### 13.2.1.2.2 TSPEC parameter examples

These examples show how the different TSPEC parameters work together.

**Figure 13-1: TSPEC parameter examples**



**Example: Limit set to 50% of maximum interface bandwidth, no allowance for burstiness**

**Table 13-2: TSPEC parameter values for 50% of maximum interface bandwidth**

Parameter value	Description	Value
<code>r_value</code>	Average number of transfers for each cycle	0.5
<code>b_value</code>	Burstiness allowance	0
<code>p_value</code>	Peak number of transfers for each cycle	0

The `r_value` of 0.5 indicates that only 0.5 beats are permitted every cycle on average.

The rate of incoming transfers is tracked every cycle:

- If there is an incoming transfer, then there is an increment by the number of beats in the transfer
- Decrement by `r_value` beats to indicate allowed average rate

### Example: Dynamically determine when the next transfer is allowed to enter

The following examples show how to use the calculation to dynamically determine when the next transfer is allowed to enter. Since there is no burstiness allowance, incoming transfers are stalled if the counter is  $\geq r\_value$ .

In the following scenario, you have an incoming single beat transfer every alternate cycle to achieve a 50% bandwidth.

**Table 13-3: Incoming single beat transfer every alternate cycle**

Single beat transfers	C1	C2	C3	C4	C5	C6	C7	C8
Beats in incoming transfers	1	0	1	0	1	–	–	–
Transfers_nxt	0.5	0	0.5	0	0.5	–	–	–

In the following scenario, you have an incoming two-beat transfer every four cycles to achieve a 50% bandwidth.

**Table 13-4: Incoming two-beat transfer every four cycles**

Two-beat transfers	C1	C2	C3	C4	C5	C6	C7	C8
Beats in incoming transfers	2	0	0	0	2	0	0	0
Transfers_nxt	1.5	1	0.5	0	1.5	1	0.5	0

**Example: An average of 25% of the maximum interface bandwidth with some allowance for burstiness. There is also a peak bandwidth limit set to 50% of the maximum interface bandwidth**

**Table 13-5: Parameter values and descriptions**

Parameter value	Description	Value
<code>r_value</code>	Average number of transfers for each cycle	0.25
<code>b_value</code>	Burstiness allowance	2
<code>p_value</code>	Peak number of transfers for each cycle	0.5



Note

The `r_value` of 0.25 indicates that only 0.25 beats are allowed every cycle on average. The `b_value` permits a burstiness allowance of two beats but it is only an allowance. Whether the full value of the allowance can be used or not depends on the dynamic window. The `p_value` of 0.5 indicates that only 0.5 beats are permitted every cycle at peak.

The rate of incoming transfers is tracked every cycle to compare with the average rate:

- If there is an incoming transfer, then increments by the number of beats in the transfer

- Decrement by  $r\_value$  beats to indicate allowed average rate

Therefore,  $Transfers\_nxt = Transfers\_q + incoming\ beats - r\_value$ .

The rate of incoming transfers is tracked every cycle to also compare with peak rate:

- If there is an incoming transfer, then increments by the number of beats in the transfer
- Decrement by  $p\_value$  beats to indicate allowed peak rate

Therefore,  $Peak\_transfers\_nxt = Peak\_transfers\_q + incoming\ beats - p\_value$ .

The following example shows how the calculation is used to dynamically determine how to adjust when the next transfer is allowed to enter.

- Since there is a burstiness allowance, incoming transfers stall if  $Transfers\_nxt > \{b\_value, r\_value\}$
- Incoming transfers also stall if  $Peak\_transfers\_nxt \geq p\_value$

If either of the preceding conditions is true, incoming transfers are stalled.

The following tables show:

- Cycles C1–C8 and C25–C32 show the peak transfer rate which permits burstiness allowance number of transfers
- Cycle C9–C16 is the average transfer rate
- Cycle C17–C24 is the idle period

In the C1–C8 eight-cycle phase, four beats have been transferred which achieves a peak rate of 50%. There are also two extra beats ( $b\_value$ ) permitted over what would have been possible in the same eight cycle window, with an average rate of (25% = two beats in eight cycles).

**Table 13-6: Cycle 1–8, transfer of four beats in eight cycles (peak transfer rate)**

Two-beat transfers	C1	C2	C3	C4	C5	C6	C7	C8
Transfer	2	0	0	0	2	0	0	0
Transfers_nxt	1.75	1.5	1.25	1	2.75	2.5	2.25	2
Peak_transfers_nxt	1.5	1	0.5	0	1.5	1	0.5	0

At the end of the C1–C8 phase,  $transfers\_nxt$  is two ( $b\_value$ ), therefore in the C9–C16 phase we are able to send only two beats in eight cycles (25% =  $r\_value$ ). So after sending the allowed  $b\_value$  transfers that exceed the  $r\_value$ , the bandwidth is limited to average rate ( $r\_value$ ).

**Table 13-7: Cycle 9–16, transfer of two beats in eight cycles (average transfer rate)**

Two-beat transfers	C9	C10	C11	C12	C13	C14	C15	C16
Transfer	2	0	0	0	0	0	0	0
Transfers_nxt	3.75	3.5	3.25	3	2.75	2.5	2.25	2
Peak_transfers_nxt	1.5	1	0.5	0	0	0	0	0

Phase C17–C24 is the idle phase and therefore at the end of the phase, transfers\_nxt reaches 0. Phase C17–C24 permits phase C25–C32 to repeat and send four beats in eight cycles. The number of beats in each transfer determines the length of each of these phases, that is the r\_value, b\_value and p\_value. So, it is a dynamic window that adjusts each cycle.

**Table 13-8: Cycle 17–24 is the idle phase**

Two-beat transfers	C17	C18	C19	C20	C21	C22	C23	C24
Transfer	0	0	0	0	0	0	0	0
Transfers_nxt	1.75	1.5	1.25	1	0.75	0.5	0.25	0
Peak_transfers_nxt	0	0	0	0	0	0	0	0

**Table 13-9: Cycle 25–32, transfer of four beats in eight cycles (peak transfer rate)**

Two-beat transfers	C25	C26	C27	C28	C29	C30	C31	C32
Transfer	2	0	0	0	2	0	0	0
Transfers_nxt	1.75	1.5	1.25	1	2.75	2.5	2.25	2
Peak_transfers_nxt	1.5	1	0.5	0	1.5	1	0.5	0

### 13.2.1.2.3 TSPEC registers and parameters

NI-710AE has programmable registers to configure the Traffic Specification (TSPEC) parameters for hard bandwidth regulation. The registers that you must use depend on the TSPEC mode you require.

NI-710AE supports configuring the TSPEC parameters for read-only, write-only, and read and write combined mode. In other words, you can set r\_value, b\_value and p\_value parameters for read and write channels separately or they can be combined.



HSNI only supports the combined regulator because AHB is a single channel.

When set for read and write channel separately, transfers from only that channel are used to determine if traffic is within specification.

In combined mode, transfers from both read and write channels are combined and are sent to the regulator. So combined rate of read and write channels is checked for being within specification.

The following table shows the registers used to program TSPEC parameters on different channels:

**Table 13-10: Registers used to program TSPEC parameters**

Register	Channel	Description
ASNI qosrdpk register	Read	Read hard bandwidth regulator peak rate (p_value)
ASNI qosrdavg register	Read	Read hard bandwidth regulator average rate (r_value)

Register	Channel	Description
ASNI qosrdbur register	Read	Read hard bandwidth regulator burstiness allowance (b_value)
ASNI qoswrpk register	Write	Write hard bandwidth regulator peak rate (p_value)
ASNI qoswavg register	Write	Write hard bandwidth regulator average rate (r_value)
ASNI qoswrbur register	Write	Write hard bandwidth regulator burstiness allowance (b_value)
ASNI qoscompk register	Combined read and write	Combined TSPEC bandwidth regulator peak rate register (p_value)
ASNI qoscombur register	Combined read and write	Combined TSPEC bandwidth regulator burstiness allowance register (b_value)
ASNI qoscomavg register	Combined read and write	Combined TSPEC bandwidth regulator average rate register (r_value)
HSNI qoscompk register	Combined read and write	Combined hard bandwidth regulator peak rate (p_value)
HSNI qoscomavg register	Combined read and write	Combined hard bandwidth regulator average rate (r_value)
HSNI qoscombur register	Combined read and write	Combined hard bandwidth regulator burstiness allowance (b_value)

## 13.2.2 Soft bandwidth regulation

NI-710AE ASNIs and HSNIs support Bandwidth QoS Value (BQV) QoS regulators. You can use BQV regulators to manage network traffic without restricting transaction requests from entering the network.

BQV regulators do not stall transactions from a particular channel when the programmed bandwidth allocation limit is exceeded. Instead, the regulator overrides the QoS value for transactions on that channel according to the amount of data that the channel has transferred. The QoS value of incoming transactions is reduced in proportion to the amount of excess bandwidth that the channel consumes.

The following table shows the parameters that are used for soft bandwidth regulation. Use the BQV control registers to program the parameters for each regulator.

**Table 13-11: BQV control parameters**

Parameter	Description
qv_max	Maximum QoS value for the channel. Used by default.
qv_min	Minimum QoS value for the channel.
overspend_per_qv	Number of excess transfers allowed for each QoS value, specified as a power of two.
bw_alloc	Threshold value for the average number of transfers in each cycle before QoS value reduction starts.
bw_burst	Number of extra transfers allowed above the average transfer threshold before QoS value reduction starts.

By default, the regulator uses the maximum QoS value of the channel.

The `bw_alloc` and `bw_burst` parameters are used to set the point at which bandwidth regulation starts. `bw_alloc` specifies the maximum value that is acceptable for the average number of transfers in a cycle. This value represents the maximum sustained traffic level that is permitted on a channel before the bandwidth is regulated.

The burstiness allowance, `bw_burst`, specifies extra transfers that can be used without triggering bandwidth regulation when the number of transfers in a cycle temporarily exceeds the `bw_alloc` setting. While the number of transfers in each cycle remains above the `bw_alloc` threshold, each

extra transfer is counted down from the `bw_burst` value until none remain. Only at this point does bandwidth regulation start. If the average number of transfers in a cycle drops below the `bw_alloc` threshold before the extra `bw_burst` transfers are all used, the counter is reset. The `bw_burst` parameter enables you to ensure that small, occasional spikes in traffic do not trigger bandwidth regulation.

The `bw_alloc` and `bw_burst` parameters function similarly to the `r_value` and `b_value` parameters that are used in TSPEC hard bandwidth regulation. For more examples of how to configure and program these parameters, see [Traffic specification regulation](#).

The channel limit specification is calculated as:

$$(\text{bw\_alloc} \times \text{number of cycles}) + \text{bw\_burst}$$

Whenever the total number of data transfers exceeds this limit, the extra transfers are divided by the allowed overspend, `overspend_per_qv`. The result is subtracted from the maximum QoS value for the channel, `qv_max_i`, to determine the reduced maximum QoS value for bandwidth regulation.

For example, if there are eight extra transfers over the specification and the allowed overspend is three, then the QoS value on the transaction reduces by:

$$[8 / (2^3)] = 1$$

That is, the regulated QoS value on the transaction is `qv_max_i - 1`.

The output QoS value, `qv_o`, can decrease to `qv_min_i` and rise back up to `qv_max_i`. This fluctuation occurs because the reduction in QoS value only depends on the current accumulated transfers through the average transfer rate.

As with TSPEC hard bandwidth regulation, you can configure ASNs with separate BQV parameter values for the read and write channels. ASNs also include registers that you can program to override incoming AxQoS values for the read and write channels. For more information, see [QoS value override programmable registers](#).

Alternatively, or in addition, you can configure ASNs with a single combined regulator to manage both the read and write channels according to the same specification. Because AHB only has a single channel, HSNs must be configured with the combined BQV regulator.

You configure soft bandwidth regulation by programming the BQV regulator registers.

**Table 13-12: BQV regulator registers**

Register	Description
QOSRDBQV	Read channel BQV regulator target bandwidth register for ASNs only
QOSWRBQV	Write channel BQV regulator target bandwidth register for ASNs only
QOSCOMBQV	Combined BQV regulator target bandwidth register for ASNs and HSNs

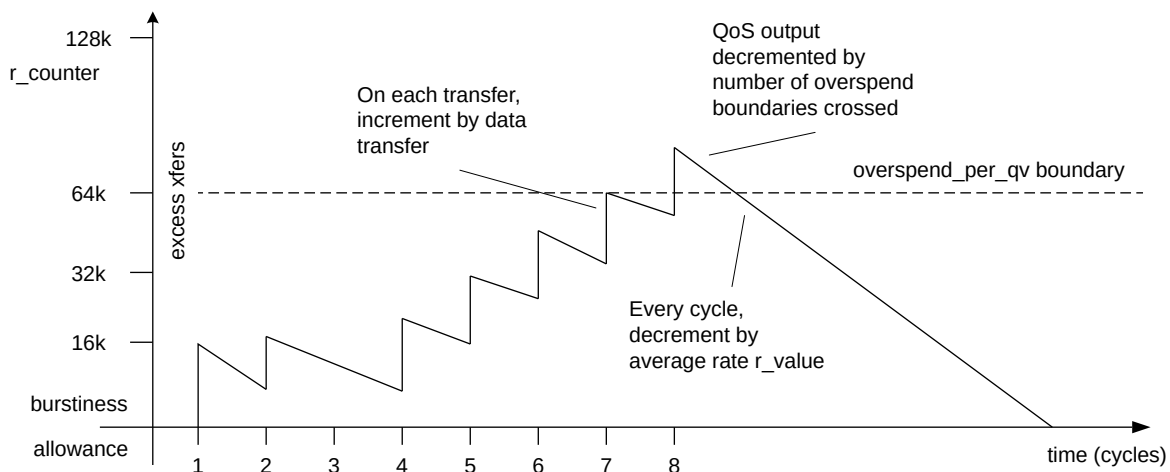
Each of the BQV registers contains the following fields, which are used to configure soft bandwidth regulation.

**Table 13-13: BQV regulator register fields**

Bit assignment	Field name	Description
[31:28]	BQV_OVRSPEND	Number of excess full data bus transfers permitted for the overspend allowance
[27:14]	BW_BURST	Number of extra full data bus transfers permitted for the burstiness allowance
[13:8]	BW_ALLOC	Threshold value for the average number of transfers in each cycle before BQV starts
[7:4]	QVMIN	Minimum QoS value for the channel
[3:0]	QVMAX	Maximum QoS value for the channel

The following figure shows how excess transfers above the average rate and the burstiness allowance determine when QoS value reduction starts.

**Figure 13-2: QoS value reduction triggering**



### 13.2.3 QoS value override programmable registers

NI-710AE provides a programmable method to override the incoming AxQoS value independently for the read and write channels by using specific ASNI registers.

The ASNI QoS value override registers are [ASNI arqos\\_value register](#) and [ASNI awqos\\_value register](#). To override the incoming AxQoS value, program these registers with the final QoS value that is applied to transactions when both of the following are true:

- The QOSOVERRIDE input signal bit is HIGH or the qos\_override\_enable bit of the [ASNI qosctl register](#) is HIGH
- The bq\_v\_enable bits of the ASNI qosctl register are not set

The following table shows how the final QoS value is determined.

**Table 13-14: Final QoS value matrix**

QoS override register	QoS override input signal	BQV regulators enabled	Combined and individual regulator QoS values	Final QoS value
0	0	0	–	AxQoS (unchanged)
–	–	1	Combined regulator QoS value higher than individual regulator QoS value	Determined by individual regulator
–	–	1	Combined regulator QoS value lower than individual regulator QoS value	Determined by combined regulator
0	1	0	–	QoS value from override register
1	0	0	–	QoS value from override register
1	1	0	–	QoS value from override register

## 13.3 Memory System Resource Partitioning and Monitoring

NI-710AE provides optional support for Memory System Resource Partitioning and Monitoring (MPAM) propagation. When MPAM support is enabled, you can override the MPAM values that are propagated through the network.

You can configure NI-710AE to include MPAM on GT flits, and also configure individual ASNI and AMNI units to support MPAM. When you enable MPAM on an ASNI or an AMNI, the interface must include the MPAM signal on all address channels. For more information about the MPAM signals, see [Signal descriptions](#).

If you enable MPAM support, NI-710AE also includes MPAM override registers for each address channel, which are included in the register block of every endpoint. These registers have various uses:

- Software can program the MPAM override register to override the MPAM value of a transaction with the value that is stored in the register.
- If you enable MPAM on GT flits, but not on a specific endpoint instance, then NI-710AE ignores whether override is enabled in the MPAM override register. The endpoint drives the override value from the MPAM override register onto the GT flit.
- HSNIs always drive the override value onto the GT flit when forwarding a transaction that targets an MPAM-enabled downstream device.

For more information about the MPAM override registers, see [Programmers model](#).



## 14. Performance monitoring

NI-710AE provides performance events which you can use to monitor and collect functional information about specific components during normal operation of the interconnect. The event information is collated and presented through the NI-710AE Performance Monitoring Unit (PMU) for your interpretation and analysis.

### PMU architecture and programming

The NI-710AE PMU consists of various counters and registers that are distributed across the clock domains. For more information about how these components are structured in the interconnect, see [PMU organization](#).

You set up and control the PMU by programming the PMU registers. For more information about the programmable functionality and programming requirements, see [PMU system programming](#).

### Performance monitoring events

Each endpoint in NI-710AE can generate and count various performance monitoring events during normal operation. For more information, see the following sections:

- [ASNI performance events](#)
- [AMNI performance events](#)
- [HSNI performance events](#)
- [HMNI performance events](#)
- [AHB performance event mapping](#)
- [PMNI performance events](#)

### Interpretation of performance monitoring events

Each performance monitoring event provides specific information about the performance of an endpoint during normal operation of NI-710AE. You can also combine certain events to give more information about the overall function of NI-710AE. For more information about interpreting the performance monitoring events, see the following sections:

- [Data bandwidth at ASNI and AMNI](#)
- [Data bandwidth at HSNI and HMNI](#)

## 14.1 PMU organization

The Performance Monitoring Unit (PMU) is distributed across each clock domain. Each clock domain contains software-visible event counters.

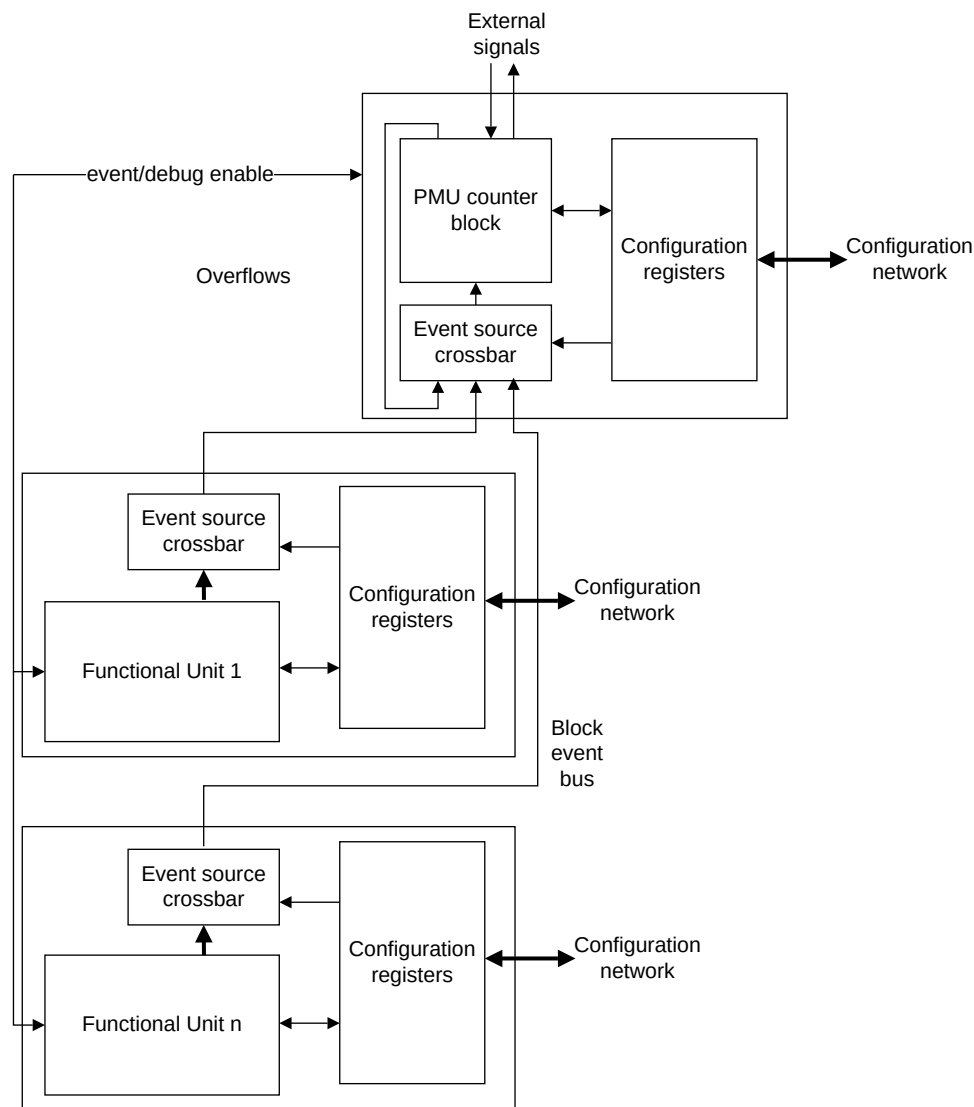
Within each clock domain, events are generated from several potential sources and multiplexed onto internal eight-bit event busses. These internal buses are in turn routed to the central set of software visible PMU counters for that clock domain. Each performance event counter

has a corresponding set of shadow snapshot registers to permit all counters to be sampled simultaneously and then read out in series.

The following figure shows the two-level hierarchical organization of the PMU. The configuration registers comprise the following counters, registers, and crossbar event selection:

- Software-visible event PMU counters
- Snapshot registers and other PMU control registers
- A configuration register for event selection in the event crossbar

**Figure 14-1: PMU hierarchical organization block diagram**



The first level of the hierarchy is at the level of the functional unit. Each functional unit, such as an individual ASNI or AMNI, can define up to 64 events. The PMU events are configured by programming the PMUSELA and PMUSELB registers in the node. See [AMNI register summary](#) and [ASNI register summary](#).

Event and Debug enable signals start the generation of events for a unit. An event source crossbar is configured through the programming interface to reduce the number of possible events, up to a maximum of eight events, minimizing top-level wiring. By programming two PMU event select registers, you can select up to eight events for publishing on an internal eight-bit event bus from each unit in that clock domain.

The individual event buses are routed to a centralized PMU counter block for each clock domain that has the second-level of the PMU event selection logic. The counter block consists of:

- A bank of eight 32-bit counters with overflow and snapshot functionality. These counters are responsible for counting the programmed events and giving memory-mapped read access to both the counters and counter snapshots.
- A programmable event source crossbar to permit selection of a particular event for a counter to monitor.
  - Each of the 8 bits of the internal event bus from each unit is routed to one of the eight counters with the matching index. For example, bit[0] to counter 0, and bit[1] to counter 1.
  - The second-level PMU event source crossbar supports PMU event type and filter registers. See [PMU register summary](#). These registers provide a programming interface that permits software to specify which unit event bus input each counter selects, according to type and source index.
- The event source crossbar can configure the PMU counters to trigger from the overflow of another counter within the PMU block. This feature permits extension of the counter range. For example, the crossbar can extend a single event counter up to a maximum 256-bit range with a single overflow.

## 14.2 PMU system programming

You can program the PMU event counters, snapshot functionality, and interrupts.

For specific register descriptions, see the [Programmers model](#).

### 14.2.1 Set up the PMU counters

To enable the Performance Monitoring Unit (PMU) to count specific events, you must set up the PMU event counters. You set up the PMU event counters by programming specific PMU-related registers that are located throughout the interconnect.

#### About this task

For PMU operation, NIDEN input must be asserted.

## Procedure

1. Program the \*\_PMUSELA/\*\_PMUSELB registers in the individual endpoints, for example ASNI and AMNI, to select the events that are published on the internal eight-bit event bus.
2. Program the eight PMEVTYPERn registers in the PMU block in every clock domain to program the PMU event source crossbar.
3. Write to the PMCNTENSET and PMCNTENCLR registers to enable specific PMU event counters.
4. Write to the PMINTENSET and PMINTENCLR registers to enable interrupts for the corresponding specific PMU event counters.
5. Use the PMCNTENSET register to reset cycle and event counters, to write to the PMCR register, and to enable PMU counting.  
This action enables all counters, whereas the PMCNTENSET and PMCNTENCLR enables specific counters.

## 14.2.2 Program PMU snapshot functionality

Program the PMU snapshot functionality to trigger a snapshot of PMU event counters.

### About this task

To trigger a snapshot of PMU event counters, use the following methods:

- Set the control bits in the PMSSCR register
- Use a four-phase handshake on the signals, as the following table shows

**Table 14-1: PMU snapshot signals**

Signal	Direction	Description	Clock relationship
<CLKNAME>_PMUSNAPSHOTREQ	Input	A four-phase request to initiate a snapshot of PMU event counters	Asynchronous
<CLKNAME>_PMUSNAPSHOTACK	Output	Acknowledgment of the PMU snapshot capture	Asynchronous



For PMU operation, NIDEN input must be asserted.

## Procedure

1. Program the PMU event counters. See [Set up the PMU counters](#).
2. Write 1 to the PMSSCR register to capture a snapshot of the contents of the PMU event counters, cycle counter, and overflow status.

After the PMU snapshot process has completed, the PMU block updates the PMSSR, PMOVSSR, PMCCNTSR both lower and upper, and PMEVCNTSRn registers. Software can poll the PMSSR register to check that the snapshot has completed.

## 14.2.3 Program PMU interrupts

Program PMU interrupts to identify cycle or event counters that have overflowed.

### About this task

If an event or cycle counter overflows, an interrupt is triggered. This interrupt is connected to the top-level interrupt, <CLKNAME>\_nPMUINTERERRUPT. You can determine the counter that has overflowed from the PMU control and configuration registers. These registers can also clear any counter overflow flags so that the interrupt can be cleared.

### Procedure

1. For PMU operation, NIDEN input has to be asserted.
2. Program the PMU counters. For more information, see [Set up the PMU counters](#).  
Any PMU counter overflow asserts <CLKNAME>\_nPMUINTERERRUPT. To determine the event counter or cycle counter that caused the interrupt, when observing assertion of <CLKNAME>\_nPMUINTERERRUPT, poll the PMOVSSR and PMOVSLR registers.
3. Write 1 into the corresponding PMOVSLR register to clear <CLKNAME>\_nPMUINTERERRUPT.

## 14.2.4 Performance monitoring and Secure Debug

If Non-secure event triggering is on, the Secure event enable signals, SPIDEN and SPNIDEN, enable the count and export of both Non-secure and Secure events.

Some events are counted irrespective of the SPNIDEN input and these events are shown as Secure exempt in the PMU event list. For the event lists, see [Performance monitoring](#).

The following table describes the PMU debug signals, their clock relationship, and signal direction.

**Table 14-2: PMU debug signal descriptions, directions, and clock relationships**

Signal	Direction	Description	Clock relationship
<CLKNAME>_NIDEN	Input	Non-invasive debug enable. If HIGH, the signal enables counting and export of PMU events.	Synchronous
<CLKNAME>_SPNIDEN	Input	Secure privileged non-invasive debug enable. When HIGH, this signal enables the counting of both Non-secure and Secure events, provided NIDEN is also HIGH.	Synchronous
<CLKNAME>_DBGEN	Input	Invasive debug enable. If HIGH, enables the counting and export of PMU events.	Synchronous
<CLKNAME>_SPIDEN	Input	Secure privileged invasive debug enable. When HIGH, this signal enables the counting of both Non-secure and Secure events, provided DBGEN is also HIGH.	Synchronous

The counting and export of events that Non-secure events trigger are enabled by the DBGEN and NIDEN inputs: Debug enable = DBGEN | NIDEN.

The full expression for counting Secure and Non-secure events is:

Secure Debug = ((SPIDEN & DBGEN) | SPIDEN) & (DBGEN | NIDEN).

## 14.3 AMNI performance events

NI-710AE AMNIs can generate various performance events. Counting these events provides information about the performance of an AMNI as it operates.

The following table shows the performance events that AMNIs can track. For events marked as Secure only, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure debug is enabled.



SPNIDEN determines whether Secure events are counted or not. However, some events such as Read data do not have the Secure or Non-secure attribute. Therefore, these events are marked as Secure only. You can only count the events if you enable them.

**Table 14-3: AMNI performance events**

Event code bits[5:0]	Event	Secure only
0x00	Read request: any (ARVALID & ARREADY).	N
0x01	Read request: device.	N
0x02	Read request: ReadNoSnoop (RNS).	N
0x03	Read request: ReadOnce (RO).	N
0x04	Cache maintenance requests: CleanShared, CleanInvalid, MakeInvalid, CleanSharedPersist.  <b>Note:</b> CleanSharedPersist is only present in ACE5-Lite.	N
0x05	Read data beat: any (RVALID & RREADY).	Y
0x06	Read data handshake with RLAST set.	Y
0x07	Write request: any (AWVALID & AWREADY).	N
0x08	Write request: device.	N
0x09	Write request: WriteNoSnoop (WNS).	N
0x0A	Write request: WriteLineUnique (WLU).	N
0x0B	Write request: WriteUnique (WU).	N
0x0C	Write request: atomic (AtomicStore, AtomicLoad, AtomicSwap, AtomicCompare).	N
0x0D	Write data beat: any (WVALID & WREADY).	Y
0x0E	Read request stall: ARVALID HIGH, ARREADY LOW.	N
0x0F	Read data stall: RVALID HIGH, RREADY LOW.	Y
0x10	Write request stall: AWVALID HIGH, AWREADY LOW.	N
0x11	Write data stall: WVALID HIGH, WREADY LOW.	Y
0x12	Write response stall: BVALID HIGH, BREADY LOW.	Y
0x13	Write request: cache stash transactions.	N

Event code bits[5:0]	Event	Secure only
0x14	Write channel: Cache Maintenance Operations (CMOs), combined write with CMOs (non-persistent type).  ((AWSNOOP == 0b0110)    (AWSNOOP == 0b1010)    (AWSNOOP == 0b1011)) && (AWCMO == non persist encodings)	N
0x15	Write channel: CMOs, combined write with CMOs (persistent and deep persistent types).  ((AWSNOOP == 0b0110)    (AWSNOOP == 0b1010)    (AWSNOOP == 0b1011)) && (AWCMO == persist and deep persist encodings)	N
0x16	Read requests with nonzero memory tagging operation.	N
0x17	Write requests with nonzero memory tagging operation.	N
0x20	Request stall because of read tracker occupancy.	N
0x21	Request stall because of write tracker occupancy.	N
0x22	Write channel B response stall because of a lack of GT credit.	Y
0x23	Read channel read response stall because of a lack of GT credit.	Y
0x24	Low wire mode arbitration stall on B channel.	Y
0x25	Low wire mode arbitration stall on R channel.	Y

## 14.4 ASNI performance events

NI-710AE ASNIs can generate various performance events. Counting these events provides information about the performance of an ASNI as it operates.

The following table shows the performance events that ASNIs can track. For events marked as Secure only, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure debug is enabled.



SPNIDEN determines whether Secure events are counted or not. However, some events, for example Read data, do not have the Secure or Non-secure attribute. Therefore, these events are marked as Secure exempt. They do not expose any Secure information, only the number of such events.

**Table 14-4: ASNI performance events**

Event code bits[5:0]	Event	Secure only
0x00	Read request: any (ARVALID & ARREADY).	N
0x01	Read request: device ARCACHE[3:1] == 0b000.	N
0x02	Read request: ReadNoSnoop (RNS).	N
0x03	Read request: ReadOnce (RO).	N

Event code bits[5:0]	Event	Secure only
0x04	Cache maintenance requests: CleanShared, CleanInvalid, MakeInvalid, CleanSharedPersist.  <b>Note:</b> CleanSharedPersist is only present in ACE5-Lite.	N
0x05	Read data beat: any (RVALID & RREADY).	Y
0x06	Read data handshake with RLAST set.	Y
0x07	Write request: any (AWVALID & AWREADY).	N
0x08	Write request: device.	N
0x09	Write request: WriteNoSnoop (WNS).	N
0x0A	Write request: WriteLineUnique (WLU).	N
0x0B	Write request: WriteUnique (WU).	N
0x0C	Write request: atomic (AtomicStore, AtomicLoad, AtomicSwap, AtomicCompare).	N
0x0D	Write data beat: any (WVALID & WREADY).	Y
0x0E	Read request stall: ARVALID HIGH, ARREADY LOW.	N
0x0F	Read data stall: RVALID HIGH, RREADY LOW.	Y
0x10	Write request stall: AWVALID HIGH, AWREADY LOW.	N
0x11	Write data stall: WVALID HIGH, WREADY LOW.	Y
0x12	Write response stall: BVALID HIGH, BREADY LOW.	Y
0x13	Write request: cache stash transactions.	N
0x14	Write channel: Cache Maintenance Operations (CMOs), combined write with CMOs (non-persistent type).  ((AWSNOOP == 0b0110)    (AWSNOOP == 0b1010)    (AWSNOOP == 0b1011)) && (AWCMO == non-persistent encodings)	N
0x15	Write channel: CMOs, combined write with CMOs (persistent and deep persistent types).  ((AWSNOOP == 0b0110)    (AWSNOOP == 0b1010)    (AWSNOOP == 0b1011)) && (AWCMO == persistent and deep persistent encodings)	N
0x16	Read requests with nonzero memory tagging operation.	N
0x17	Write requests with nonzero memory tagging operation.	N
0x20	Request stall cycle because of the OT transaction limit.	N
0x21	Request stall cycle because of the Traffic Specification (TSPEC) hard bandwidth regulation limit.	N
0x22	Request stall because of arbitration caused by collision of read and write requests onto shared resources for atomics.	N
0x23	Request stall because of read tracker occupancy.	N
0x24	Request stall because of write tracker occupancy.	N
0x25	AW channel stall because the WDATA FIFO is full.	Y
0x26	AR channel stall because the reorder buffer is full.	N
0x27	AW channel Cyclic Dependency Avoidance Scheme (CDAS) stall.	N
0x28	AR channel CDAS stall.	N
0x29	Atomic RD stall because the read resource is unavailable.	N
0x2A	Write channel write request stall because of a lack of GT credit.	Y



Event code bits[5:0]	Event	Secure only
0x2B	Read channel read request stall because of a lack of GT credit.	Y
0x2C	AW channel stall because of AW or combined Outstanding Transaction (OT) regulation.	N
0x2D	AR channel stall because of AR or combined OT regulation.	N
0x2E	AW channel stall because of AW or combined TSPEC regulation.	N
0x2F	AR channel stall because of AR or combined TSPEC regulation.	N
0x30	Low wire mode arbitration stall on W channel.	Y
0x31	Low wire mode arbitration stall on R channel.	Y

## 14.5 Data bandwidth at ASNI and AMNI

External AXI and ACE-Lite devices connect to the interconnect at ASNIs and AMNIs. You can monitor the data bandwidth through these blocks using specific PMU events.

### 14.5.1 Read and write bandwidth at ASNI and AMNI

NI-710AE provides performance monitoring events to track the number of read and write data beats being transferred. Use these values to calculate the total read and write bandwidth in the interconnect.

The following table shows the events that measure the number of read and write data beats.

**Table 14-5: Read and write data beat tracking events**

Event code bits[5:0]	Description
0x05	Read data beat: Any (RVALID & RREADY)
0x0D	Write data beat: Any (WVALID & WREADY)

Calculate the read and write bandwidth according to the following calculations:

- Read bandwidth = ((number of read data beats × AXIDataBeatSize) / cycles) × frequency
- Write bandwidth = ((number of write data beats × AXIDataBeatSize) / cycles) × frequency



AXIDataBeatSize is the number of bytes for each AXI beat. Usually, this number is the same size as AxSIZE.

## 14.5.2 Delays at ASNI and AMNI because of backpressure

To analyze the delays in ASNI and AMNI, NI-710AE enables you to monitor the source of backpressure.

The following table shows the events that monitor such backpressure:

**Table 14-6: Backpressure monitoring events**

Event code bits[5:0]	Description
0x0E	Read request stall: ARVALID HIGH, ARREADY LOW
0x0F	Read data stall: RVALID HIGH, RREADY LOW
0x10	Write request stall: AWVALID HIGH, AWREADY LOW
0x11	Write data stall: WVALID HIGH, WREADY LOW
0x12	Write response stall: BVALID HIGH, BREADY LOW
0x2A (ASNI) / 0x22 (AMNI)	Write request stall because of a lack of GT credit
0x2B (ASNI) / 0x23 (AMNI)	Read request stall because of a lack of GT credit

## 14.5.3 Delays at ASNI because of structural backpressure

To analyze the delays in ASNI specifically, NI-710AE enables you to monitor the source of backpressure because of structure full or other AXI ordering conditions.

The following table shows events that monitor such backpressure.

**Table 14-7: Structural backpressure monitoring events**

Event code bits[5:0]	Description
0x23	AR stall because of read tracker occupancy
0x24	AW stall because of write tracker occupancy
0x25	W stall because WDATA FIFO is full
0x26	AR stall because of reorder buffer full
0x27	AW CDAS stall
0x28	AR CDAS stall
0x29	Atomic RD stall because of read resource unavailable

## 14.6 AHB performance event mapping

NI-710AE AHB performance events are mapped to AHB memory types.

The AHB PMU events are based on the memory types that are shown in the following table, which is reproduced from the [AMBA® AHB Protocol Specification](#).

**Table 14-8: AHB memory types**

HPROT[6] Shareable	HPROT[5] Allocate	HPROT[4] Lookup	HPROT[3] Modifiable	HPROT[2] Bufferable	Memory type
0	0	0	0	0	Device-nE
0	0	0	0	1	Device-E
0	0	0	1	0	Normal Non-cacheable, Non-shareable
0	0 or 1	1	1	0	Write through, Non-shareable
0	0 or 1	1	1	1	Write back, Non-shareable
1	0	0	1	0	Normal Non-cacheable, Shareable
0	0 or 1	1	1	0	Write through, Shareable
0	0 or 1	1	1	1	Write back, Shareable

## 14.7 HSNi performance events

The NI-710AE HSNi can generate various performance events. Counting these events provides information about the performance of the HSNi as it operates.

The following table shows the performance events that the HSNi can track.

**Table 14-9: HSNi performance events**

Event code bits[4:0]	Event	Secure only
0x00	Read request: any.	N
0x01	Read request: device (Device-nE and Device-E).	N
0x02	Read request: 1. Normal Non-cacheable, Non-shareable 2. Write-Through, Non-shareable 3. Write-Back, Non-shareable	N
0x03	Read request: Normal, Non-cacheable, Shareable.	N
0x04	Read request: 1. Write-Through, Shareable 2. Write-Back, Shareable	N/A
0x05	Read data beat: any.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x06	N/A.	Y
0x07	Write request: any.	N
0x08	Write request: device (Device-nE and Device-E).	N
0x09	Write request: Normal, Non-cacheable, Non-shareable.	N

Event code bits[4:0]	Event	Secure only
0x0A	Write request: Write-Through or Write-Back, Shareable, Non-shareable.	N
0x0B	Write request: Normal, Non-cacheable, Shareable.	N
0x0C	Write request: 1. Write-Through Shareable 2. Write-Back, Shareable	N
0x0D	Write data beat: any.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x0E	Read address phase stall. Not implemented in the HSNI, tied to 0.	N/A
0x0F	Read data phase stall. Prior read address phase, HREADY LOW.	Y
0x10	Write address phase stall. Not implemented in the HSNI, tied to 0.	N/A
0x11	Write data phase stall. Prior write address phase, HREADY LOW.	Y
0x12	Reserved.	N/A
0x13	N/A.	N
0x20	Request stall cycle because of OT transaction limit.	N
0x21	Request stall cycle because of Hard BW (TSPEC) regulation limit.	N
0x22	Read request stall because of early write responses: Early write response needs read hazarding until all the write responses have returned on GT. This condition leads to stalling of read request.	N
0x23	N/A.	N
0x24	Request stall because of nonzero outstanding write counter.	N
0x25	W stall because WDATA FIFO is full. HSNI uses the WDATA FIFO to store and forward data for improving GT efficiency.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x26	N/A.	N
0x27	N/A.	N
0x28	N/A.	N
0x29	N/A.	N

Event code bits[4:0]	Event	Secure only
0x2A	Write request stall because of a lack of GT credit.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x2B	Read request stall because of a lack of GT credit.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x2C	N/A.	N
0x2D	N/A.	N
0x2E	N/A.	N
0x2F	N/A.	N
0x30	N/A.	N
0x31	N/A.	N

## 14.8 HMNI performance events

The NI-710AE HMNI can generate various performance events. Counting these events provides information about the performance of the HMNI as it operates.

The following table shows the performance events that the HMNI can track.

**Table 14-10: HMNI performance events**

Event code bits[4:0]	Event	Secure only
0x00	Read request: any.	N
0x01	Read request: device (Device-nE and Device-E).	N
0x02	Read request: <ol style="list-style-type: none"> <li>1. Normal Non-cacheable, Non-shareable</li> <li>2. Write-Through, Non-shareable</li> <li>3. Write-back, Non-shareable</li> </ol>	N
0x03	Read request: Normal, Non-cacheable, Shareable.	N

Event code bits[4:0]	Event	Secure only
0x04	Read request: 1. Write-Through, Shareable 2. Write-back, Shareable	
0x05	Read data beat: any.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x06	N/A.	N
0x07	Write request: any.	N
0x08	Write request: device (Device-nE and Device-E).	N
0x09	Write request: Normal, Non-cacheable, Non-shareable.	N
0x0A	Write request: Write-Through or Write-Back, Non-shareable.	N
0x0B	Write request: Normal, Non-cacheable, Shareable.	N
0x0C	Write request: 1. Write-Through, Shareable 2. Write-back, Shareable	N
0x0D	Write data beat: any.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x0E	Read address phase stall. HTRANS[1] HIGH, HWRITE LOW, HREADY LOW.	N
0x0F	Read data phase stall. Prior read address phase, HREADY LOW.	Y
0x10	Write address phase stall. HTRANS[1] HIGH, HWRITE HIGH, HREADY LOW.	N
0x11	Write data phase stall. Prior write address phase, HREADY LOW.	Y
0x12	Reserved.	N/A
0x13	N/A.	N
0x20	N/A.	N
0x21	N/A.	N

Event code bits[4:0]	Event	Secure only
0x22	Write response stall because of a lack of GT credit.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x23	Read response stall because of a lack of GT credit.	Y  For this event, the request security attribute (Secure or Non-secure) is not available at the point the event is captured. Therefore, to ensure Secure information is not exposed, the event is captured only when Secure Debug is enabled.
0x24	N/A.	N
0x25	N/A.	N

## 14.9 Data bandwidth at HSNi and HMNI

Data bandwidth performance can be monitored at HSNIs and HMNI.

### 14.9.1 Read and write bandwidth at HSNi and HMNI

NI-710AE provides performance monitoring events to track the number of read and write data beats being transferred. These values can be used to calculate the total read and write bandwidth in the interconnect.

The following table shows the events that measure the number of read and write data beats.

**Table 14-11: Read and write data beat tracking events**

Event code bits[5:0]	Description
0x05	Read data beat: any
0x0D	Write data beat: any

Calculate the read and write bandwidth according to the following calculations:

- Read bandwidth = ((number of read data beats × AHBDDataBeatSize) / cycles) × frequency
- Write bandwidth = ((number of write data beats × AHBDDataBeatSize) / cycles) × frequency



Note

AHBDDataBeatSize is the number of bytes for each AHB beat. HSIZE determines this number which must be less than or equal to the size of the AHB bus.

## 14.9.2 Delays at HSNl and HMNI because of backpressure

To analyze the delays in HSNl and HMNI, NI-710AE enables you to monitor the source of backpressure.

The following table shows the events that monitor such backpressure.

**Table 14-12: Backpressure monitoring events**

Event code bits[5:0]	Description
0x0E	Read request stall: HREADY LOW from the completer
0x0F	N/A
0x10	Write request stall: HREADY LOW
0x11	Write data stall: HREADY LOW
0x12	N/A

## 14.9.3 Delays at HSNl because of structural backpressure

To analyze the delays in HSNl specifically, NI-710AE enables you to monitor the source of backpressure because of structure full or other AXI ordering conditions.

The following table shows events that monitor such backpressure.

**Table 14-13: Structural backpressure monitoring events**

Event code bits[5:0]	Description
0x24	Request stall because of nonzero outstanding write counter.
0x25	W stall because WDATA FIFO is full. HSNl uses the WDATA FIFO to store and forward data for improving GT efficiency.

## 14.10 PMNI performance events

The NI-710AE PMNI can generate various performance events. Counting these events provides information about the performance of the PMNI as it operates.

The following table shows the performance events that the PMNI can track.

**Table 14-14: PMNI performance events**

Event code bits[4:0]	Event	Secure only
0x00	Read request: any (PENABLE & PREADY) and ~PWRITE	N
0x01	Read request: device	N
0x02	Read request: Non-shareable (Domain == Non-shareable or system shareable)	N
0x03	N/A	N
0x04	N/A	N



Event code bits[4:0]	Event	Secure only
0x05	Read data beat: any PRDATA	Y
0x06	N/A	Y
0x07	Write request: any (PENABLE & PREADY) and PWRITE	N
0x08	Write request: device	N
0x09	Write request: Non-shareable (Domain == Non-shareable or system shareable)	N
0x0A	N/A	N
0x0B	N/A	N
0x0C	N/A	N
0x0D	Write data beat: any (PWRITE & PREADY) and write	N
0x0E	Read request stall: PREADY LOW for read, when PENABLE is HIGH	N
0x0F	Read data stall: PREADY LOW for Read, when PENABLE is HIGH	N
0x10	Write request stall: PREADY LOW for write, when PENABLE is HIGH	N
0x11	Write data stall: PREADY LOW for write, when PENABLE is HIGH	N
0x12	N/A	N
0x13	N/A	N
0x20	N/A	N
0x21	N/A	N
0x22	Write response stall because of a lack of GT credit	N
0x23	Read response stall because of a lack of GT credit	N
0x24	N/A	N
0x25	N/A	N

## 15. Error handling and interrupts

The NI-710AE endpoints have error handling and interrupt functionality. This functionality is related to the IDM functionality and other specific non-IDM conditions.

For more information about the IDM feature, see [Interconnect Device Management](#).

The error logging and interrupt registers are distributed in the NI-710AE ASNI, AMNI, HSNI, HMNI, and PMNI endpoints. These registers communicate with central interrupt handling logic in each power domain.

All NI-710AE errors are Uncorrected Errors (UEs).

### 15.1 IDM error logging interrupts and status flags

The IDM error logging functionality records details of any transaction that generates an error so that software can examine the transaction. When multiple errors are generated at the same time, the system uses error storage rules to determine the priority for logging.

The following rules are used to determine the error logging priority for simultaneous errors:

- If read and write transactions generate an error simultaneously, the write transaction has higher priority for error logging.
- If a timeout is detected in the same cycle as read or write bus error, the transaction that has timed out has higher priority for error logging.

When the error logging logic receives an error, an interrupt is raised. The error logging logic can raise separate interrupts for the following conditions:

- Bus errors (SLVERR or DECERR)
- Timeout errors
- Endpoint receives incoming requests while in soft reset or isolation state

A software-readable status flag indicates that the logic is processing an error and the type of error being processed. If the error logging logic receives an error while processing another error, an overflow flag is used to record that multiple errors have occurred. This overflow flag is used when simultaneous read, write, and timeout errors occur.

To process an error, software must access the address and associated characteristics of the transaction that caused the error. After processing an error, software can clear the following items separately:

- The interrupt raised by the error logging logic on receiving the error
- The error status flag. Clearing the flag indicates that the error logging logic can store another error-causing transaction for processing.

When the IDM block detects a timeout for a device, software can perform an IDM soft reset or isolate the external device. However, bus errors or timeout errors and their corresponding interrupts can still occur even after the external device enters the active soft reset state. These errors can happen under certain circumstances where soft reset was entered in the middle of a pending transaction, and the error status was cleared. In such cases, new timeout errors can be reported. However, since software is aware that any further timeout errors on that interface are not meaningful, software can choose to:

- Disable all error detection using the relevant `idm_errctlr` register
- Disable the timeout error detection using the relevant `idm_timeout_control` register during the period when soft reset is in active state

For more information on the `idm_errctlr` and `idm_timeout_control` register bit assignments, see:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

To exit from soft reset, you must clear the relevant Secure `idm_errstatus` or Non-secure `idm_errstatus_ns` register. See the preceding register summaries to view the relevant registers.

## 15.2 IDM error logging registers

The IDM error logging functionality uses specific registers to store details of the transaction that caused the error. When you configure a network interface to include IDM functionality, these registers are added to the interface register block.

NI-710AE uses the following registers for IDM error logging:

### **`idm_errstatus`**

Indicates the following information about IDM errors:

- Whether an error has occurred
- The type of error
- The overflow flag
- The validity of other error attribute registers, such as `idm_errmisc0` and `idm_errmisc1`

### **`idm_erradr_lsb` and `idm_erradr_msb`**

Stores the address of the transaction that caused the error.

### **`idm_errmisc0` and `idm_errmisc1`**

Stores other attributes of the transaction that caused the error, including AXI ID, node ID, burst length, and size.

For more information about these registers, see:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

## 15.3 IDM error processing sequence

When the endpoint logs an IDM error, the system uses a specific sequence of register writes to process the error.

The system uses the following sequence to process IDM errors:

1. Log the error information in the applicable `idm_errstatus`, `idm_erradr_lsb/idm_erradr_msb`, and `idm_errmisc0/idm_errmisc1` registers.
2. Set the V and UE fields of the associated `idm_errstatus` register.
3. Set the UI field of the `idm_errctlr` register to mask signaling of the error to the RAS control block.
4. If there are multiple UEs, set the OF field of the `idm_errstatus` register.

For more information about these registers, see:

- [ASNI register summary](#)
- [AMNI register summary](#)
- [HSNI register summary](#)
- [HMNI register summary](#)
- [PMNI register summary](#)
- [Power domain register summary](#)

## 15.4 Interrupts

AMNIs, HSNIs, and HMNIs implement interrupt signals to indicate specific conditions.

When enabled on an interface, the Interconnect Device Management (IDM) block can also generate interrupts to signal errors. For more information about IDM interrupts, see [IDM error logging interrupts and status flags](#).

The following table shows the non-IDM interrupt conditions for AMNIs, HSNIs, and HMNIs.

**Table 15-1: Non-IDM interrupt conditions for network interfaces**

Endpoint	Interrupt condition
AMNI	<ul style="list-style-type: none"> <li>Non-modifiable transaction split into multiple individual burst transactions</li> <li>Unsupported ACE5-LiteACP request</li> </ul>
HSNI	<ul style="list-style-type: none"> <li>Non-modifiable transaction split into multiple individual burst transactions</li> <li>Imprecise errors detected on actual write response received for a request when early write responses have already been sent for that request</li> </ul>
HMNI	Non-modifiable transaction burst split into multiple transactions

Each network interface generates interrupts using a set of registers. For more information about these registers, see [Programmers model](#).

The AXI specification defines ACE5-LiteACP as a subset of ACE5-Lite with specific constraints. NI-710AE supports the following combinations of ACE5-Lite and ACE5-LiteACP.

**Table 15-2: Supported ACE5-Lite and ACE5-LiteACP combinations**

Requester upstream of ASNI	Completer downstream of AMNI	Interoperability
ACE5-Lite	ACE5-LiteACP	<p>Can connect directly if the requester upstream of an ASNI uses the ACE5-LiteACP subset of transactions.</p> <p>If the AMNI is configured for ACE5-LiteACP, the AMNI expects to receive the subset of transactions that is defined in the ACE5-LiteACP specification. The AMNI checks whether the transaction properties satisfy the ACE5-LiteACP constraints. If the constraints are not met, then the AMNI raises an interrupt. For example, an interrupt is raised if the AMNI receives a WRAP burst or if the original AxSIZE of the transaction was 256 bits. ACE5-LiteACP only permits INCRs with an AxSIZE of 128 bits.</p>
ACE5-LiteACP	ACE5-Lite	If the ASNI only supports ACE5-Lite, then this interface is fully compatible with the ACE5-LiteACP subset of transactions. Unused inputs to the ASNI can be tied off.

## 15.5 Interrupt generation

To minimize the number of top-level interrupts in large interconnect designs, NI-710AE implements a hierarchical interrupt structure. Interrupts that are generated by network interfaces are passed to an internal status unit for each power domain.

The power domain status units are responsible for:

- Asserting external interrupts
- Storing the first interface to raise an interrupt of a specific type
- Recording the number of interrupts that are raised internally

The NI-710AE interrupt hierarchy consists of two levels.

### Level 1

AMNIs, HSNIs, and HMNIs have interrupt status registers for every type of error that is reported. For more information, see [Interrupts](#).

When Interconnect Device Management (IDM) is enabled on an interface, extra interrupt registers are added to communicate bus errors and timeout errors. There are also interrupt registers for incoming requests to devices in the IDM soft reset and isolation states. For more information, see [IDM error logging interrupts and status flags](#).

An internal interrupt is asserted whenever any bits in the relevant register are set to 1. The internal interrupt targets the central interrupt handling block in each power domain.

Every interface has an interrupt mask register to mask interrupt generation for specific types of events. If IDM is enabled on the interface, a separate interrupt mask register masks interrupt generation for IDM errors.

## Level 2

The collated control and status registers for each interrupt type contain the number of interfaces that have asserted an interrupt type. These registers can also mask further interrupts. Software can use the information in these registers to determine if there are multiple internal interrupts to clear.



There is only one register to record the Node ID of the first interrupt that the system receives. This register updates with further asserted interrupts when the indicated interface has been serviced. To clear an interrupt, software must act on the associated registers that are located within the address region of the interface.

If multiple interfaces raise an interrupt at the same time, the following order of priority is used to determine the first interrupt to report:

1. Highest priority: xMNI, completer device.
2. Lowest priority: xSNI, requester device. No conflicting xMNI interrupt.

Where there are multiple endpoints with the same priority, the interface with the lowest internal Node ID takes precedence. The programmers view provides the Node ID.

## 15.6 Error interrupt handler flow

A specific sequence of events must occur for software to process errors.

When Interconnect Device Management (IDM) is enabled, this process also applies to software processing of IDM errors.

The following sequence of events describes the process for determining the error source and type of interrupt:

1. A network interface generates an interrupt.

There is a separate wire for each interrupt type. The wires are used to communicate the internal interrupt to the central interrupt handling block of the power domain. Across interfaces, the central interrupt handling block groups individual internal interrupt signals in order of interface Node ID.

2. The central interrupt handling block uses an arbitration mechanism to record the Node ID of the interface for which the external interrupt is raised. For more information about the arbitration mechanism, see [Interrupt generation](#).

The interrupt handler reads the register and uses the Node ID value to read the corresponding interrupt registers in the interface. For IDM interrupts, the interrupt handler also accesses the IDM error logging registers.

3. The interrupt status registers for the interface indicate the type of error.

When the interrupt is caused by a non-IDM error at an AMNI, HSNI, and HMNI, the `interrupt_status[_ns]` and `interrupt_mask[_ns]` registers contain the error information.

For IDM errors, the error type is logged in the `idm_errstatus` register. IDM-enabled interfaces can also log further attributes of the request that generated the IDM interrupt in the `idm_erradr_lsb`, `idm_erradr_msb`, `idm_errmisc0`, and `idm_errmisc1` registers.

For more information about the interrupt status, mask, and logging registers, see [Programmers model](#).

4. When software has finished processing an error, it can separately clear any interrupt that was asserted in relation to the error. Software can clear the interrupt by clearing the interrupt status register.

For IDM interrupts, software can also clear the error status flag by clearing the `status_valid` field of the `idm_errstatus` register. A subsequent error-causing transaction can then be stored and processed.

## 15.7 Error handling and interrupt security

NI-710AE separates interrupt pins, interrupt registers, and error logging registers into Secure and Non-secure variants.

When a Secure request generates an error, the error properties of the request are logged in the Secure error logging and interrupt registers. The Secure interrupt pin is asserted.

When a Non-secure request generates an error, the error conditions are logged in the Non-secure error logging and interrupt registers. The Non-secure interrupt pin is asserted.

This separation permits Non-secure software to access the Non-secure registers, while preventing access to the Secure registers.

## 15.8 Error responses

AMNIs, HMNIs, PMNIs, and the Configuration Network Interface (CFGNI) send error responses when an unsupported transaction type is received.

### AMNI

#### Requests on the read channel

- When an AMNI with an AXI interface downstream receives an ACE-Lite transaction with AxDOMAIN set to 0b01 or 0b10, the transaction is terminated at the AMNI with an SLVERR response:
  - If there is a requirement to send shareable requests downstream, the interface type must be set to ACE-Lite.
  - If the downstream device is an AXI completer, the AxSNOOP or AxDOMAIN signals can be left unconnected. Leaving these signals unconnected effectively downgrades the requests. For example, ReadOnce is downgraded to ReadNoSnoop.
- Cache Maintenance Operation (CMO) transactions on the read channel:
  - If an AMNI has an AXI interface downstream, incoming CMO requests are terminated at the AMNI with an OK response by default. However, you can program a configuration control register to change the OK response to an SLVERR response. For more information, see [AMNI cmoovrd register](#).
  - If the CMO\_ON\_READ and CMO\_ON\_WRITE properties are set to FALSE on an AMNI with an ACE-Lite interface downstream, this configuration indicates that there is no downstream cache. In this scenario, the transaction is terminated at the AMNI with an OK response. Alternatively, you can program a configuration control register to change the OK response to an SLVERR response. For more information, see [AMNI cmoovrd register](#).
  - If the CMO\_ON\_WRITE property is set to TRUE on an AMNI with an ACE-Lite interface downstream, this configuration indicates a possible downstream cache. In this scenario, the transaction is terminated at the AMNI with an SLVERR response.

#### Requests on the write channel

- When an AMNI with an AXI interface downstream receives an ACE-Lite transaction with AxDOMAIN set to 0b01 or 0b10, the transaction is terminated at the AMNI with an SLVERR response:
  - If there is a requirement to send shareable requests downstream, the interface type must be set to ACE-Lite.
  - If the downstream device is an AXI completer, the AxSNOOP or AxDOMAIN signals can be left unconnected. Leaving these signals unconnected effectively downgrades the requests. For example, WriteUnique is downgraded to WriteNoSnoop.
- Incoming atomic transactions are terminated at the AMNI with an error response if either:
  - The Atomic\_Transactions property is set to FALSE.
  - The Atomic\_Transactions property is set to TRUE, but the incoming request has AxDOMAIN set to 0b01 or 0b10 and the downstream interface is an AXI interface.



- Incoming prefetch transactions to an AMNI with an AXI interface or an ACE-Lite interface always return an OK response.
- Incoming cache stash transactions that target an AMNI which does not support cache stashing are converted to the transaction types shown in the following table. However this conversion depends on AxDOMAIN.

**Table 15-3: AMNI cache stash transaction handling**

Cache stash transaction	Domain	ACE-Lite AMNI with Cache_Stash_Transactions property set to FALSE	AXI4 AMNI
WriteUniquePtlStash	Non-shareable or System	Convert to WriteNoSnoop	Do not propagate and give an immediate SLVERR response
WriteUniquePtlStash	Inner or Outer Shareable	Convert to WriteUnique (WriteUniquePtl)	Do not propagate and give an immediate SLVERR response
WriteUniqueFullStash	Non-shareable or System	Convert to WriteNoSnoop	Do not propagate and give an immediate SLVERR response
WriteUniqueFullStash	Inner or Outer Shareable	Convert to WriteUnique (WriteUniqueFull)	Do not propagate and give an immediate SLVERR response
StashOnceShared	Any	Do not propagate and give immediate OK response	Do not propagate and give an OK response
StashOnceUnique	Any	Do not propagate and give immediate OK response	Do not propagate and give an OK response

#### CMO transactions on the write channel

- If an AMNI has an AXI interface downstream, incoming CMO requests are terminated at the AMNI with an OK response by default. However, you can program a configuration control register to change the OK response to an SLVERR response. For more information, see [AMNI cmoovrd register](#).
- If the CMO\_ON\_READ and CMO\_ON\_WRITE properties are set to FALSE on an AMNI with an ACE-Lite interface downstream, this configuration indicates that there is no downstream cache. In this scenario, the transaction is terminated at the AMNI with an OK response. Alternatively, you can program a configuration control register to change the OK response to an SLVERR response. For more information, see [AMNI cmoovrd register](#).
- If the CMO\_ON\_WRITE property is set to TRUE on an AMNI with an ACE-Lite interface downstream, this configuration indicates a possible downstream cache. In this scenario, the transaction is terminated at the AMNI with an SLVERR response.

#### Write+CMO transactions on the write channel

- If an AMNI has an AXI interface downstream, incoming Write+CMO requests are terminated at the AMNI with an SLVERR response.
- If the WRITE\_PLUS\_CMO, CMO\_ON\_READ, and CMO\_ON\_WRITE properties are set to FALSE on an AMNI with an ACE-Lite interface downstream, this configuration indicates that there is no downstream cache. In this scenario, the transaction is downgraded and only the write part of the transaction is issued downstream. The response indication for the downstream write, which comes from one of two sources, determines the response error indication. The two sources are the actual response for the write from downstream, and the response value that is indicated by the value of the configuration control register.

The highest priority between these sources is used for the response indication. For example, if the downstream response indicates SLVERR and the configuration control register value indicates an OK response, the final response is an SLVERR. Alternatively, if the downstream response is an OK response but the configuration control register value indicates an SLVERR response, then the final response is an SLVERR. For more information about the responses, see [AMNI cmoovrd register](#).

- If an AMNI has an ACE-Lite interface downstream, the WRITE\_PLUS\_CMO property is set to FALSE. However, if either or both of the CMO\_ON\_READ property or the CMO\_ON\_WRITE properties are set to TRUE, this configuration indicates a possible downstream cache. In this scenario, the transaction is terminated at the AMNI with an SLVERR response.

## HMNI

The following request types are terminated at the HMNI and an SLVERR response is sent based on the configuration of the interface. An HMNI with an AHB-Lite interface, or an AHB5 interface without extended memory type support, responds with an SLVERR to shareable requests with DOMAIN set to 0b01 or 0b10.

**Table 15-4: HMNI error responses**

Request	AHB5 HMNI with Extended_Memory_Types property set to TRUE	AHB-Lite HMNI or AHB5 HMNI with Extended_Memory_Types property set to FALSE
WriteNoSnoop	Write, Non-shareable	Write, Non-shareable
WriteUnique, WriteLineUnique	Write, shareable	SLVERR
WriteUniqueStash, WriteLineUniqueStash	Write, shareable	SLVERR
WriteCMO, WriteLinePlusCMO, WritePlusCMO	SLVERR	SLVERR
WritePrefetch	OK	OK
StashOnceShared, StashOnceUnique	OK	OK
ReadNoSnoop	Read, Non-shareable	Read, Non-shareable
ReadOnce	Read, shareable	SLVERR
DeAllocating transactions (ReadOnceCleanInvalid, ReadOnceMakeInvalid)	Read, shareable	SLVERR
CMO (CleanShared, CleanInvalid, MakeInvalid, CleanSharedPersist)	SLVERR	SLVERR
Atomic transactions (AtomicSwap, AtomicStore, AtomicCompare, AtomicLoad)	SLVERR	SLVERR

## PMNI

PMNIs only support the ReadNoSnoop and WriteNoSnoop request types. All other requests to PMNIs are terminated at the interface and an SLVERR response is sent. Unsupported requests include WriteUnique, WriteLineUnique, ReadOnce, cache maintenance requests, cache stashing transactions, deallocating transactions (ReadOnceCleanInvalid and ReadOnceMakeInvalid), and atomics.

## Internal Configuration Network Interface

All requests that map to the configuration address space are sent to the internal CFGNI. This interface only supports ReadNoSnoop and WriteNoSnoop request types. All other requests to the CFGNI are terminated at the CFGNI and an SLVERR response is sent. Unsupported requests include WriteUnique, WriteLineUnique, ReadOnce, cache maintenance requests, cache stash transactions, deallocating transactions (ReadOnceCleanInvalid and ReadOnceMakeInvalid), and atomics.

## 16. Programmers model

The NI-710AE interconnect consists of various components, such as ASNIs, AMNIs, HSNIs, HMNIs, and PMNIs. You can access these components through memory-mapped registers for configuration, topology, and status information.

The memory-mapped registers are organized in a series of 4KB regions. They are accessed through AXI or ACE-Lite read and write commands.

The base address of the configuration registers is not fixed and can be different for any particular system implementation. The offset of each register from the configuration base address is fixed.

When accessing the configuration registers, do not attempt to access reserved or unused address locations. Attempting to access these locations can result in **UNPREDICTABLE** behavior. Unless otherwise stated in the accompanying text:

- Do not modify **UNDEFINED** register bits.
- Ignore **UNDEFINED** register bits on reads.
- All register bits are reset to 0 by a system or Cold reset.

Each register has an associated access type. The NI-710AE registers use the following access type abbreviations:

### **RW**

Read and write

### **RO**

Read-only

### **WO**

Write-only

### **RAZ**

Read-As-Zero

### **WI**

Write ignored

Some bit positions in registers are described as reserved. These bit positions have the following access types:

- **RAZ**/**WI** in an **RW** register
- **RAZ** in an **RO** register
- **WI** in a **WO** register

When accessing the configuration registers, no error is returned on RRESP and BRESP responses.

The NI-710AE registers are accessed using the AXI and ACE-Lite completer interfaces that are configured through Socrates.

The programmers model contains regions for control, upstream NIs, downstream NIs, and PMUs. Accesses to unmapped or reserved registers are **WI** or **RAZ**. Non-secure accesses to Secure registers are **WI** or **RAZ**.

NI-710AE contains several control registers that enable software to modify the behavior of the product. Usually, programming the control registers immediately impacts the execution of transactions that flow through NI-710AE.

Before programming a control register in a specific unit instance, such as a particular ASNI instance, Arm recommends bringing the unit to a quiesced state. The BRESP response for the configuration write to the register confirms that the register write is complete. After a write to the register occurs and the BRESP is received, further transactions can be issued. Following this recommendation provides a clear boundary, after which, further transactions to that instance use the updated control register value. For more information, see [Requirements of configuration register reads and writes](#).

NI-710AE provides a mechanism for software to discover the configuration of the product. For more information, see [Discovery](#).

## 16.1 Requirements of configuration register reads and writes

Reads and writes to the NI-710AE configuration registers must meet certain requirements, otherwise the interconnect returns an error response. There are also security considerations when reading from and writing to the NI-710AE configuration registers.

Reads and writes to the NI-710AE configuration registers must meet the following requirements:

- The request must be of Device type
- The request must be ReadNoSnoop or WriteNoSnoop
- The request must not be an exclusive access
- The AxDOMAIN must be system shareable
- The burst type must be INCR
- The size must be 4 bytes
- The address must be 32-bit word-aligned
- All write strobes for the 4 bytes must be set

If an incoming request does not obey these constraints, NI-710AE returns the request with an SLVERR. Reads are handled as **RAZ** and writes as **WI**. However, the transaction completes in a protocol-compliant manner with SLVERR on the RRESP or BRESP as appropriate.

Secure registers are only accessible through a Secure request, depending on the value of the Secure access register in the unit. Non-secure registers are accessible through either a Secure or a Non-secure request.

Security mismatches are not reflected as an SLVERR, although other conditions determine the error response indicated. For example, if there is a security mismatch together with an unsupported request opcode, then an SLVERR is indicated due to the unsupported request opcode. However, if the only cause is the security mismatch then an OK response is returned.

## 16.2 Discovery

Discovery is an algorithm that software can use to determine the structure of the NI-710AE configuration as the system boots. In other words, software can determine the structure of the NI-710AE domains, components, and subfeatures without previous knowledge of the configuration.

The programming network in NI-710AE connects to all the blocks of configuration registers, also known as configuration nodes, across the interconnect. The discovery process uses these configuration nodes to determine the NI-710AE configuration. For more information about configuration nodes, see [Configuration nodes](#).

Different configuration nodes are distributed across the interconnect, in their associated interconnect component. Therefore, to configure all the interconnect components, software must locate each configuration node. The discovery mechanism generates a structure of all locations of the configuration nodes for each domain, component, and subfeature in the configuration. This structure is called a discovery tree.

To build the discovery tree, the discovery process starts at the base address of the configuration space, PERIPHBASE. Then discovery uses pointer values to determine the number and type of each interconnect domain or component, their attributes, and the location of the configuration registers. Software can use this information to access these registers for configuration purposes. For more information, see [Discovery flow](#).

### 16.2.1 Configuration nodes

Each NI-710AE domain, component, and subfeature in the configuration has a block of programmable configuration registers in the configuration space. These blocks are called configuration nodes and they contain information and settings for the domain, component, or subfeature with which they are associated, and pointers to child configuration nodes.

NI-710AE has the following types of configuration nodes:

#### **Global configuration node**

The first level, or root, of the discovery tree. The global configuration node provides the number of voltage domains and pointers to voltage domain registers. It also contains global interconnect registers. The global configuration node is at the base of the NI-710AE configuration space, and this location is also known as PERIPHBASE. For more information, see [Global configuration register region](#).

#### **Voltage domain**

Indicates the number of power domains in a voltage domain and provides pointers to each power domain that it contains. Each voltage domain configuration node also contains voltage

domain-specific control registers. For more information, see [Voltage domain configuration register region](#).

### Power domain

Indicates the number of clock domains in a power domain and provides pointers to each clock domain that it contains. Each power domain configuration node also contains power domain-specific control registers. For more information, see [Power domain configuration register region](#).

### Clock domain

Indicates the number of components in a clock domain and provides pointers to each component that it contains. Each clock domain configuration node also contains clock domain-specific control registers. For more information, see [Clock domain configuration register region](#).

### Component

Indicates the type of component node, the number of subfeatures and pointers, and the component-specific registers. For more information, see [Component configuration register region](#).

### Subfeature

Contains the subfeature-specific registers. For more information, see [Subfeature configuration register region](#).

For more information about the structure and organization of configuration nodes, see the following sections:

- [Configuration node hierarchy](#)
- [Structure of configuration nodes](#)

#### 16.2.1.1 Configuration node hierarchy

NI-710AE organizes the configuration nodes for all the units of an interconnect configuration into a hierarchy. Each level of the hierarchy contains the configuration nodes for a specific type of logical unit.

NI-710AE is structured so that a logical unit, such as a domain, component, or subfeature, can be contained by another logical unit. For example:

- A power domain contains one or more clock domains
- A clock domain contains one or more components
- A component might contain a subfeature

The configuration node for a unit that is contained by another unit is referred to as a child node. For example, each clock domain is contained by a power domain. Therefore, a clock domain configuration node is a child node of a power domain configuration node. If a configuration node does not contain any child nodes, it is considered to be a leaf node.

Every configuration node that has child nodes contains pointers to all of its child nodes. These pointers indicate the base address of the child node. For example, consider a power domain that

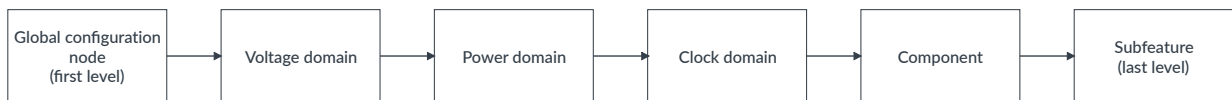
contains multiple clock domains. The configuration node for this power domain contains pointers that indicate the base address of the configuration node for each clock domain it contains.

The size of the pointer region for each type of configuration node is fixed, regardless of the number of configured child nodes. For example, NI-710AE supports up to 32 clock domains in total. If necessary, all of those clock domains can be contained by a single power domain. Therefore, the pointer region for all power domain configuration nodes is large enough to contain up to 32 clock domain pointer registers. If a power domain has one clock domain child node in your configuration, the pointer region is still large enough to contain 32 pointer registers. However, in this case, only one pointer register is present in the pointer region.

Each configuration node also contains local configuration registers for the unit with which it is associated. For example, the configuration node for a power domain contains power domain configuration registers, alongside registers that indicate the number of child nodes and their pointers. The clock domain-specific configuration registers for each child of the power domain are in the individual child clock domain configuration nodes. Leaf nodes only contain registers that are specific to that unit.

The following figure shows the hierarchy of NI-710AE configuration nodes in the discovery tree.

**Figure 16-1: Discovery tree structure**



### 16.2.1.2 Structure of configuration nodes

The NI-710AE configuration nodes contain identifying information about the unit that they are associated with. You can configure the size of the configuration nodes in NI-710AE.

Each configuration node contains the following identifying information about the associated logical domain or unit:

#### Node ID

NI-710AE assigns an identifier to all the xSNI and xMNI nodes.

The node ID spaces of the xSNI and xMNI nodes can overlap. However, two nodes of the same type cannot have the same node ID. Therefore, discovery software can identify nodes with the same node ID through their node type values.

#### Node type

Most of the configuration nodes have their own distinct node type value, as indicated by the node\_type register. NI-710AE has the following node type values:

##### 0x0000

NI-710AE global configuration node



<b>0x0001</b>	Voltage domain
<b>0x0002</b>	Power domain
<b>0x0003</b>	Clock domain
<b>0x0004</b>	ASNI
<b>0x0005</b>	AMNI
<b>0x0006</b>	Performance Monitoring Unit (PMU)
<b>0x0007</b>	HSNI
<b>0x0008</b>	HMNI
<b>0x0009</b>	PMNI
<b>0x0040</b>	Clock controller
<b>0x0041</b>	Power controller
<b>0x0060</b>	Configuration Network Interface (CFGNI)
<b>0x0061</b>	Fault Management Unit (FMU)

All configuration nodes with child nodes also contain a pointer region. This region contains pointer values to the base address for each child node associated with that configuration node.

The discovery process also permits software to capture more information about the node configuration. For example, it can capture the interface ID of the external interface or interfaces of the block in question.

Furthermore, each configuration node contains all the configuration, information, and status registers for the associated unit. Each configuration node is associated with a NI-710AE instance.

The NI-710AE configuration nodes are all 4KB large. You can determine the base address of each configuration node at compile time or at initialization through the discovery flow. For more information, see [Discovery flow](#).

The structure and contents the configuration nodes vary according to the node type. For the specific structure of the different configuration node types, see the following sections:

- [Global configuration register region](#)
- [Voltage domain configuration register region](#)
- [Power domain configuration register region](#)
- [Clock domain configuration register region](#)
- [Component configuration register region](#)
- [Subfeature configuration register region](#)

#### 16.2.1.2.1 Global configuration register region

The first 4KB block above PERIPHBASE contains global information and configuration registers for NI-710AE. It also contains the first level of discovery information for components in the system.

The following table shows the structure of this lowest register block. For more information about these register descriptions, see the [Programmers model](#).

**Table 16-1: NI-710AE ID registers**

Offset	Contents
0x0	NI-710AE global node type register
<b>NI-710AE voltage domain configuration mapping</b>	
0x4	Number of voltage domain regions present
0x8	Voltage domain 0 base address, offset from PERIPHBASE
0xC	Voltage domain 1 base address, offset from PERIPHBASE
0x10	Voltage domain 2 base address, offset from PERIPHBASE
...	...
...	Voltage domain[N], where N is the total number of voltage domains in NI-710AE
<b>NI-710AE global configuration registers</b>	
0x00FD0	Peripheral ID4
0x00FD4	Peripheral ID5
...	Extra global configuration registers

Each voltage domain base address register in the table contains the offset from PERIPHBASE of a voltage domain configuration node. These registers also contain:

- Information about a single voltage domain
- Discovery information for components that are associated with that voltage domain

### 16.2.1.2.2 Voltage domain configuration register region

Each voltage domain has a 4KB configuration register region that contains information about the voltage domain. This region also contains offset addresses for all associated power domain configuration nodes.

The following table shows the structure of the voltage domain configuration register region.

**Table 16-2: Contents of the voltage domain configuration register region**

Offset	Contents
0x0	Voltage domain ID register
<b>NI-710AE power domain configuration mapping</b>	
0x4	Number of power domain regions present
0x8	Power domain 0, within voltage domain, base address, offset from PERIPHBASE
0xC	Power domain 1, within voltage domain, base address, offset from PERIPHBASE
0x10	Power domain 2, within voltage domain, base address, offset from PERIPHBASE
...	...
...	Power domain[N], where N is the total number of power domains in this voltage domain

Each power domain base address register in the table contains the offset from PERIPHBASE of a power domain configuration node. These registers also contain:

- Information about a single power domain
- Discovery information for clock domains that are associated with that power domain

### 16.2.1.2.3 Power domain configuration register region

Each power domain has a 4KB configuration register region that contains information about that power domain and all associated clock domains.

The following table shows the structure of the power domain configuration register region.

**Table 16-3: Contents of power domain configuration register region**

Offset	Contents
0x0	Power domain ID register
<b>NI-710AE clock domain configuration mapping</b>	
0x4	Number of clock domain regions present
0x8	Clock domain 0, within power domain, base address, offset from PERIPHBASE
0xC	Clock domain 1, within power domain, base address, offset from PERIPHBASE
0x10	Clock domain 2, within power domain, base address, offset from PERIPHBASE
...	...
...	Clock domain[N], where N is the total number of clock domains in this power domain
<b>NI-710AE power domain configuration</b>	
...	Extra configuration registers, as required

Each clock domain base address register in the table contains the offset from PERIPHBASE of a clock domain configuration node. These registers also contain:

- The information about a single clock domain
- The discovery information for leaf nodes that are associated with that clock domain

#### 16.2.1.2.4 Clock domain configuration register region

Each clock domain contains a 4KB configuration register region that contains information about that clock domain and all associated components.

The following table shows the structure of the clock domain configuration register region.

**Table 16-4: Contents of clock domain configuration register region**

Offset	Contents
0x0	Clock domain ID register
<b>NI-710AE component configuration mapping</b>	
0x4	Number of components present
0x8	Component 0, within clock domain, base address, offset from PERIPHBASE
0xC	Component 1, within clock domain, base address, offset from PERIPHBASE
0x10	Component 2, within clock domain, base address, offset from PERIPHBASE
...	...
...	Component[N], where N is the total number of components in this clock domain
<b>NI-710AE clock domain configuration</b>	
...	Extra configuration registers, as required

Each component base address register in the table contains the offset from PERIPHBASE for a component configuration node for a single component.

#### 16.2.1.2.5 Component configuration register region

Each component contains a 4KB configuration register region that contains information about that component, and all associated subfeatures.

NI-710AE instances can include the following component nodes:

- ASNI
- AMNI
- HSNi
- HMNI
- PMNI
- PMU

The following table shows the organization of the component node register region.

**Table 16-5: Component configuration register regions**

Offset	Category
0x0000	Node information
<b>Subfeature configuration region mapping</b>	
0x0024	Number of subfeatures
0x0028	Subfeature 0 type
0x002C	Subfeature 0, within component, base address, offset from PERIPHBASE
<b>Component configuration region mapping</b>	
...	...

#### 16.2.1.2.6 Subfeature configuration register region

Each subfeature contains a 4KB configuration register region that contains information about the subfeature. NI-710AE xSNIs and xMNIs support Access Protection Units (APUs) as subfeatures.

For more information about the APU registers, see [APU register summary](#).

## 16.2.2 Discovery flow

The discovery process uses a specific flow to identify the information about each type of logical unit in turn. As the process reads information at each layer, it builds a discovery tree, starting at the global configuration node and extending out to the leaf nodes.

All the NI-710AE configuration registers are mapped to an address range starting at PERIPHBASE, so PERIPHBASE is the starting point for discovery. You set the reset value of PERIPHBASE with Socrates.

Software discovery comprises the following steps:

1. Read information in the first 4KB region at PERIPHBASE.

This information determines:

- The number of voltage domains in NI-710AE
- The offset from PERIPHBASE for each 4KB voltage domain address region

2. Read information in the 4KB region that is associated with each voltage domain.

This information determines:

- The power domains that are associated with the voltage domain
- The topology information for these components
- The offset from PERIPHBASE for the 4KB base region of each power domain

3. Read information in the 4KB region that is associated with each power domain.

This information determines:

- The clock domains that are associated with the power domain
  - The topology information for these components
  - The offset from PERIPHBASE for the 4KB base region of each clock domain
4. Read information in the 4KB region that is associated with each clock domain.

This information determines:

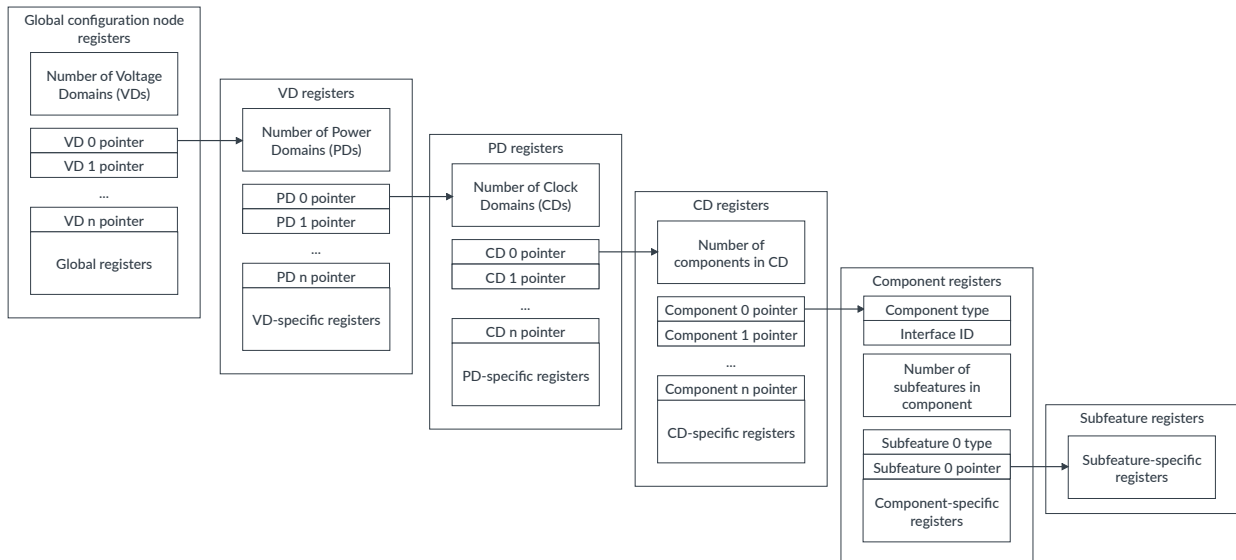
- The components that are associated with the clock domain
  - The topology information for these components
  - The offset from PERIPHBASE for the 4KB base region of each component
5. Read information in the 4KB region that is associated with the component.
- This information determines:
- The type of component
  - The configuration details of the component
  - The subfeatures that are associated with the component
  - The offset from PERIPHBASE for each 4KB base region of each subfeature
6. Read information in the 4KB region that is associated with the subfeature for the component.

This information determines the configuration details of the subfeature

With this sequence, software can build a list of all the units in the system and the addresses of their respective configuration regions.

The following figure shows the different layers of the discovery tree and how they interact with each other.

**Figure 16-2: Access mechanism**



When the software has identified all subfeature nodes for all components in the configuration, the discovery process is complete. At the end of the discovery process, software builds a discovery tree that provides:

- All pointers to the 4KB configuration node register regions corresponding to the voltage domains.
- For each voltage domain identified by a voltage domain ID, all pointers to the 4KB configuration node register regions corresponding to the power domains.
- For each power domain identified by the power domain ID, all pointers to the 4KB configuration node register regions corresponding to the clock domains.
- For each clock domain identified by the clock domain ID:
  - All pointers to the 4KB configuration node register regions corresponding to all the component nodes. The component nodes are the xSNI nodes, xMNI nodes, and the PMU node in that clock domain.
  - All pointers to the 4KB configuration nodes corresponding to any subfeature that is associated with each component node.
- For each configuration node, the location of the node type, Node ID, and other node information.

## 16.3 Configuration register address region calculation

When configuring NI-710AE, you must specify the size of the address region, as the size depends on your design.

Each Configuration Network Interface (CFGNI) occupies 4KB of the address map. The final number of configuration nodes in your design depends on the number of:

- Voltage domains
- Power domains
- Clock domains
- Endpoints. The number of endpoints is the sum of the number of ASNIs, AMNIs, HSNIs, HMNIs, and PMNIs in your design.
- Performance Monitoring Units (PMUs). The NI-710AE design contains one PMU for each clock domain, so the number of PMUs is equivalent to the number of clock domains in your design.

To calculate the size of the configuration register address region, use the following equation:

$$\text{Config space (in KB)} = 4 \times (1 + V + P + 2C + 2E)$$

Where:

**V**

Number of voltage domains

**P**

Number of power domains

**C**

Number of clock domains

**E**

Number of endpoints

All NI-710AE configurations have one CFGNI that contains the global registers. The global CFGNI is accounted for in the equation.

## 16.4 Configuration address space example for design with multiple voltage, power, and clock domains

NI-710AE contains multiple voltage, power, and clock domains that are configurable. The number of domains in your design affects the size of the configuration address space and the layout of the NI-710AE programmers view.

The configurable topology of NI-710AE alters the programmers view by changing the number of Configuration Network Interfaces (CFGNIs) required. To illustrate how the configurable design of NI-710AE affects the programmers view, consider an example configuration, which contains:

- Two voltage domains
- Four power domains
- Eight clock domains
- Eight PMUs
- Eight ASNIs



- Seven AMNIs
- Three HSNIs
- Three HMNIs
- Three PMNIs

The following table shows the programmers view for the example configuration.

**Table 16-6: Example programmers view for multiple voltage, power, and clock domain NI-710AE configuration**

Offset	Contents
0KB	Global registers
4KB	Voltage domain 0 registers
8KB	Power domain 0 registers
12KB	Clock domain 0 registers
16KB	ASNI 0 registers
20KB	ASNI 0 subfeature 0 registers
24KB	AMNI 0 registers
28KB	AMNI 0 subfeature 0 registers
32KB	PMU 0 registers
36KB	Clock domain 1 registers
40KB	ASNI 1 registers
44KB	ASNI 1 subfeature 0 registers
48KB	AMNI 1 registers
52KB	AMNI 1 subfeature 0 registers
56KB	PMU 1 registers
60KB	Power domain 1 registers
64KB	Clock domain 2 registers
68KB	ASNI 2 registers
72KB	ASNI 2 subfeature 0 registers
76KB	AMNI 2 registers
80KB	AMNI 2 subfeature 0 registers
84KB	HSNI 0 registers
88KB	HSNI 0 subfeature 0 registers
92KB	HMNI 0 registers
96KB	HMNI 0 subfeature 0 registers
100KB	PMNI 0 registers
104KB	PMNI 0 subfeature 0 registers
108KB	PMU 2 registers
112KB	Clock domain 3 registers
116KB	ASNI 3 registers
120KB	ASNI 3 subfeature 0 registers
124KB	AMNI 3 registers

Offset	Contents
128KB	AMNI 3 subfeature 0 registers
132KB	PMU 3 registers
136KB	Voltage domain 1 registers
140KB	Power domain 2 registers
144KB	Clock domain 4 registers
148KB	ASNI 4 registers
152KB	ASNI 4 subfeature 0 registers
156KB	AMNI 4 registers
160KB	AMNI 4 subfeature 0 registers
164KB	HSNI 1 registers
168KB	HSNI 1 subfeature 0 registers
172KB	HMNI 1 registers
176KB	HMNI 1 subfeature 0 registers
180KB	PMNI 1 registers
184KB	PMNI 1 subfeature 0 registers
188KB	PMU 4 registers
192KB	Clock domain 5 registers
196KB	ASNI 5 registers
200KB	ASNI 5 subfeature 0 registers
204KB	AMNI 5 registers
208KB	AMNI 5 subfeature 0 registers
212KB	PMU 5 registers
216KB	Power domain 3 registers
220KB	Clock domain 6 registers
224KB	ASNI 6 registers
228KB	ASNI 6 subfeature 0 registers
232KB	AMNI 6 registers
236KB	AMNI 6 subfeature 0 registers
240KB	PMU 6 registers
244KB	Clock domain 7 registers
248KB	ASNI 7 registers
252KB	ASNI 7 subfeature 0 registers
256KB	HSNI 2 registers
260KB	HSNI 2 subfeature 0 registers
264KB	HMNI 2 registers
268KB	HMNI 2 subfeature 0 registers
272KB	PMNI 2 registers
276KB	PMNI 2 subfeature 0 registers
280KB	PMU 7 registers

Each node type within NI-710AE requires a unique ID to enable device discovery to determine the set of registers at each configuration region.

## 16.5 Global register summary

This section describes the Global registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-7: Global register summary**

Offset	Name	Type	Reset	Width	Description
0x0	<a href="#">node_type</a>	RO	0x0	32-bit	This register identifies the node type as global or base registers.
0x04	<a href="#">child_node_info</a>	RO	See individual bit resets.	32-bit	This register identifies the number of voltage domains that are present in the interconnect system.
0x008	<a href="#">vd_pointers</a>	RO	See individual bit resets.	32-bit	This register points to the offset from the peripheral base, for the base address of the voltage domain register region.
0xF08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register contains controls for specifying access security requirements for global domain registers.
0xFD0	<a href="#">peripheral_id4</a>	RO	See individual bit resets.	32-bit	This register indicates the number of register blocks that are occupied and the value for bits[11:8] of the JEP106 code that identifies Arm.
0xFD4	<a href="#">peripheral_id5</a>	RO	0x00000000	32-bit	This register is reserved.
0xFD8	<a href="#">peripheral_id6</a>	RO	0x00000000	32-bit	This register is reserved.
0xFDC	<a href="#">peripheral_id7</a>	RO	0x00000000	32-bit	This register is reserved.
0xFE0	<a href="#">peripheral_id0</a>	RO	0x0000003D	32-bit	This register indicates the value for bits[7:0] of the interconnect part number.
0xFE4	<a href="#">peripheral_id1</a>	RO	0x000000B4	32-bit	This register indicates the value for bits[3:0] of the JEP106 ID code that identifies Arm and bits[11:8] of the interconnect part number.
0xFE8	<a href="#">peripheral_id2</a>	RO	0x0000000B	32-bit	This register indicates the product version, and the value for bits[6:4] of the JEP106 ID code that identifies Arm.
0xFEC	<a href="#">peripheral_id3</a>	RO	0x00000000	32-bit	This register indicates the Arm-approved ECO number, and the customer modification number.
0xFF0	<a href="#">component_id0</a>	RO	0x0000000D	32-bit	This register identifies the interconnect as an Arm component.
0xFF4	<a href="#">component_id1</a>	RO	0x000000F0	32-bit	This register identifies the interconnect as an Arm component.
0xFF8	<a href="#">component_id2</a>	RO	0x00000005	32-bit	This register identifies the interconnect as an Arm component.
0xFFC	<a href="#">component_id3</a>	RO	0x000000B1	32-bit	This register identifies the interconnect as an Arm component.

### 16.5.1 Global node\_type register

This register identifies the node type as global or base registers.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x0

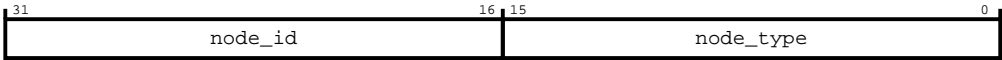
**Type**  
RO

**Reset value**  
0x0

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the node\_type register bit assignments.

**Figure 16-3: Bit assignment diagram for the node\_type register**



The following table shows the node\_type register bit descriptions.

**Table 16-8: node\_type bit descriptions**

Bits	Name	Description	Type	Reset
[31:16]	node_id	The value of this field is 0x0000 for the global register region	RO	0x0
[15:0]	node_type	The value of this field is 0x0000, indicating that the associated node is a global register node	RO	0x0

16.5.2 Global child\_node\_info register

This register identifies the number of voltage domains that are present in the interconnect system.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x04

Type

RO

Reset value

See individual bit resets.

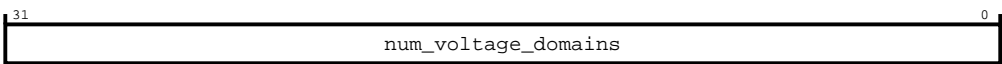
Constraints

None.

Bit descriptions

The following figure shows the child\_node\_info register bit assignments.

Figure 16-4: Bit assignment diagram for the child\_node\_info register



The following table shows the child\_node\_info register bit descriptions.

Table 16-9: child\_node\_info bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	num_voltage_domains	The value of this field is the number of voltage domains that are present in the interconnect	RO	Configuration dependent

16.5.3 Global vd\_pointers register

This register points to the offset from the peripheral base, for the base address of the voltage domain register region.

Configurations

The number of registers of this type that are present depends on the number of voltage domains in the NI-710AE configuration.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x008

Type

RO

Reset value

See individual bit resets.

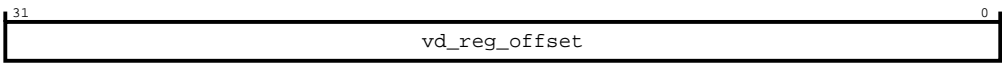
Constraints

None.

Bit descriptions

The following figure shows the vd\_pointers register bit assignments.

Figure 16-5: Bit assignment diagram for the vd\_pointers register



The following table shows the `vd_pointers` register bit descriptions.

Table 16-10: `vd_pointers` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	<code>vd_reg_offset</code>	Offset from the peripheral base, for the base address of the voltage domain register region	RO	Configuration dependent

16.5.4 Global `secure_access` register

This register contains controls for specifying access security requirements for global domain registers.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF08

Type

RW

Reset value

0x00000000

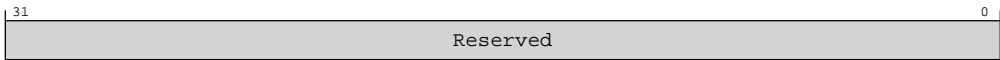
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `secure_access` register bit assignments.

Figure 16-6: Bit assignment diagram for the secure\_access register



The following table shows the secure\_access register bit descriptions.

Table 16-11: secure\_access bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0

16.5.5 Global peripheral\_id4 register

This register indicates the number of register blocks that are occupied and the value for bits[11:8] of the JEP106 code that identifies Arm.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFD0

Type

RO

Reset value

See individual bit resets.

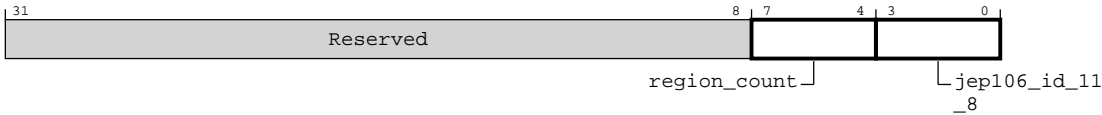
Constraints

None.

Bit descriptions

The following figure shows the peripheral\_id4 register bit assignments.

Figure 16-7: Bit assignment diagram for the peripheral\_id4 register



The following table shows the peripheral\_id4 register bit descriptions.

**Table 16-12: peripheral\_id4 bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:4]	region_count	The log_2 value of the number of register blocks that are occupied for the interconnect programmers view	RO	Configuration dependent
[3:0]	jep106_id_11_8	Bits[11:8] of the JEP106 ID code that identifies Arm value of 0x4	RO	0b0100

## 16.5.6 Global peripheral\_id5 register

This register is reserved.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0xFD4

#### Type

RO

#### Reset value

0x00000000

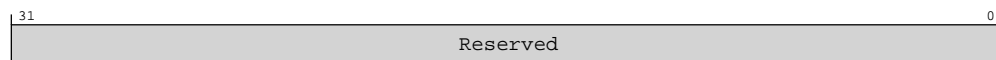
### Constraints

None.

### Bit descriptions

The following figure shows the peripheral\_id5 register bit assignments.

**Figure 16-8: Bit assignment diagram for the peripheral\_id5 register**



The following table shows the peripheral\_id5 register bit descriptions.



**Table 16-13: peripheral\_id5 bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved	RO	0×0

## 16.5.7 Global peripheral\_id6 register

This register is reserved.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0×FD8

#### Type

RO

#### Reset value

0×00000000

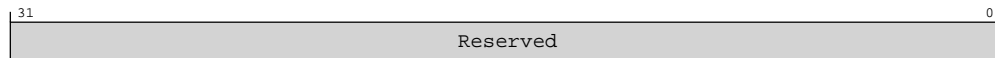
### Constraints

None.

### Bit descriptions

The following figure shows the peripheral\_id6 register bit assignments.

**Figure 16-9: Bit assignment diagram for the peripheral\_id6 register**



The following table shows the peripheral\_id6 register bit descriptions.

**Table 16-14: peripheral\_id6 bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved	RO	0×0

### 16.5.8 Global peripheral\_id7 register

This register is reserved.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xFDC

##### Type

RO

##### Reset value

0x00000000

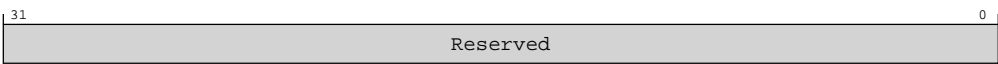
#### Constraints

None.

#### Bit descriptions

The following figure shows the peripheral\_id7 register bit assignments.

**Figure 16-10: Bit assignment diagram for the peripheral\_id7 register**



The following table shows the peripheral\_id7 register bit descriptions.

**Table 16-15: peripheral\_id7 bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved	RO	0x0

### 16.5.9 Global peripheral\_id0 register

This register indicates the value for bits[7:0] of the interconnect part number.

#### Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFE0

Type

RO

Reset value

0x0000003D

Constraints

None.

Bit descriptions

The following figure shows the peripheral\_id0 register bit assignments.

Figure 16-11: Bit assignment diagram for the peripheral\_id0 register



The following table shows the peripheral\_id0 register bit descriptions.

Table 16-16: peripheral\_id0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:0]	part_number	Bits[7:0] of the interconnect part number with a value of 0x3B	RO	0x3d

16.5.10 Global peripheral\_id1 register

This register indicates the value for bits[3:0] of the JEP106 ID code that identifies Arm and bits[11:8] of the interconnect part number.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFE4

Type

RO

Reset value

0x00000B4

Constraints

None.

Bit descriptions

The following figure shows the peripheral\_id1 register bit assignments.

Figure 16-12: Bit assignment diagram for the peripheral\_id1 register



The following table shows the peripheral\_id1 register bit descriptions.

Table 16-17: peripheral\_id1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:4]	jep106_id_3_0	Bits[3:0] of the JEP106 ID code that identifies Arm with the value of 0xB	RO	0b1011
[3:0]	part_number	Bits[11:8] of the interconnect part number with the value of 0x4	RO	0b0100

16.5.11 Global peripheral\_id2 register

This register indicates the product version, and the value for bits[6:4] of the JEP106 ID code that identifies Arm.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFE8

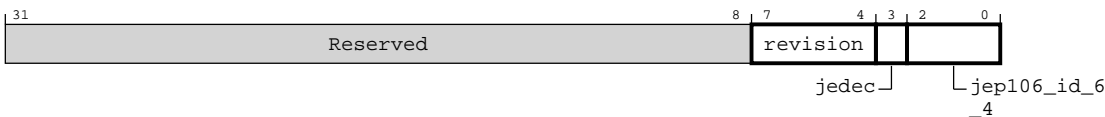
**Type**  
RO

**Reset value**  
0x0000001B

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the peripheral\_id2 register bit assignments.

**Figure 16-13: Bit assignment diagram for the peripheral\_id2 register**



The following table shows the peripheral\_id2 register bit descriptions.

**Table 16-18: peripheral\_id2 bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved.	RO	0x0
[7:4]	revision	Product revision:  <b>0x0</b> r0p0 EAC release  <b>0x1</b> r0p1 EAC maintenance and safety pack release	RO	0b0001
[3]	jedec	When set, this bit indicates that the JEP106 ID code is used and has a value of 1.	RO	1
[2:0]	jep106_id_6_4	Bits[6:4] of the JEP106 ID code that identifies Arm and has a value of 0b011.	RO	0b011

16.5.12 Global peripheral\_id3 register

This register indicates the Arm-approved ECO number, and the customer modification number.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

Address offset

0xFEC

Type

RO

Reset value

0x00000000

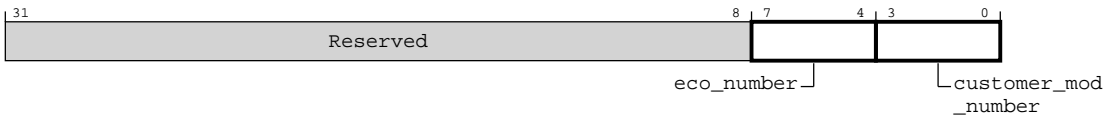
Constraints

None.

Bit descriptions

The following figure shows the peripheral\_id3 register bit assignments.

Figure 16-14: Bit assignment diagram for the peripheral\_id3 register



The following table shows the peripheral\_id3 register bit descriptions.

Table 16-19: peripheral\_id3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved.	RO	0x0
[7:4]	eco_number	Arm approved ECO number. Use the ECOREVNUM input to modify this value. For more information, see the topic on power, clock, reset, IDM, and other control signals in the signal descriptions.	RO	0b0000
[3:0]	customer_mod_number	The customer modification number. Do not modify this number unless you have permission from Arm.	RO	0b0000

16.5.13 Global component\_id0 register

This register identifies the interconnect as an Arm component.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFF0

Type

RO

Reset value

0x000000D

Constraints

None.

Bit descriptions

The following figure shows the component\_id0 register bit assignments.

Figure 16-15: Bit assignment diagram for the component\_id0 register



The following table shows the component\_id0 register bit descriptions.

Table 16-20: component\_id0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:0]	component_id	The component_id identifies the interconnect as an Arm component and has a value of 0x0D	RO	0xd

16.5.14 Global component\_id1 register

This register identifies the interconnect as an Arm component.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFF4

Type

RO

Reset value

0x000000F0

Constraints

None.

Bit descriptions

The following figure shows the component\_id1 register bit assignments.

Figure 16-16: Bit assignment diagram for the component\_id1 register



The following table shows the component\_id1 register bit descriptions.

Table 16-21: component\_id1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:0]	component_id	The component_id identifies the interconnect as an Arm component and has a value of 0xF, 0 (Arm® PrimeCell)	RO	0xF0

16.5.15 Global component\_id2 register

This register identifies the interconnect as an Arm component.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFF8

Type

RO

Reset value

0x00000005

Constraints

None.



Bit descriptions

The following figure shows the component\_id2 register bit assignments.

Figure 16-17: Bit assignment diagram for the component\_id2 register



The following table shows the component\_id2 register bit descriptions.

Table 16-22: component\_id2 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:0]	component_id	The component_id identifies the interconnect as an Arm component and has a value of 0x5	RO	0x5

16.5.16 Global component\_id3 register

This register identifies the interconnect as an Arm component.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFC

Type

RO

Reset value

0x000000B1

Constraints

None.

Bit descriptions

The following figure shows the component\_id3 register bit assignments.

**Figure 16-18: Bit assignment diagram for the component\_id3 register**



The following table shows the component\_id3 register bit descriptions.

**Table 16-23: component\_id3 bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved	RO	0x0
[7:0]	component_id	The component_id identifies the interconnect as an Arm component and has a value of 0xB1	RO	0xb1

## 16.6 Voltage domain register summary

This section describes the Voltage domain registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-24: Voltage domain register summary**

Offset	Name	Type	Reset	Width	Description
0x0	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies node type as voltage domain node.
0x04	<a href="#">child_node_info</a>	RO	See individual bit resets.	32-bit	This register indicates the number of power domains that are present in the voltage domain.
0x008	<a href="#">pd_pointers</a>	RO	See individual bit resets.	32-bit	This register points to the offset from the peripheral base, for the base address of the power domain register region.
0xF08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register contains controls for specifying access security requirements for voltage domain registers.

### 16.6.1 Voltage domain node\_type register

This register identifies node type as voltage domain node.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x0

Type

RO

Reset value

See individual bit resets.

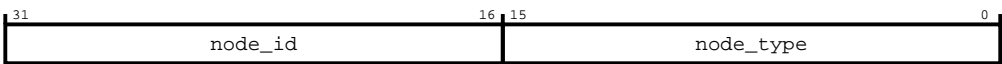
Constraints

None.

Bit descriptions

The following figure shows the node\_type register bit assignments.

Figure 16-19: Bit assignment diagram for the node\_type register



The following table shows the node\_type register bit descriptions.

Table 16-25: node\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	node_id	The voltage domain ID that is assigned during network construction	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x0001, indicating that the associated node is a power domain node.	RO	0x1

16.6.2 Voltage domain child\_node\_info register

This register indicates the number of power domains that are present in the voltage domain.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RO

Reset value

See individual bit resets.

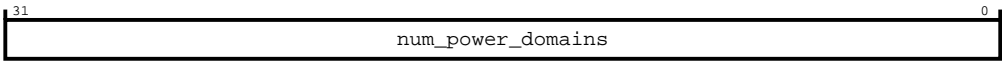
Constraints

None.

Bit descriptions

The following figure shows the child\_node\_info register bit assignments.

Figure 16-20: Bit assignment diagram for the child\_node\_info register



The following table shows the child\_node\_info register bit descriptions.

Table 16-26: child\_node\_info bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	num_power_domains	The number of power domains, leaf nodes, that are present in the voltage domain.	RO	Configuration dependent

16.6.3 Voltage domain pd\_pointers register

This register points to the offset from the peripheral base, for the base address of the power domain register region.

Configurations

The number of registers of this type that are present depends on the number of power domains in the parent voltage domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x008

Type

RO

Reset value

See individual bit resets.

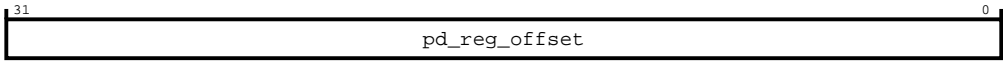
Constraints

None.

Bit descriptions

The following figure shows the pd\_pointers register bit assignments.

Figure 16-21: Bit assignment diagram for the pd\_pointers register



The following table shows the pd\_pointers register bit descriptions.

Table 16-27: pd\_pointers bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	pd_reg_offset	The offset from the peripheral base, for the base address of the power domain register region.	RO	Configuration dependent

16.6.4 Voltage domain secure\_access register

This register contains controls for specifying access security requirements for voltage domain registers.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF08

Type

RW

Reset value

0x00000000

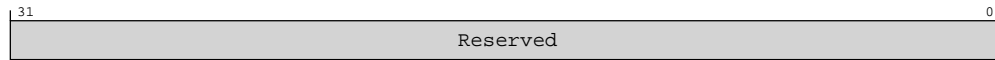
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the secure\_access register bit assignments.

**Figure 16-22: Bit assignment diagram for the secure\_access register**



The following table shows the secure\_access register bit descriptions.

**Table 16-28: secure\_access bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0

## 16.7 Power domain register summary

This section describes the Power domain registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-29: Power domain register summary**

Offset	Name	Type	Reset	Width	Description
0x0	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a power domain node.
0x04	<a href="#">child_node_info</a>	RO	See individual bit resets.	32-bit	This register indicates the number of clock domains that are present in the power domain.
0x008	<a href="#">cd_pointers</a>	RO	See individual bit resets.	32-bit	This register points to the offset from the peripheral base, for the base address of the clock domain register region of the power domain.
0x900	<a href="#">endpoint_pd_irq_status</a>	RO	0x00000000	32-bit	This register contains information about status of non-IDM interrupts from Secure transactions in the power domain.
0x904	<a href="#">endpoint_pd_irq_control</a>	RW	0x00000000	32-bit	This register contains controls for configuring the IDM interrupts of Secure transactions in the power domain.
0x908	<a href="#">idm_pd_error_status</a>	RO	0x00000000	32-bit	This register contains information about the error status of Secure transactions in the power domain.
0x90C	<a href="#">idm_pd_error_control</a>	RW	0x00000000	32-bit	This register contains controls for configuring the error interrupts of Secure transactions in the power domain.
0x910	<a href="#">idm_pd_timeout_status</a>	RO	0x00000000	32-bit	This register contains information about the timeout status of Secure transactions in the power domain.
0x914	<a href="#">idm_pd_timeout_control</a>	RW	0x00000000	32-bit	This register contains controls for configuring the timeout interrupts of Secure transactions in the power domain.
0x918	<a href="#">idm_pd_reset_status</a>	RO	0x00000000	32-bit	This register contains information about the reset access status of Secure transactions in the power domain.
0x91C	<a href="#">idm_pd_reset_control</a>	RW	0x00000000	32-bit	This register contains controls for configuring the reset interrupts of Secure transactions in the power domain.
0x920	<a href="#">idm_pd_access_status</a>	RO	0x00000000	32-bit	This register contains information about the isolation access status of Secure transactions in the power domain.

Offset	Name	Type	Reset	Width	Description
0x924	<a href="#">idm_pd_access_control</a>	RW	0x00000000	32-bit	This register contains controls for configuring the access interrupts of Secure transactions in the power domain.
0x928	<a href="#">endpoint_pd_irq_status_ns</a>	RO	0x00000000	32-bit	This register contains information about status of non-IDM interrupts from Non-secure transactions in the power domain.
0x92C	<a href="#">endpoint_pd_irq_control_ns</a>	RW	0x00000000	32-bit	This register contains controls for configuring the interrupts of Non-secure transactions in the power domain.
0x930	<a href="#">idm_pd_error_status_ns</a>	RO	0x00000000	32-bit	This register contains information about the error status of Non-secure transactions in the power domain.
0x934	<a href="#">idm_pd_error_control_ns</a>	RW	0x00000000	32-bit	This register contains controls for configuring the error interrupts of Non-secure transactions in the power domain.
0x938	<a href="#">idm_pd_timeout_status_ns</a>	RO	0x00000000	32-bit	This register contains information about the timeout status of Non-secure transactions in the power domain.
0x93C	<a href="#">idm_pd_timeout_control_ns</a>	RW	0x00000000	32-bit	This register contains controls for configuring the timeout interrupts of Non-secure transactions in the power domain.
0x940	<a href="#">idm_pd_reset_status_ns</a>	RO	0x00000000	32-bit	This register contains information about the reset access status of Non-secure transactions in the power domain.
0x944	<a href="#">idm_pd_reset_control_ns</a>	RW	0x00000000	32-bit	This register contains controls for configuring the reset interrupts of Non-secure transactions in the power domain.
0x948	<a href="#">idm_pd_access_status_ns</a>	RO	0x00000000	32-bit	This register contains information about the isolation access status of Non-secure transactions in the power domain.
0x94C	<a href="#">idm_pd_access_control_ns</a>	RW	0x00000000	32-bit	This register contains controls for configuring the access interrupts of Non-secure transactions in the power domain.
0xF08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register contains controls for specifying access security requirements for power domain registers.

## 16.7.1 Power domain node\_type register

This register identifies the node type as a power domain node.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x0

#### Type

RO

#### Reset value

See individual bit resets.

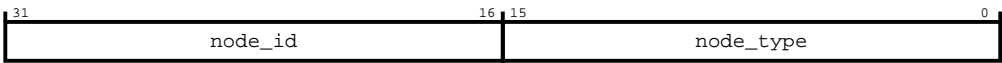
Constraints

None.

Bit descriptions

The following figure shows the node\_type register bit assignments.

Figure 16-23: Bit assignment diagram for the node\_type register



The following table shows the node\_type register bit descriptions.

Table 16-30: node\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	node_id	The power domain ID that is assigned during network construction.	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x0002, indicating that the associated node is a power domain node.	RO	0x2

16.7.2 Power domain child\_node\_info register

This register indicates the number of clock domains that are present in the power domain.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RO

Reset value

See individual bit resets.

Constraints

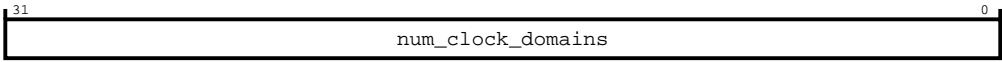
None.



Bit descriptions

The following figure shows the child\_node\_info register bit assignments.

Figure 16-24: Bit assignment diagram for the child\_node\_info register



The following table shows the child\_node\_info register bit descriptions.

Table 16-31: child\_node\_info bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	num_clock_domains	The value of this field is the number of clock domains that are present in the power domain.	RO	Configuration dependent

16.7.3 Power domain cd\_pointers register

This register points to the offset from the peripheral base, for the base address of the clock domain register region of the power domain.

Configurations

The number of registers of this type that are present depends on the number of clock domains in the parent power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x008

Type

RO

Reset value

See individual bit resets.

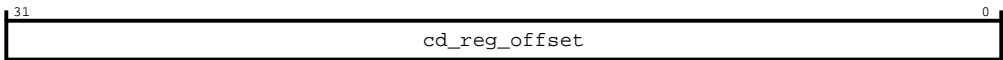
Constraints

None.

Bit descriptions

The following figure shows the cd\_pointers register bit assignments.

Figure 16-25: Bit assignment diagram for the cd\_pointers register



The following table shows the `cd_pointers` register bit descriptions.

Table 16-32: `cd_pointers` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	<code>cd_reg_offset</code>	Offset from the peripheral base, for the base address of the clock domain register region of the power domain.	RO	Configuration dependent

16.7.4 Power domain endpoint\_pd\_irq\_status register

This register contains information about status of non-IDM interrupts from Secure transactions in the power domain.

Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x900

Type

RO

Reset value

0x00000000

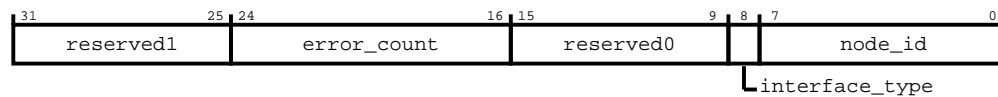
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `endpoint_pd_irq_status` register bit assignments.

**Figure 16-26: Bit assignment diagram for the endpoint\_pd\_irq\_status register**



The following table shows the endpoint\_pd\_irq\_status register bit descriptions.

**Table 16-33: endpoint\_pd\_irq\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	error_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an interrupt.  <b>0</b> xSNI raises an error interrupt first.  <b>1</b> xMNI raises an error interrupt first.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an interrupt.	RO	0x0

## 16.7.5 Power domain endpoint\_pd\_irq\_control register

This register contains controls for configuring the IDM interrupts of Secure transactions in the power domain.

### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x904

#### Type

RW

#### Reset value

0x00000000

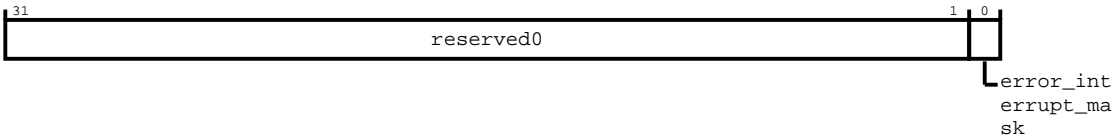
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the endpoint\_pd\_irq\_control register bit assignments.

Figure 16-27: Bit assignment diagram for the endpoint\_pd\_irq\_control register



The following table shows the endpoint\_pd\_irq\_control register bit descriptions.

Table 16-34: endpoint\_pd\_irq\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	error_interrupt_mask	The value of this field specifies whether all error interrupts are masked for Secure transactions in the power domain.  0 No Secure transaction error interrupts masked  1 All Secure transaction error interrupts masked	RW	0

16.7.6 Power domain idm\_pd\_error\_status register

This register contains information about the error status of Secure transactions in the power domain.

Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x908

Type

RO

## Reset value

0x00000000

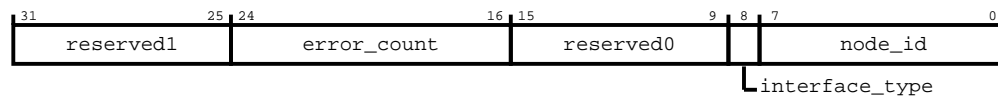
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_pd_error_status` register bit assignments.

**Figure 16-28: Bit assignment diagram for the `idm_pd_error_status` register**



The following table shows the `idm_pd_error_status` register bit descriptions.

### Table 16-35: idm\_pd\_error\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	error_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an error interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	<p>The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an error interrupt.</p> <p><b>0</b></p> <p>xSNI raises an error interrupt first.</p> <p><b>1</b></p> <p>xMNI raises an error interrupt first.</p>	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an error interrupt.	RO	0x0

### 16.7.7 Power domain idm\_pd\_error\_control register

This register contains controls for configuring the error interrupts of Secure transactions in the power domain.

## Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

## Attributes

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x90C

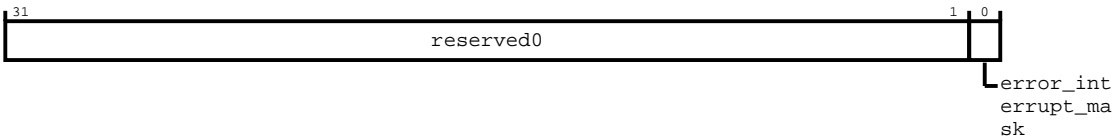
**Type**  
RW

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions.

**Bit descriptions**  
The following figure shows the idm\_pd\_error\_control register bit assignments.

**Figure 16-29: Bit assignment diagram for the idm\_pd\_error\_control register**



The following table shows the idm\_pd\_error\_control register bit descriptions.

**Table 16-36: idm\_pd\_error\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	error_interrupt_mask	The value of this field specifies whether all error interrupts are masked for Secure transactions in the power domain.  <b>0</b> No Secure transaction error interrupts masked  <b>1</b> All Secure transaction error interrupts masked	RW	0

16.7.8 Power domain idm\_pd\_timeout\_status register

This register contains information about the timeout status of Secure transactions in the power domain.

**Configurations**  
This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x910

Type

RO

Reset value

0x00000000

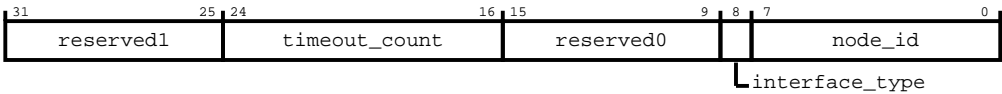
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_pd\_timeout\_status register bit assignments.

Figure 16-30: Bit assignment diagram for the idm\_pd\_timeout\_status register



The following table shows the idm\_pd\_timeout\_status register bit descriptions.

Table 16-37: idm\_pd\_timeout\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	timeout_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting a timeout interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises a timeout interrupt.  0 xSNI raises a timeout interrupt first.  1 xMNI raises a timeout interrupt first.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises a timeout interrupt.	RO	0x0

16.7.9 Power domain idm\_pd\_timeout\_control register

This register contains controls for configuring the timeout interrupts of Secure transactions in the power domain.

Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x914

Type

RW

Reset value

0x00000000

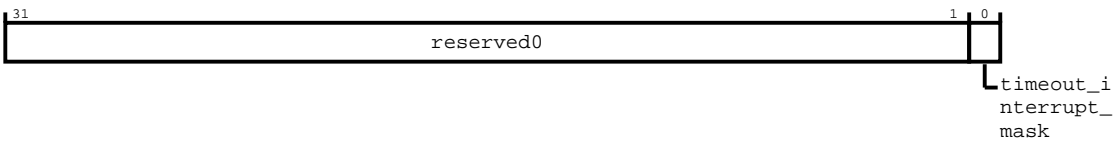
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_pd\_timeout\_control register bit assignments.

Figure 16-31: Bit assignment diagram for the idm\_pd\_timeout\_control register



The following table shows the idm\_pd\_timeout\_control register bit descriptions.

Table 16-38: idm\_pd\_timeout\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0



Bits	Name	Description	Type	Reset
[0]	timeout_interrupt_mask	<p>The value of this field specifies whether all timeout interrupts are masked for Secure transactions in the power domain.</p> <p><b>0</b></p> <p>No Secure transaction timeout interrupts masked</p> <p><b>1</b></p> <p>All Secure transaction timeout interrupts masked</p>	RW	0

## 16.7.10 Power domain idm\_pd\_reset\_status register

This register contains information about the reset access status of Secure transactions in the power domain.

### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x918

#### Type

RO

#### Reset value

0x00000000

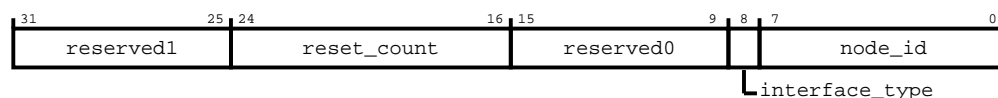
### Constraints

Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the idm\_pd\_reset\_status register bit assignments.

**Figure 16-32: Bit assignment diagram for the idm\_pd\_reset\_status register**



The following table shows the idm\_pd\_reset\_status register bit descriptions.

**Table 16-39: idm\_pd\_reset\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	reset_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an reset interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an activity while in reset interrupt.  <b>0</b> xSNI first raises an activity while in reset interrupt.  <b>1</b> xMNI first raises an activity while in reset interrupt.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an activity while in reset interrupt.	RO	0x0

### 16.7.11 Power domain idm\_pd\_reset\_control register

This register contains controls for configuring the reset interrupts of Secure transactions in the power domain.

#### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x91C

##### Type

RW

##### Reset value

0x00000000

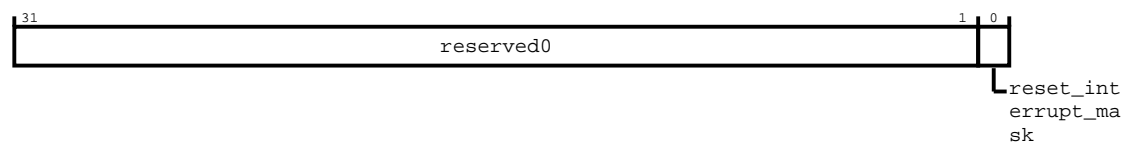
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_pd\_reset\_control register bit assignments.

Figure 16-33: Bit assignment diagram for the `idm_pd_reset_control` register



The following table shows the `idm_pd_reset_control` register bit descriptions.

Table 16-40: `idm_pd_reset_control` bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	reset_interrupt_mask	The value of this field specifies whether all reset interrupts are masked for Secure transactions in the power domain.  0 No Secure transaction reset interrupts masked  1 All Secure transaction reset interrupts masked	RW	0

16.7.12 Power domain `idm_pd_access_status` register

This register contains information about the isolation access status of Secure transactions in the power domain.

Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x920

Type

RO

Reset value

0x00000000

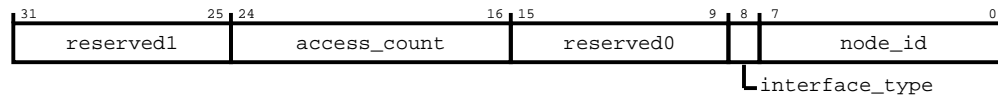
Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_pd_access_status` register bit assignments.

**Figure 16-34: Bit assignment diagram for the `idm_pd_access_status` register**



The following table shows the `idm_pd_access_status` register bit descriptions.

**Table 16-41: `idm_pd_access_status` bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	access_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an access interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an access interrupt.  <b>0</b> xSNI raises an access interrupt first.  <b>1</b> xMNI raises an access interrupt first.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an access interrupt.	RO	0x0

## 16.7.13 Power domain `idm_pd_access_control` register

This register contains controls for configuring the access interrupts of Secure transactions in the power domain.

### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x924

#### Type

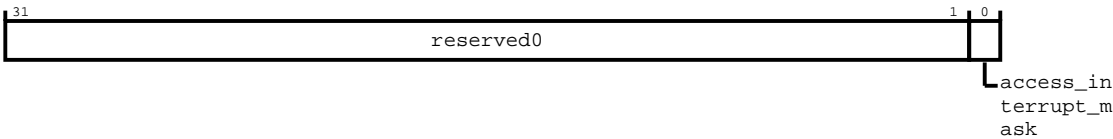
RW

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions.

**Bit descriptions**  
The following figure shows the idm\_pd\_access\_control register bit assignments.

**Figure 16-35: Bit assignment diagram for the idm\_pd\_access\_control register**



The following table shows the idm\_pd\_access\_control register bit descriptions.

**Table 16-42: idm\_pd\_access\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	access_interrupt_mask	The value of this field specifies whether all access interrupts are masked for Secure transactions in the power domain.  <b>0</b> No Secure transaction access interrupts masked  <b>1</b> All Secure transaction access interrupts masked	RW	0

16.7.14 Power domain endpoint\_pd\_irq\_status\_ns register

This register contains information about status of non-IDM interrupts from Non-secure transactions in the power domain.

**Configurations**  
This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x928

## Type

RO

## Reset value

0x00000000

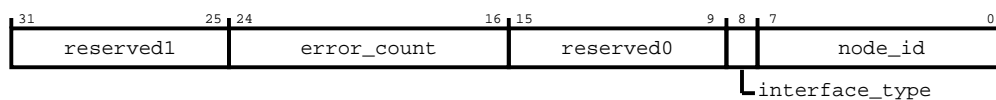
## Constraints

None.

## Bit descriptions

The following figure shows the endpoint\_pd\_irq\_status\_ns register bit assignments.

**Figure 16-36: Bit assignment diagram for the endpoint\_pd\_irq\_status\_ns register**



The following table shows the endpoint\_pd\_irq\_status\_ns register bit descriptions.

**Table 16-43: endpoint\_pd\_irq\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b00000000
[24:16]	error_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b00000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an interrupt.  <b>0</b> xSNI raises an error interrupt first.  <b>1</b> xMNI raises an error interrupt first.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an error interrupt.	RO	0x0

## 16.7.15 Power domain endpoint\_pd\_irq\_control\_ns register

This register contains controls for configuring the interrupts of Non-secure transactions in the power domain.

## Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x92C

Type

RW

Reset value

0x00000000

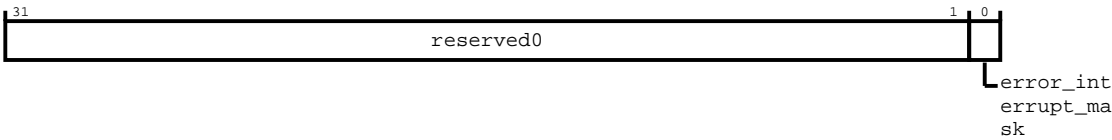
Constraints

None.

Bit descriptions

The following figure shows the endpoint\_pd\_irq\_control\_ns register bit assignments.

Figure 16-37: Bit assignment diagram for the endpoint\_pd\_irq\_control\_ns register



The following table shows the endpoint\_pd\_irq\_control\_ns register bit descriptions.

Table 16-44: endpoint\_pd\_irq\_control\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	error_interrupt_mask	<div>The value of this field specifies whether all interrupts are masked for Non-secure transactions in the power domain.</div> <div><b>0</b> No Non-secure transaction error interrupts masked</div> <div><b>1</b> All Non-secure transaction error interrupts masked</div>	RW	0

## 16.7.16 Power domain idm\_pd\_error\_status\_ns register

This register contains information about the error status of Non-secure transactions in the power domain.

### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x930

#### Type

RO

#### Reset value

0x00000000

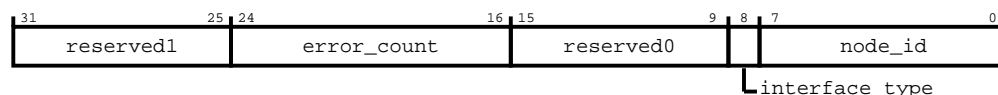
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_pd\_error\_status\_ns register bit assignments.

**Figure 16-38: Bit assignment diagram for the idm\_pd\_error\_status\_ns register**



The following table shows the idm\_pd\_error\_status\_ns register bit descriptions.

**Table 16-45: idm\_pd\_error\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	error_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an error interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000



Bits	Name	Description	Type	Reset
[8]	interface_type	<p>The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an error interrupt.</p> <p><b>0</b></p> <p>xSNI raises an error interrupt first.</p> <p><b>1</b></p> <p>xMNI raises an error interrupt first.</p>	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an error interrupt.	RO	0x0

### 16.7.17 Power domain idm\_pd\_error\_control\_ns register

This register contains controls for configuring the error interrupts of Non-secure transactions in the power domain.

## Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x934

## Type

RW

## Reset value

0x00000000

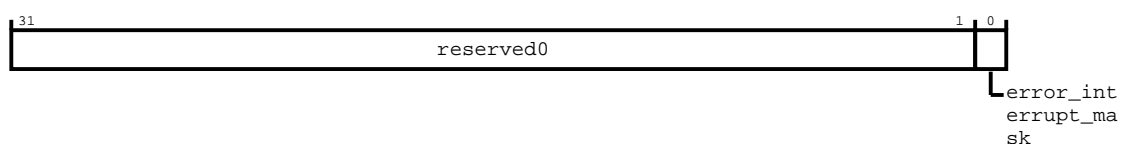
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_pd_error_control_ns` register bit assignments.

**Figure 16-39: Bit assignment diagram for the `idm_pd_error_control_ns` register**



The following table shows the `idm_pd_error_control_ns` register bit descriptions.

**Table 16-46: `idm_pd_error_control_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	<code>reserved0</code>	Reserved	RO	0x0
[0]	<code>error_interrupt_mask</code>	<p>The value of this field specifies whether all error interrupts are masked for Non-secure transactions in the power domain.</p> <p><b>0</b></p> <p>No Non-secure transaction error interrupts masked</p> <p><b>1</b></p> <p>All Non-secure transaction error interrupts masked</p>	RW	0

### 16.7.18 Power domain `idm_pd_timeout_status_ns` register

This register contains information about the timeout status of Non-secure transactions in the power domain.

#### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x938

##### Type

RO

##### Reset value

0x00000000

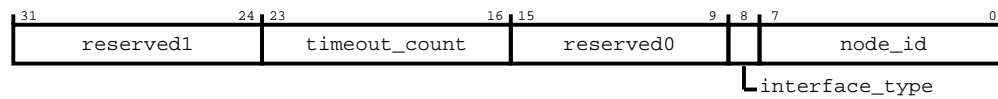
#### Constraints

None.

#### Bit descriptions

The following figure shows the `idm_pd_timeout_status_ns` register bit assignments.

**Figure 16-40: Bit assignment diagram for the idm\_pd\_timeout\_status\_ns register**



The following table shows the idm\_pd\_timeout\_status\_ns register bit descriptions.

**Table 16-47: idm\_pd\_timeout\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:24]	reserved1	Reserved	RO	0x0
[23:16]	timeout_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting a timeout interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises a timeout interrupt.  <b>0</b> xSNI raises a timeout interrupt first.  <b>1</b> xMNI raises a timeout interrupt first.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises a timeout interrupt.	RO	0x0

### 16.7.19 Power domain idm\_pd\_timeout\_control\_ns register

This register contains controls for configuring the timeout interrupts of Non-secure transactions in the power domain.

#### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x93C

##### Type

RW

##### Reset value

0x00000000

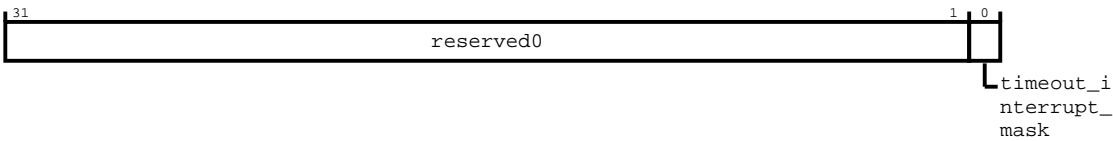
Constraints

None.

Bit descriptions

The following figure shows the `idm_pd_timeout_control_ns` register bit assignments.

Figure 16-41: Bit assignment diagram for the `idm_pd_timeout_control_ns` register



The following table shows the `idm_pd_timeout_control_ns` register bit descriptions.

Table 16-48: `idm_pd_timeout_control_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	timeout_interrupt_mask	The value of this field specifies whether all timeout interrupts are masked for Non-secure transactions in the power domain.  <b>0</b> No Non-secure transaction timeout interrupts masked  <b>1</b> All Non-secure transaction timeout interrupts masked	RW	0

16.7.20 Power domain `idm_pd_reset_status_ns` register

This register contains information about the reset access status of Non-secure transactions in the power domain.

Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x940

Type

RO

## Reset value

0x00000000

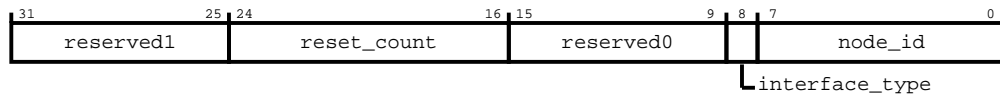
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_pd_reset_status_ns` register bit assignments.

**Figure 16-42: Bit assignment diagram for the `idm_pd_reset_status_ns` register**



The following table shows the `idm_pd_reset_status_ns` register bit descriptions.

**Table 16-49: `idm_pd_reset_status_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b00000000
[24:16]	reset_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an reset interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b00000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an activity while in reset interrupt.  <b>0</b> xSNI first raises an activity while in reset interrupt.  <b>1</b> xMNI first raises an activity while in reset interrupt.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an activity while in reset interrupt.	RO	0x0

## 16.7.21 Power domain `idm_pd_reset_control_ns` register

This register contains controls for configuring the reset interrupts of Non-secure transactions in the power domain.

## Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

## Attributes

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x944

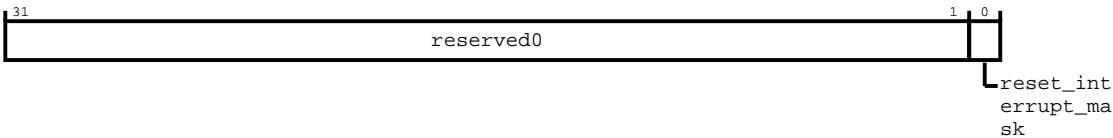
**Type**  
RW

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the `idm_pd_reset_control_ns` register bit assignments.

**Figure 16-43: Bit assignment diagram for the `idm_pd_reset_control_ns` register**



The following table shows the `idm_pd_reset_control_ns` register bit descriptions.

**Table 16-50: `idm_pd_reset_control_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	reset_interrupt_mask	The value of this field specifies whether all reset interrupts are masked for Non-secure transactions in the power domain.  0 No Non-secure transaction reset interrupts masked  1 All Non-secure transaction reset interrupts masked	RW	0

16.7.22 Power domain `idm_pd_access_status_ns` register

This register contains information about the isolation access status of Non-secure transactions in the power domain.

Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x948

Type

RO

Reset value

0x00000000

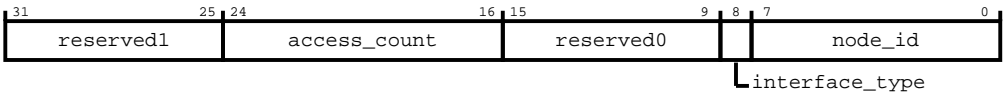
Constraints

None.

Bit descriptions

The following figure shows the idm\_pd\_access\_status\_ns register bit assignments.

Figure 16-44: Bit assignment diagram for the idm\_pd\_access\_status\_ns register



The following table shows the idm\_pd\_access\_status\_ns register bit descriptions.

Table 16-51: idm\_pd\_access\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:25]	reserved1	Reserved	RO	0b0000000
[24:16]	access_count	The value of this field specifies the number of endpoints in the power domain that are currently asserting an access interrupt.	RO	0x0
[15:9]	reserved0	Reserved	RO	0b0000000
[8]	interface_type	The value of this field specifies the endpoint type of the first endpoint in the power domain that raises an access interrupt.  0 xSNI raises an access interrupt first.  1 xMNI raises an access interrupt first.	RO	0
[7:0]	node_id	The value of this field specifies the node ID of the first endpoint in the power domain that raises an access interrupt.	RO	0x0

### 16.7.23 Power domain idm\_pd\_access\_control\_ns register

This register contains controls for configuring the access interrupts of Non-secure transactions in the power domain.

#### Configurations

This register is only present if IDM is enabled on at least one of the interfaces in the power domain.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x94C

##### Type

RW

##### Reset value

0x00000000

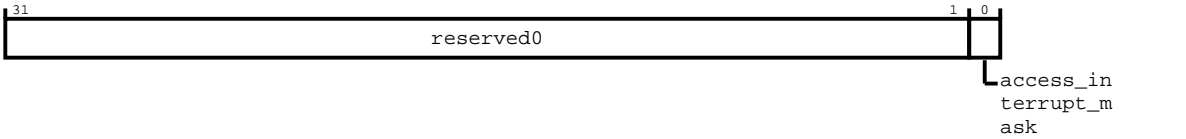
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_pd\_access\_control\_ns register bit assignments.

Figure 16-45: Bit assignment diagram for the idm\_pd\_access\_control\_ns register



The following table shows the idm\_pd\_access\_control\_ns register bit descriptions.

Table 16-52: idm\_pd\_access\_control\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[0]	access_interrupt_mask	When set to 1, enables mask of all error interrupts.	RW	0



16.7.24 Power domain secure\_access register

This register contains controls for specifying access security requirements for power domain registers.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF08

Type

RW

Reset value

0x00000000

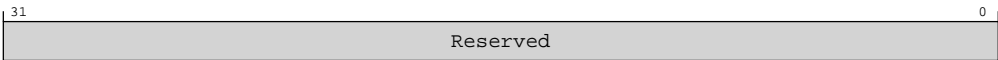
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the secure\_access register bit assignments.

Figure 16-46: Bit assignment diagram for the secure\_access register



The following table shows the secure\_access register bit descriptions.

Table 16-53: secure\_access bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0

## 16.8 Clock domain register summary

This section describes the Clock domain registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-54: Clock domain register summary**

Offset	Name	Type	Reset	Width	Description
0x0	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a clock domain register node.
0x04	<a href="#">child_node_info</a>	RO	See individual bit resets.	32-bit	This register indicates the number of network components that are present in the clock domain.
0x008	<a href="#">component_pointers</a>	RO	See individual bit resets.	32-bit	This register points to the offset from the peripheral base, for the base address of the component register region of the clock domain.
0xF08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register contains controls for specifying access security requirements for clock domain registers.

### 16.8.1 Clock domain node\_type register

This register identifies the node type as a clock domain register node.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets.

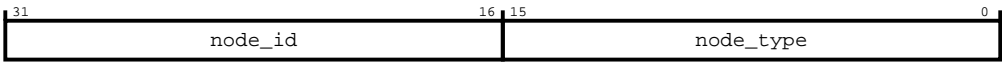
#### Constraints

None.

#### Bit descriptions

The following figure shows the node\_type register bit assignments.

Figure 16-47: Bit assignment diagram for the node\_type register



The following table shows the node\_type register bit descriptions.

Table 16-55: node\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	node_id	The clock domain ID that is assigned during network construction.	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x0003, indicating that the associated node contains clock domain registers for a particular power domain.	RO	0x3

16.8.2 Clock domain child\_node\_info register

This register indicates the number of network components that are present in the clock domain.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RO

Reset value

See individual bit resets.

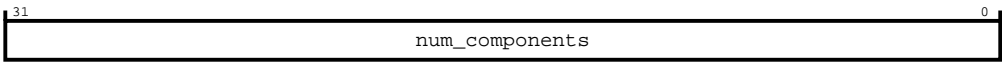
Constraints

None.

Bit descriptions

The following figure shows the child\_node\_info register bit assignments.

Figure 16-48: Bit assignment diagram for the child\_node\_info register



The following table shows the child\_node\_info register bit descriptions.

**Table 16-56: child\_node\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	num_components	The value of this field is the number of network components that are present in the clock domain.	RO	Configuration dependent

### 16.8.3 Clock domain component\_pointers register

This register points to the offset from the peripheral base, for the base address of the component register region of the clock domain.

#### Configurations

The number of registers of this type that are present depends on the number of interfaces in the parent clock domain.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x008

##### Type

RO

##### Reset value

See individual bit resets.

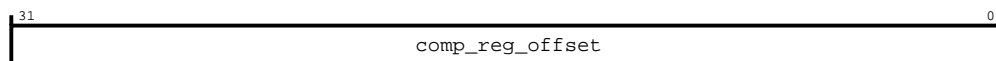
#### Constraints

None.

#### Bit descriptions

The following figure shows the component\_pointers register bit assignments.

**Figure 16-49: Bit assignment diagram for the component\_pointers register**



The following table shows the component\_pointers register bit descriptions.

**Table 16-57: component\_pointers bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	comp_reg_offset	A pointer to the offset from the peripheral base, for the base address of the component register region of the clock domain.	RO	Configuration dependent

## 16.8.4 Clock domain secure\_access register

This register contains controls for specifying access security requirements for clock domain registers.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0xF08

#### Type

RW

#### Reset value

0x00000000

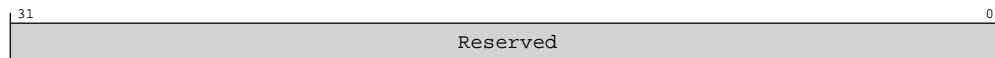
### Constraints

Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the secure\_access register bit assignments.

**Figure 16-50: Bit assignment diagram for the secure\_access register**



The following table shows the secure\_access register bit descriptions.

**Table 16-58: secure\_access bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0

## 16.9 PMU register summary

This section describes the PMU registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-59: PMU register summary**

Offset	Name	Type	Reset	Width	Description
0x0	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a node for PMU registers.
0x04	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register controls whether only Non-secure transactions can read and program the NI-710AE registers.
0x00000008	<a href="#">pmevcntr0</a>	RW	0x00000000	32-bit	Performance monitor 0 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x00000010	<a href="#">pmevcntr1</a>	RW	0x00000000	32-bit	Performance monitor 1 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x00000018	<a href="#">pmevcntr2</a>	RW	0x00000000	32-bit	Performance monitor 2 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x00000020	<a href="#">pmevcntr3</a>	RW	0x00000000	32-bit	Performance monitor 3 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x00000028	<a href="#">pmevcntr4</a>	RW	0x00000000	32-bit	Performance monitor 4 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x00000030	<a href="#">pmevcntr5</a>	RW	0x00000000	32-bit	Performance monitor 5 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clockdomain in the interconnect system.
0x00000038	<a href="#">pmevcntr6</a>	RW	0x00000000	32-bit	Performance monitor 6 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x00000040	<a href="#">pmevcntr7</a>	RW	0x00000000	32-bit	Performance monitor 7 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.
0x000000F8	<a href="#">pmccntr_l</a>	RW	0x00000000	32-bit	This register contains the value of lower 64-bit cycle counter bits[31:0].
0x000000FC	<a href="#">pmccntr_u</a>	RW	0x00000000	32-bit	This register contains the value of upper 64-bit cycle counter bits[63:32].
0x00000400	<a href="#">pmevtyper0</a>	RW	0x00000000	32-bit	Performance monitor event type filter 0. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.
0x00000404	<a href="#">pmevtyper1</a>	RW	0x00000000	32-bit	Performance monitor event type filter 1. Registers PMEVTYPER0-7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.
0x00000408	<a href="#">pmevtyper2</a>	RW	0x00000000	32-bit	Performance monitor event type filter 2. Registers PMEVTYPER0-7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.

Offset	Name	Type	Reset	Width	Description
0x0000040C	<a href="#">pmevtyper3</a>	RW	0x00000000	32-bit	Performance monitor event type filter 3. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.
0x00000410	<a href="#">pmevtyper4</a>	RW	0x00000000	32-bit	Performance monitor event type filter 4. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.
0x00000414	<a href="#">pmevtyper5</a>	RW	0x00000000	32-bit	Performance monitor event type filter 5. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.
0x00000418	<a href="#">pmevtyper6</a>	RW	0x00000000	32-bit	Performance monitor event type filter 6. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.
0x0000041C	<a href="#">pmevtyper7</a>	RW	0x00000000	32-bit	Performance monitor event type filter 7. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.
0x610	<a href="#">pmssr</a>	RO	0x00000001	32-bit	This register records the status of a performance event counter when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x614	<a href="#">pmovssr</a>	RO	0x00000000	32-bit	This register records the overflow status of a performance event counter when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x00000618	<a href="#">pmccntsr_l</a>	RO	0x00000000	32-bit	This register contains the snapshot value of the lower 64-bit cycle counter bits[31:0].
0x0000061C	<a href="#">pmccntsr_u</a>	RO	0x00000000	32-bit	This register contains the snapshot value of the upper 64-bit cycle counter bits[63:32].
0x00000620	<a href="#">pmevcntsr0</a>	RO	0x00000000	32-bit	Performance monitor 0 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x00000624	<a href="#">pmevcntsr1</a>	RO	0x00000000	32-bit	Performance monitor 1 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x00000628	<a href="#">pmevcntsr2</a>	RO	0x00000000	32-bit	Performance monitor 2 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x0000062C	<a href="#">pmevcntsr3</a>	RO	0x00000000	32-bit	Performance monitor 3 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x00000630	<a href="#">pmevcntsr4</a>	RO	0x00000000	32-bit	Performance monitor 4 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x00000634	<a href="#">pmevcntsr5</a>	RO	0x00000000	32-bit	Performance monitor 5 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x00000638	<a href="#">pmevcntsr6</a>	RO	0x00000000	32-bit	Performance monitor 6 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.

Offset	Name	Type	Reset	Width	Description
0x0000063C	<a href="#">pmevcntsr7</a>	RO	0x00000000	32-bit	Performance monitor 7 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>_PMUSNAPSHOTREQ input signal.
0x6F0	<a href="#">pmsscr</a>	RW	0x00000000	32-bit	This register captures a snapshot of the performance monitors.
0xC00	<a href="#">pmcntenset</a>	RW	0x00000000	32-bit	This register sets the performance monitors count enable.
0xC20	<a href="#">pmcntenclr</a>	RW	0x00000000	32-bit	This register clears the performance monitors count enable.
0xC40	<a href="#">pmintenset</a>	RW	0x00000000	32-bit	This register sets the performance monitors interrupt enable.
0xC60	<a href="#">pmintenclr</a>	RW	0x00000000	32-bit	This register clears the performance monitors interrupt enable.
0xC80	<a href="#">pmovsclr</a>	RW	0x00000000	32-bit	This register clears the performance monitors overflow flag status.
0xCC0	<a href="#">pmovsset</a>	RW	0x00000000	32-bit	This register sets the performance monitors overflow flag status.
0xD80	<a href="#">pmccgr</a>	RW	0x00000000	32-bit	This register controls the cycle counter clock gating enable.
0xE00	<a href="#">pmcfgr</a>	RO	0x00417F08	32-bit	This register contains configuration values for the performance monitors.
0xE04	<a href="#">pmcr</a>	RW	0x00000000	32-bit	This register controls the performance monitors.

## 16.9.1 PMU node\_type register

This register identifies the node type as a node for PMU registers.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x0

#### Type

RO

#### Reset value

See individual bit resets.

### Constraints

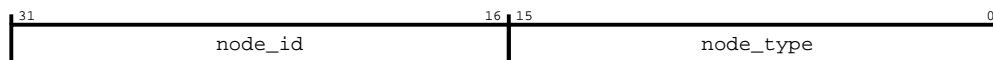
None.

### Bit descriptions

The following figure shows the node\_type register bit assignments.



**Figure 16-51: Bit assignment diagram for the node\_type register**



The following table shows the node\_type register bit descriptions.

**Table 16-60: node\_type bit descriptions**

Bits	Name	Description	Type	Reset
[31:16]	node_id	The PMU ID that is assigned during network construction	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x06, and it identifies the associated node type as a node for the PMU registers	RO	0x6

## 16.9.2 PMU secure\_access register

This register controls whether only Non-secure transactions can read and program the NI-710AE registers.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x04

#### Type

RW

#### Reset value

0x00000000

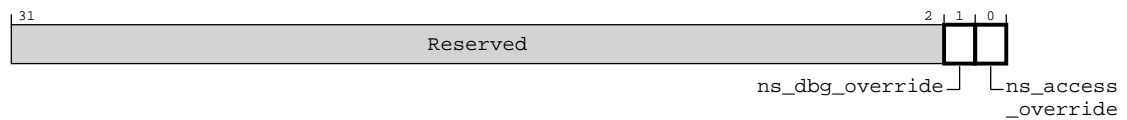
### Constraints

Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the secure\_access register bit assignments.

**Figure 16-52: Bit assignment diagram for the secure\_access register**



The following table shows the secure\_access register bit descriptions.

**Table 16-61: secure\_access bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[1]	ns_dbg_override	Debug monitor security override: <b>0</b> Disable. Non-secure access to the interconnect PMU and interface registers <b>1</b> Enable. Non-secure access to the interconnect PMU and interface registers	RW	0
[0]	ns_access_override	Non-secure access override: <b>0</b> Disable. Non-secure access to the NI-710AE registers <b>1</b> Enable. Non-secure access to the NI-710AE registers	RW	0

### 16.9.3 PMU pmevcntr0 register

Performance monitor 0 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x00000008

##### Type

RW

##### Reset value

0x00000000

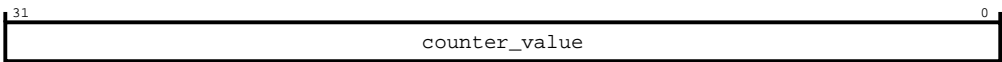
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr0 register bit assignments.

Figure 16-53: Bit assignment diagram for the pmevcntr0 register



The following table shows the pmevcntr0 register bit descriptions.

Table 16-62: pmevcntr0 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 0 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.4 PMU pmevcntr1 register

Performance monitor 1 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000010

Type

RW

Reset value

0x00000000

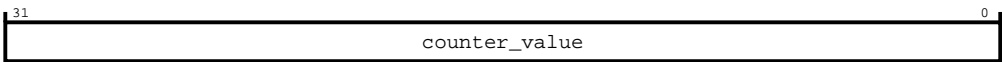
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr1 register bit assignments.

Figure 16-54: Bit assignment diagram for the pmevcntr1 register



The following table shows the pmevcntr1 register bit descriptions.

Table 16-63: pmevcntr1 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 1 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.5 PMU pmevcntr2 register

Performance monitor 2 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000018

Type

RW

Reset value

0x00000000

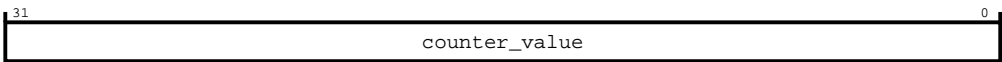
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr2 register bit assignments.

Figure 16-55: Bit assignment diagram for the pmevcntr2 register



The following table shows the pmevcntr2 register bit descriptions.

Table 16-64: pmevcntr2 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 2 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.6 PMU pmevcntr3 register

Performance monitor 3 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000020

Type

RW

Reset value

0x00000000

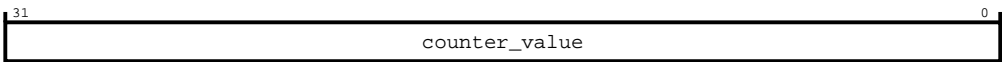
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr3 register bit assignments.

Figure 16-56: Bit assignment diagram for the pmevcntr3 register



The following table shows the pmevcntr3 register bit descriptions.

Table 16-65: pmevcntr3 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 3 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.7 PMU pmevcntr4 register

Performance monitor 4 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000028

Type

RW

Reset value

0x00000000

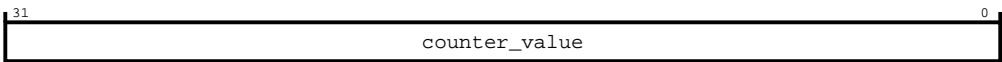
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr4 register bit assignments.

Figure 16-57: Bit assignment diagram for the pmevcntr4 register



The following table shows the pmevcntr4 register bit descriptions.

Table 16-66: pmevcntr4 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 4 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.8 PMU pmevcntr5 register

Performance monitor 5 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clockdomain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000030

Type

RW

Reset value

0x00000000

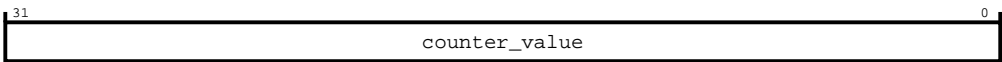
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr5 register bit assignments.

Figure 16-58: Bit assignment diagram for the pmevcntr5 register



The following table shows the pmevcntr5 register bit descriptions.

Table 16-67: pmevcntr5 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 5 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.9 PMU pmevcntr6 register

Performance monitor 6 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000038

Type

RW

Reset value

0x00000000



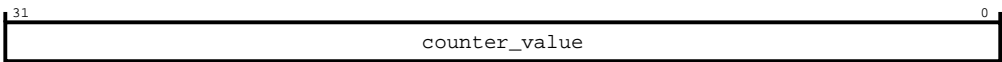
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr6 register bit assignments.

Figure 16-59: Bit assignment diagram for the pmevcntr6 register



The following table shows the pmevcntr6 register bit descriptions.

Table 16-68: pmevcntr6 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 6 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.10 PMU pmevcntr7 register

Performance monitor 7 event counter register. Registers pmevcntr0-pmevcntr7 record performance events that occur within each clock domain in the interconnect system.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000040

Type

RW

Reset value

0x00000000

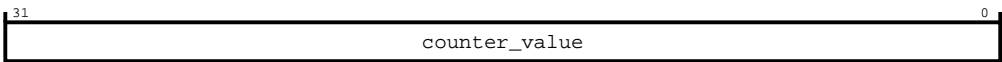
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr7 register bit assignments.

Figure 16-60: Bit assignment diagram for the pmevcntr7 register



The following table shows the pmevcntr7 register bit descriptions.

Table 16-69: pmevcntr7 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The recorded number of program-specified events observed by performance monitor 7 that have occurred in the clock domain within a programmed period. An event can fire no more than one time in each cycle.	RW	0x0

16.9.11 PMU pmccntr\_l register

This register contains the value of lower 64-bit cycle counter bits[31:0].

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x000000F8

Type

RW

Reset value

0x00000000

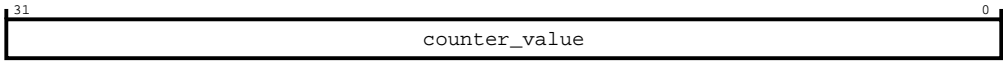
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmccntr\_l register bit assignments.

Figure 16-61: Bit assignment diagram for the pmccntr\_l register



The following table shows the pmccntr\_l register bit descriptions.

Table 16-70: pmccntr\_l bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The value of lower 64-bit cycle counter bits[31:0]	RW	0x0

16.9.12 PMU pmccntr\_u register

This register contains the value of upper 64-bit cycle counter bits[63:32].

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x000000FC

Type

RW

Reset value

0x00000000

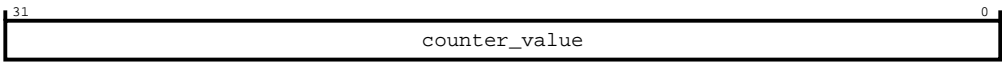
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmccntr\_u register bit assignments.

Figure 16-62: Bit assignment diagram for the pmccntr\_u register



The following table shows the pmccntr\_u register bit descriptions.

Table 16-71: pmccntr\_u bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The value of upper 64-bit cycle counter bits[63:32]	RW	0x0

16.9.13 PMU pmevtyper0 register

Performance monitor event type filter 0. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000400

Type

RW

Reset value

0x00000000

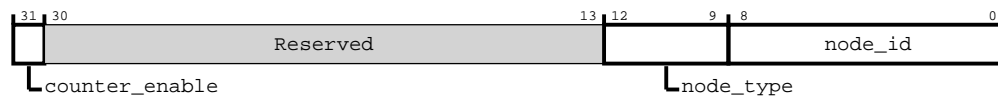
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevtyper0 register bit assignments.

**Figure 16-63: Bit assignment diagram for the pmevtyper0 register**



The following table shows the pmevtyper0 register bit descriptions.

**Table 16-72: pmevtyper0 bit descriptions**

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable:  0 Trigger snapshot capture on overflow disabled 1 Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

### 16.9.14 PMU pmevtyper1 register

Performance monitor event type filter 1. Registers PMEVTYPEPER0-7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x00000404

##### Type

RW

##### Reset value

0x00000000

#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevtyper1 register bit assignments.

Figure 16-64: Bit assignment diagram for the pmevtyper1 register



The following table shows the pmevtyper1 register bit descriptions.

Table 16-73: pmevtyper1 bit descriptions

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable: 0 Trigger snapshot capture on overflow disabled 1 Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

16.9.15 PMU pmevtyper2 register

Performance monitor event type filter 2. Registers PMEVTYPER0-7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000408

Type

RW

Reset value

0x00000000

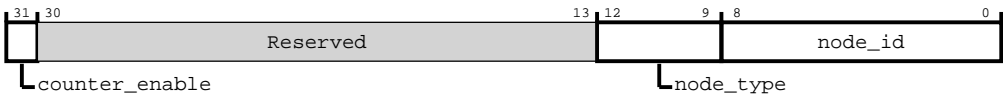
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevtyper2 register bit assignments.

Figure 16-65: Bit assignment diagram for the pmevtyper2 register



The following table shows the pmevtyper2 register bit descriptions.

Table 16-74: pmevtyper2 bit descriptions

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable:  0 Trigger snapshot capture on overflow disabled  1 Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

16.9.16 PMU pmevtyper3 register

Performance monitor event type filter 3. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, event type, and type and ID of the target node.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x0000040C

**Type**  
RW

**Reset value**  
0x00000000

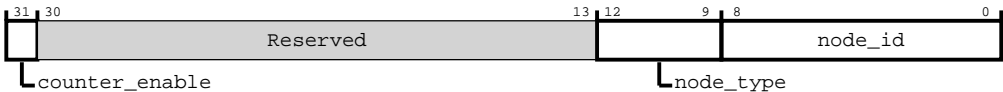
**Constraints**

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the pmevtyper3 register bit assignments.

**Figure 16-66: Bit assignment diagram for the pmevtyper3 register**



The following table shows the pmevtyper3 register bit descriptions.

**Table 16-75: pmevtyper3 bit descriptions**

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable:  0 Trigger snapshot capture on overflow disabled  1 Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

16.9.17 PMU pmevtyper4 register

Performance monitor event type filter 4. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:



**Width**  
32-bit

**Address offset**  
0x00000410

**Type**  
RW

**Reset value**  
0x00000000

**Constraints**

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the pmevtyper4 register bit assignments.

**Figure 16-67: Bit assignment diagram for the pmevtyper4 register**



The following table shows the pmevtyper4 register bit descriptions.

**Table 16-76: pmevtyper4 bit descriptions**

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable:  0 Trigger snapshot capture on overflow disabled 1 Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

16.9.18 PMU pmevtyper5 register

Performance monitor event type filter 5. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000414

Type

RW

Reset value

0x00000000

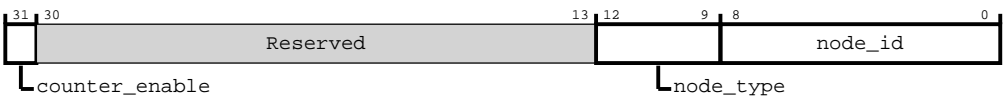
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevtyper5 register bit assignments.

Figure 16-68: Bit assignment diagram for the pmevtyper5 register



The following table shows the pmevtyper5 register bit descriptions.

Table 16-77: pmevtyper5 bit descriptions

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable:  0 Trigger snapshot capture on overflow disabled  1 Trigger snapshot capture on overflow enabled	RW	0



**Table 16-78: pmevtyper6 bit descriptions**

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable: <b>0</b> Trigger snapshot capture on overflow disabled <b>1</b> Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

## 16.9.20 PMU pmevtyper7 register

Performance monitor event type filter 7. Registers pmevtyper0-pmevtyper7 control the performance monitor event counter start and stop period, eventtype, and type and ID of the target node.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x0000041C

#### Type

RW

#### Reset value

0x00000000

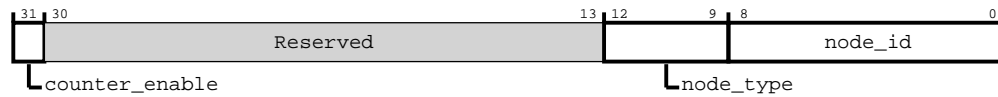
### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the pmevtyper7 register bit assignments.

**Figure 16-70: Bit assignment diagram for the pmevtyper7 register**



The following table shows the pmevtyper7 register bit descriptions.

**Table 16-79: pmevtyper7 bit descriptions**

Bits	Name	Description	Type	Reset
[31]	counter_enable	Counter enable: <b>0</b> Trigger snapshot capture on overflow disabled <b>1</b> Trigger snapshot capture on overflow enabled	RW	0
[30:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12:9]	node_type	The node type	RW	0b0000
[8:0]	node_id	The Node ID	RW	0x0

### 16.9.21 PMU pmsr register

This register records the status of a performance event counter when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x610

##### Type

RO

##### Reset value

0x00000001

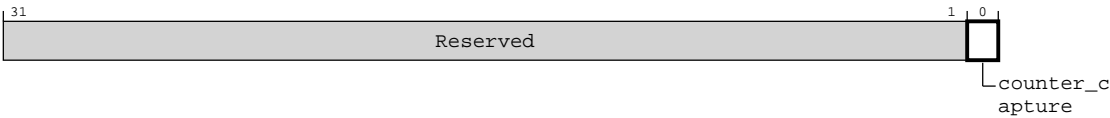
#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmssr register bit assignments.

Figure 16-71: Bit assignment diagram for the pmssr register



The following table shows the pmssr register bit descriptions.

Table 16-80: pmssr bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[0]	counter_capture	No capture. Indicates whether the PMU counters have been captured. The values are:  0 PMU counters are captured.  1 PMU counters are not captured.	RO	1

16.9.22 PMU pmovssr register

This register records the overflow status of a performance event counter when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x614

Type

RO

Reset value

0x00000000

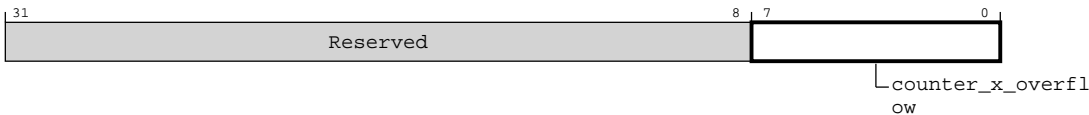
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmovssr register bit assignments.

Figure 16-72: Bit assignment diagram for the pmovssr register



The following table shows the pmovssr register bit descriptions.

Table 16-81: pmovssr bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	counter_x_overflow	Counter overflow status	RO	0x0

16.9.23 PMU pmccntr\_l register

This register contains the snapshot value of the lower 64-bit cycle counter bits[31:0].

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000618

Type

RO

Reset value

0x00000000

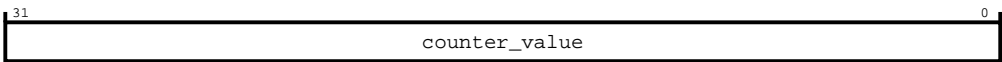
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmccntr\_l register bit assignments.

Figure 16-73: Bit assignment diagram for the pmccntr\_l register



The following table shows the pmccntr\_l register bit descriptions.

Table 16-82: pmccntr\_l bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The snapshot value of the lower 64-bit cycle counter bits[31:0]	RO	0x0

16.9.24 PMU pmccntr\_u register

This register contains the snapshot value of the upper 64-bit cycle counter bits[63:32].

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x0000061C

Type

RO

Reset value

0x00000000

Constraints

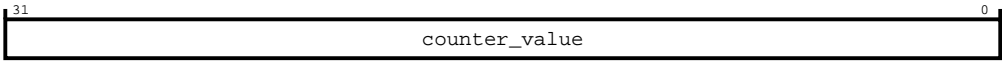
Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.



Bit descriptions

The following figure shows the pmccntsr\_u register bit assignments.

Figure 16-74: Bit assignment diagram for the pmccntsr\_u register



The following table shows the pmccntsr\_u register bit descriptions.

Table 16-83: pmccntsr\_u bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	The snapshot value of the upper 64-bit cycle counter bits[63:32]	RO	0x0

16.9.25 PMU pmevcntsr0 register

Performance monitor 0 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000620

Type

RO

Reset value

0x00000000

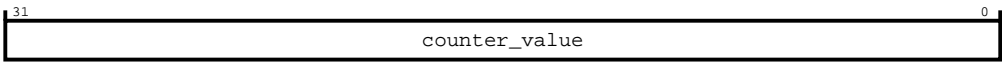
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntsr0 register bit assignments.

Figure 16-75: Bit assignment diagram for the pmevcntrs0 register



The following table shows the pmevcntrs0 register bit descriptions.

Table 16-84: pmevcntrs0 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 0	RO	0x0

16.9.26 PMU pmevcntrs1 register

Performance monitor 1 event counter snapshot register. pmevcntrs0-pmevcntrs7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000624

Type

RO

Reset value

0x00000000

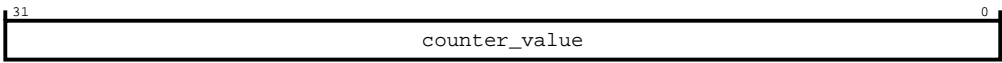
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntrs1 register bit assignments.

Figure 16-76: Bit assignment diagram for the pmevcntr1 register



The following table shows the pmevcntr1 register bit descriptions.

Table 16-85: pmevcntr1 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 1	RO	0x0

16.9.27 PMU pmevcntr2 register

Performance monitor 2 event counter snapshot register. pmevcntr0-pmevcntr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000628

Type

RO

Reset value

0x00000000

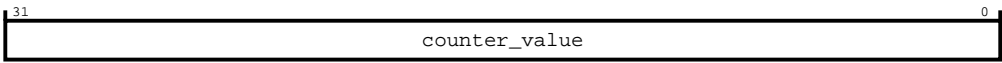
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr2 register bit assignments.

Figure 16-77: Bit assignment diagram for the pmevcntsr2 register



The following table shows the pmevcntsr2 register bit descriptions.

Table 16-86: pmevcntsr2 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 2	RO	0x0

16.9.28 PMU pmevcntsr3 register

Performance monitor 3 event counter snapshot register. pmevcntsr0-pmevcntsr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x0000062C

Type

RO

Reset value

0x00000000

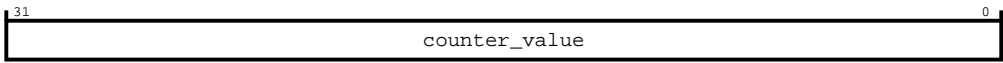
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntsr3 register bit assignments.

Figure 16-78: Bit assignment diagram for the pmevcntr3 register



The following table shows the pmevcntr3 register bit descriptions.

Table 16-87: pmevcntr3 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 3	RO	0x0

16.9.29 PMU pmevcntr4 register

Performance monitor 4 event counter snapshot register. pmevcntr0-pmevcntr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000630

Type

RO

Reset value

0x00000000

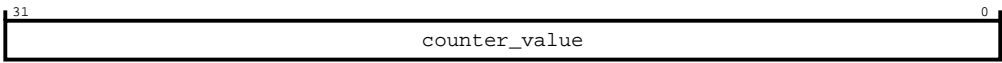
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr4 register bit assignments.

Figure 16-79: Bit assignment diagram for the pmevcntrs4 register



The following table shows the pmevcntrs4 register bit descriptions.

Table 16-88: pmevcntrs4 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 4	RO	0x0

16.9.30 PMU pmevcntrs5 register

Performance monitor 5 event counter snapshot register. pmevcntrs0-pmevcntrs7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000634

Type

RO

Reset value

0x00000000

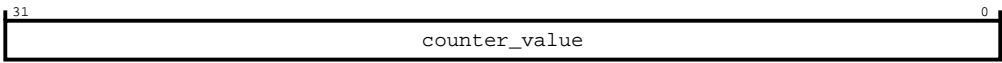
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntrs5 register bit assignments.

Figure 16-80: Bit assignment diagram for the pmevcntr5 register



The following table shows the pmevcntr5 register bit descriptions.

Table 16-89: pmevcntr5 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 5	RO	0x0

16.9.31 PMU pmevcntr6 register

Performance monitor 6 event counter snapshot register. pmevcntr0-pmevcntr7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00000638

Type

RO

Reset value

0x00000000

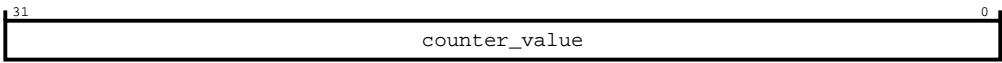
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntr6 register bit assignments.

Figure 16-81: Bit assignment diagram for the pmevcntrs6 register



The following table shows the pmevcntrs6 register bit descriptions.

Table 16-90: pmevcntrs6 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 6	RO	0x0

16.9.32 PMU pmevcntrs7 register

Performance monitor 7 event counter snapshot register. pmevcntrs0-pmevcntrs7 are shadow registers that record an Event counter n snapshot value of the performance event counters when enabled by the <CLKNAME>\_PMUSNAPSHOTREQ input signal.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x0000063C

Type

RO

Reset value

0x00000000

Constraints

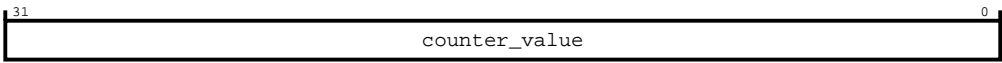
Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmevcntrs7 register bit assignments.



Figure 16-82: Bit assignment diagram for the pmevcntsr7 register



The following table shows the pmevcntsr7 register bit descriptions.

Table 16-91: pmevcntsr7 bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	counter_value	Contains the snapshot of the number of events observed by performance monitor 7	RO	0x0

16.9.33 PMU pmsscr register

This register captures a snapshot of the performance monitors.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x6F0

Type

RW

Reset value

0x00000000

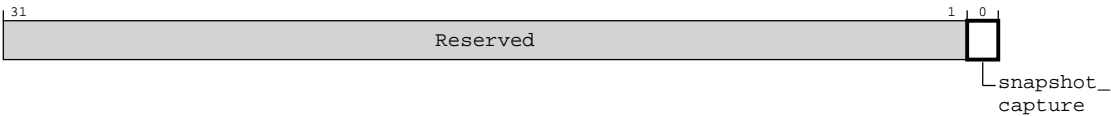
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmsscr register bit assignments.

**Figure 16-83: Bit assignment diagram for the pmsscr register**



The following table shows the pmsscr register bit descriptions.

**Table 16-92: pmsscr bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[0]	snapshot_capture	Initiates snapshot of PMU register bank	WO	0

16.9.34 PMU pmcntenset register

This register sets the performance monitors count enable.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC00

Type

RW

Reset value

0x00000000

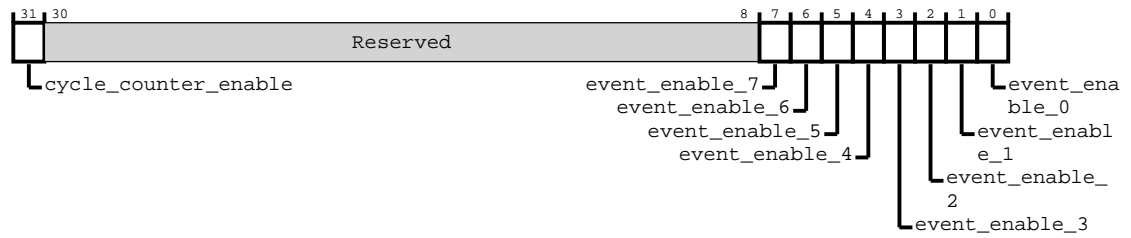
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmcntenset register bit assignments.

**Figure 16-84: Bit assignment diagram for the pmcntenset register**



The following table shows the pmcntenset register bit descriptions.

**Table 16-93: pmcntenset bit descriptions**

Bits	Name	Description	Type	Reset
[31]	cycle_counter_enable	<p>The pmccntr enable bit. Enables the cycle counter register. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the cycle counter is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the cycle counter is enabled. Writing this value enables the cycle counter.</p> <p>Write 1 to set.</p>	RW	0
[30:8]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[7]	event_enable_7	<p>The event counter enable bit for pmevcntr7. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr7 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr7 event counter is enabled. Writing this value enables pmevcntr7.</p> <p>Write 1 to set.</p>	RW	0
[6]	event_enable_6	<p>The event counter enable bit for pmevcntr6. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr6 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr6 event counter is enabled. Writing this value enables pmevcntr6.</p> <p>Write 1 to set.</p>	RW	0

Bits	Name	Description	Type	Reset
[5]	event_enable_5	<p>The event counter enable bit for pmevcntr5. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr5 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr5 event counter is enabled. Writing this value enables pmevcntr5.</p> <p>Write 1 to set.</p>	RW	0
[4]	event_enable_4	<p>The event counter enable bit for pmevcntr4. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr4 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr4 event counter is enabled. Writing this value enables pmevcntr4.</p> <p>Write 1 to set.</p>	RW	0
[3]	event_enable_3	<p>The event counter enable bit for pmevcntr3. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr3 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr3 event counter is enabled. Writing this value enables pmevcntr3.</p> <p>Write 1 to set.</p>	RW	0
[2]	event_enable_2	<p>The event counter enable bit for pmevcntr2. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr2 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr2 event counter is enabled. Writing this value enables pmevcntr2.</p> <p>Write 1 to set.</p>	RW	0
[1]	event_enable_1	<p>The event counter enable bit for pmevcntr1. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr1 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr1 event counter is enabled. Writing this value enables pmevcntr1.</p> <p>Write 1 to set.</p>	RW	0

Bits	Name	Description	Type	Reset
[0]	event_enable_0	<p>The event counter enable bit for PMEVCNTR0. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr0 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr0 event counter is enabled. Writing this value enables pmevcntr0.</p> <p>Write 1 to set.</p>	RW	0

### 16.9.35 PMU pmcntencr register

This register clears the performance monitors count enable.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xC20

##### Type

RW

##### Reset value

0x00000000

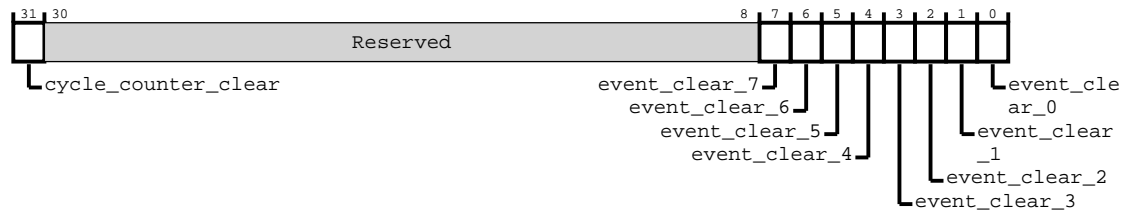
#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the pmcntencr register bit assignments.

**Figure 16-85: Bit assignment diagram for the pmcntencr register**



The following table shows the pmcntencr register bit descriptions.

**Table 16-94: pmcntencr bit descriptions**

Bits	Name	Description	Type	Reset
[31]	cycle_counter_clear	The pmcncr disable bit. Disables the cycle counter register. The values are: <b>0</b> When read, indicates that the cycle counter is disabled. Writing this value has no effect. <b>1</b> When read, indicates that the cycle counter is enabled. Writing this value disables the cycle counter. Write 1 to clear.	RW	0
[30:8]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[7]	event_clear_7	The Event counter disable bit for pmevncnr7. The values are: <b>0</b> When read, indicates that pmevncnr7 is disabled. Writing this value has no effect. <b>1</b> When read, indicates that pmevncnr7 is enabled. Writing this value disables pmevncnr7. Write 1 to clear.	RW	0
[6]	event_clear_6	The Event counter disable bit for pmevncnr6. The values are: <b>0</b> When read, indicates that pmevncnr6 is disabled. Writing this value has no effect. <b>1</b> When read, indicates that pmevncnr6 is enabled. Writing this value disables pmevncnr6. Write 1 to clear.	RW	0
[5]	event_clear_5	The Event counter disable bit for pmevncnr5. The values are: <b>0</b> When read, indicates that pmevncnr5 is disabled. Writing this value has no effect. <b>1</b> When read, indicates that pmevncnr5 is enabled. Writing this value disables pmevncnr5. Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[4]	event_clear_4	<p>The Event counter disable bit for pmevcntr4. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr4 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr4 is enabled. Writing this value disables pmevcntr4.</p> <p>Write 1 to clear.</p>	RW	0
[3]	event_clear_3	<p>The Event counter disable bit for pmevcntr3. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr3 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr3 is enabled. Writing this value disables pmevcntr3.</p> <p>Write 1 to clear.</p>	RW	0
[2]	event_clear_2	<p>The Event counter disable bit for pmevcntr2. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr2 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr2 is enabled. Writing this value disables pmevcntr2.</p> <p>Write 1 to clear.</p>	RW	0
[1]	event_clear_1	<p>The Event counter disable bit for pmevcntr1. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr1 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr1 is enabled. Writing this value disables pmevcntr1.</p> <p>Write 1 to clear.</p>	RW	0
[0]	event_clear_0	<p>The Event counter disable bit for pmevcntr0. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr0 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr0 is enabled. Writing this value disables pmevcntr0.</p> <p>Write 1 to clear.</p>	RW	0

### 16.9.36 PMU pmintenset register

This register sets the performance monitors interrupt enable.

#### Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC40

Type

RW

Reset value

0x00000000

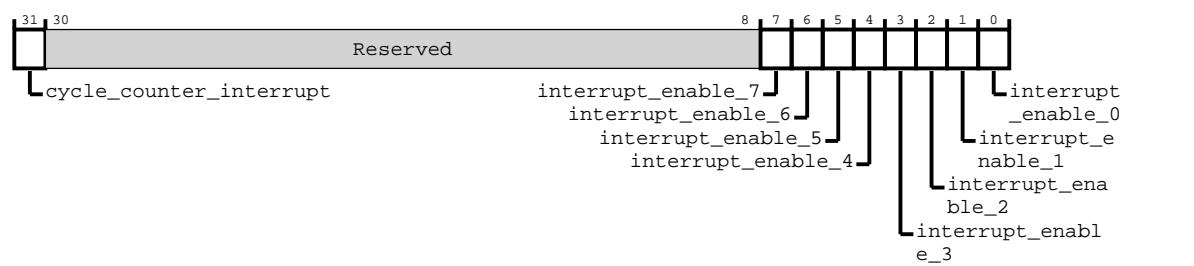
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the pmintenset register bit assignments.

Figure 16-86: Bit assignment diagram for the pmintenset register



The following table shows the pmintenset register bit descriptions.

Table 16-95: pmintenset bit descriptions

Bits	Name	Description	Type	Reset
[31]	cycle_counter_interrupt	The PMCCNTR overflow interrupt request enable bit. The values are: 0 When read, means that the cycle counter overflow interrupt request is disabled. When written, has no effect. 1 When read, means that the cycle counter overflow interrupt request is enabled. When written, enables the cycle count overflow interrupt request  Write 1 to set.	RW	0
[30:8]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0



Bits	Name	Description	Type	Reset
[7]	interrupt_enable_7	<p>Event counter overflow interrupt request enable bit for pmevcntr7. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr7_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr7_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr7_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0
[6]	interrupt_enable_6	<p>Event counter overflow interrupt request enable bit for pmevcntr6. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr6_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr6_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr6_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0
[5]	interrupt_enable_5	<p>Event counter overflow interrupt request enable bit for pmevcntr5. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr5_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr5_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr5_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0
[4]	interrupt_enable_4	<p>Event counter overflow interrupt request enable bit for pmevcntr4. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr4_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr4_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr4_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0
[3]	interrupt_enable_3	<p>Event counter overflow interrupt request enable bit for pmevcntr3. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr3_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr3_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr3_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0

Bits	Name	Description	Type	Reset
[2]	interrupt_enable_2	<p>Event counter overflow interrupt request enable bit for pmevcntr2. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr2_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr2_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr2_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0
[1]	interrupt_enable_1	<p>Event counter overflow interrupt request enable bit for pmevcntr1. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr1_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr1_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr1_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0
[0]	interrupt_enable_0	<p>Event counter overflow interrupt request enable bit for pmevcntr0. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr0_el0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr0_el0 event counter interrupt request is enabled. Writing this value enables the pmevcntr0_el0 interrupt request.</p> <p>Write 1 to set.</p>	RW	0

### 16.9.37 PMU pmintencnr register

This register clears the performance monitors interrupt enable.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xC60

##### Type

RW

## Reset value

0x00000000

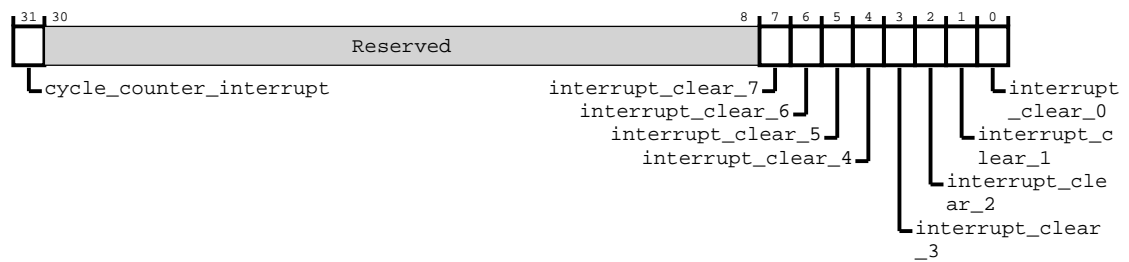
## Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmintenclr register bit assignments.

**Figure 16-87: Bit assignment diagram for the pmintenclr register**



The following table shows the pmintenclr register bit descriptions.

**Table 16-96: pmintenclr bit descriptions**

Bits	Name	Description	Type	Reset
[31]	cycle_counter_interrupt	The pmccntr overflow interrupt request disable bit. The values are:  <b>0</b>  When read, indicates that the cycle counter overflow interrupt request is disabled. Writing this value has no effect.  <b>1</b>  When read, indicates that the cycle counter overflow interrupt request is enabled. Writing this value disables the cycle count overflow interrupt request.  Write 1 to clear.	RW	0
[30:8]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[7]	interrupt_clear_7	The event counter overflow interrupt request disable bit for pmevcntr7. The values are:  <b>0</b>  When read, indicates that the pmevcntr7 event counter interrupt request is disabled. Writing this value has no effect.  <b>1</b>  When read, indicates that the pmevcntr7 event counter interrupt request is enabled. Writing this value disables the pmevcntr7 interrupt request.  Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[6]	interrupt_clear_6	<p>The event counter overflow interrupt request disable bit for pmevcntr6. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr6 is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr6 event counter interrupt request is enabled. Writing this value disables the pmevcntr6 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0
[5]	interrupt_clear_5	<p>The event counter overflow interrupt request disable bit for pmevcntr5. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr5 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr5 event counter interrupt request is enabled. Writing this value disables the pmevcntr5 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0
[4]	interrupt_clear_4	<p>The event counter overflow interrupt request disable bit for pmevcntr4. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr4 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr4 event counter interrupt request is enabled. Writing this value disables the pmevcntr4 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0
[3]	interrupt_clear_3	<p>The event counter overflow interrupt request disable bit for pmevcntr3. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr3 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr3 event counter interrupt request is enabled. Writing this value disables the pmevcntr3 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0
[2]	interrupt_clear_2	<p>The event counter overflow interrupt request disable bit for pmevcntr2. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr2 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr2 event counter interrupt request is enabled. Writing this value disables the pmevcntr2 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0

Bits	Name	Description	Type	Reset
[1]	interrupt_clear_1	<p>The event counter overflow interrupt request disable bit for pmevcntr1. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr1 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr1 event counter interrupt request is enabled. Writing this value disables the pmevcntr1 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0
[0]	interrupt_clear_0	<p>The event counter overflow interrupt request disable bit for pmevcntr0. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the pmevcntr0 event counter interrupt request is disabled. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the pmevcntr0 event counter interrupt request is enabled. Writing this value disables the pmevcntr0 interrupt request.</p> <p>Write 1 to clear.</p>	RW	0

## 16.9.38 PMU pmovalr register

This register clears the performance monitors overflow flag status.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0xC80

#### Type

RW

#### Reset value

0x00000000

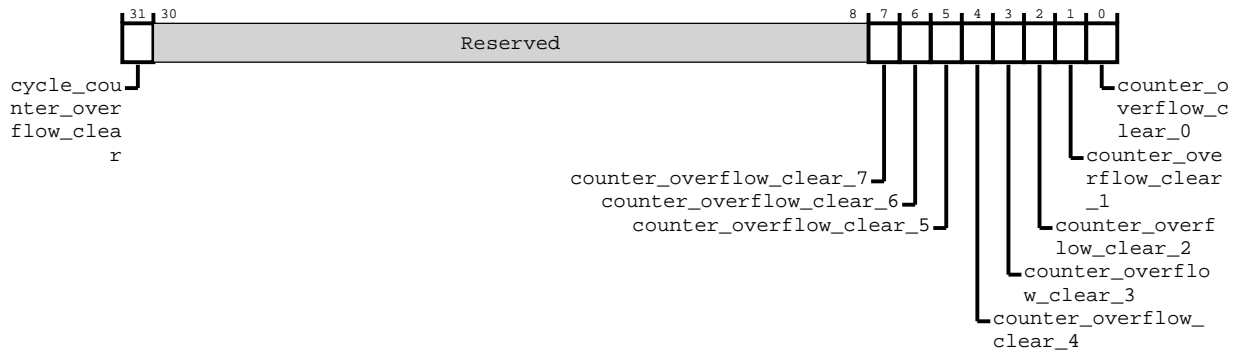
### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmovsclr register bit assignments.

**Figure 16-88: Bit assignment diagram for the pmovsclr register**



The following table shows the pmovsclr register bit descriptions.

**Table 16-97: pmovsclr bit descriptions**

Bits	Name	Description	Type	Reset
[31]	cycle_counter_overflow_clear	<p>The pmcncr cycle counter overflow bit. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the cycle counter has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the cycle counter has overflowed. Writing this value clears the overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0
[30:8]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[7]	counter_overflow_clear_7	<p>The event counter overflow clear bit for pmevcntr7. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr7 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr7 has overflowed. Writing this value clears the pmevcntr7 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0

Bits	Name	Description	Type	Reset
[6]	counter_overflow_clear_6	<p>The event counter overflow clear bit for pmevcntr6. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr6 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr6 has overflowed. Writing this value clears the pmevcntr6 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0
[5]	counter_overflow_clear_5	<p>The event counter overflow clear bit for pmevcntr5. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr5 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr5 has overflowed. Writing this value clears the pmevcntr5 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0
[4]	counter_overflow_clear_4	<p>The event counter overflow clear bit for pmevcntr4. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr4 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr4 has overflowed. Writing this value clears the pmevcntr4 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0
[3]	counter_overflow_clear_3	<p>The event counter overflow clear bit for pmevcntr3. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr3 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr3 has overflowed. Writing this value clears the pmevcntr3 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0
[2]	counter_overflow_clear_2	<p>The event counter overflow clear bit for pmevcntr2. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr2 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr2 has overflowed. Writing this value clears the pmevcntr2 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0

Bits	Name	Description	Type	Reset
[1]	counter_overflow_clear_1	<p>The event counter overflow clear bit for pmevcntr1. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr1 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr1 has overflowed. Writing this value clears the pmevcntr1 overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0
[0]	counter_overflow_clear_0	<p>The event counter overflow clear bit for pmevcntr0. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr has overflowed. Writing this value clears the pmevcntr overflow bit to 0.</p> <p>Write 1 to clear.</p>	RW	0

### 16.9.39 PMU pmovsset register

This register sets the performance monitors overflow flag status.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xCC0

##### Type

RW

##### Reset value

0x00000000

#### Constraints

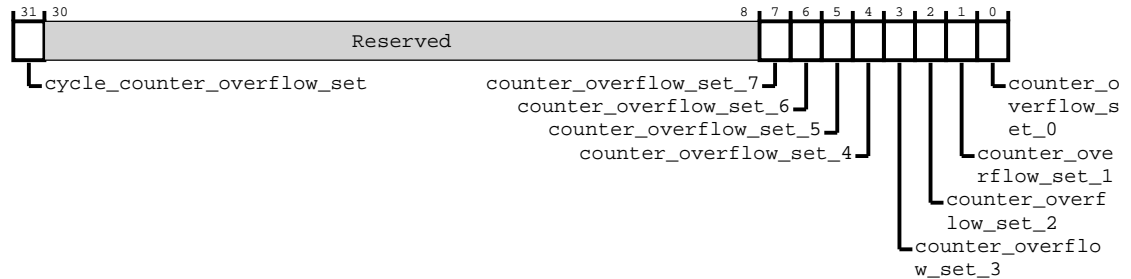
Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.



## Bit descriptions

The following figure shows the pmovsset register bit assignments.

**Figure 16-89: Bit assignment diagram for the pmovsset register**



The following table shows the pmovsset register bit descriptions.

**Table 16-98: pmovsset bit descriptions**

Bits	Name	Description	Type	Reset
[31]	cycle_counter_overflow_set	<p>The pmccntr cycle counter overflow bit. The values are:</p> <p><b>0</b></p> <p>When read, indicates that the cycle counter has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that the cycle counter has overflowed. Writing this value sets the overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0
[30:8]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[7]	counter_overflow_set_7	<p>The event counter overflow set bit for pmevcntr7. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr7 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr7 has overflowed. Writing this value sets the pmevcntr7 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0
[6]	counter_overflow_set_6	<p>The event counter overflow set bit for pmevcntr6. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr6 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr6 has overflowed. Writing this value sets the pmevcntr6 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0

Bits	Name	Description	Type	Reset
[5]	counter_overflow_set_5	<p>The event counter overflow set bit for pmevcntr5. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr5 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr5 has overflowed. Writing this value sets the pmevcntr5 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0
[4]	counter_overflow_set_4	<p>The event counter overflow set bit for pmevcntr4. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr4 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr4 has overflowed. Writing this value sets the pmevcntr4 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0
[3]	counter_overflow_set_3	<p>The event counter overflow set bit for pmevcntr3. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr3 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr3 has overflowed. Writing this value sets the pmevcntr3 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0
[2]	counter_overflow_set_2	<p>The event counter overflow set bit for pmevcntr2. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr2 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr2 has overflowed. Writing this value sets the pmevcntr2 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0
[1]	counter_overflow_set_1	<p>The event counter overflow set bit for pmevcntr1. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr1 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr1 has overflowed. Writing this value sets the pmevcntr1 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0

Bits	Name	Description	Type	Reset
[0]	counter_overflow_set_0	<p>The event counter overflow set bit for pmevcntr0. The values are:</p> <p><b>0</b></p> <p>When read, indicates that pmevcntr0 has not overflowed. Writing this value has no effect.</p> <p><b>1</b></p> <p>When read, indicates that pmevcntr0 has overflowed. Writing this value sets the pmevcntr0 overflow bit to 1.</p> <p>Write 1 to set.</p>	RW	0

## 16.9.40 PMU pmcccgr register

This register controls the cycle counter clock gating enable.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0xD80

#### Type

RW

#### Reset value

0x00000000

### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the pmcccgr register bit assignments.

**Figure 16-90: Bit assignment diagram for the pmcccgr register**



The following table shows the pmcccgr register bit descriptions.

**Table 16-99: pmcccgr bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[0]	clock_gate	<p>Defines whether to drive or gate the QACTIVE signal.</p> <p><b>0</b></p> <p>Gate the QACTIVE signal for the clock domain when no events are present.</p> <p><b>1</b></p> <p>Drive the QACTIVE signal for the clock domain when no events are present.</p>	RW	0

### 16.9.41 PMU pmcfgr register

This register contains configuration values for the performance monitors.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xE00

##### Type

RO

##### Reset value

0x00417F08

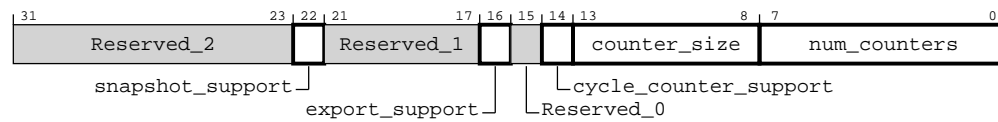
#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the pmcfgr register bit assignments.

**Figure 16-91: Bit assignment diagram for the pmcfgr register**



The following table shows the pmcfgr register bit descriptions.

**Table 16-100: pmcfgr bit descriptions**

Bits	Name	Description	Type	Reset
[31:23]	Reserved_2	Bits within this register segment are reserved for future product development.	RO	0x0
[22]	snapshot_support	Always 1	RO	1
[21:17]	Reserved_1	Bits within this register segment are reserved for future product development.	RO	0b00000
[16]	export_support	Always 1	RO	1
[15]	Reserved_0	Bits within this register segment are reserved for future product development.	RO	0
[14]	cycle_counter_support	Always 1	RO	1
[13:8]	counter_size	Always 0b111111 (SIZE)	RO	0b111111
[7:0]	num_counters	Always 0b00001000 (8 counters)	RO	0x8

## 16.9.42 PMU pmcr register

This register controls the performance monitors.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0xE04

#### Type

RW

#### Reset value

0x00000000

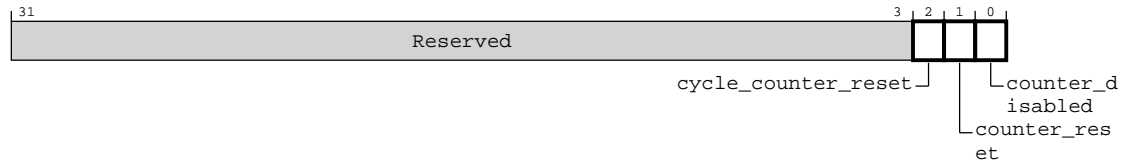
### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmcr register bit assignments.

**Figure 16-92: Bit assignment diagram for the pmcr register**



The following table shows the pmcr register bit descriptions.

**Table 16-101: pmcr bit descriptions**

Bits	Name	Description	Type	Reset
[31:3]	Reserved	Bits within this register segment are reserved for future product development.	RO	0x0
[2]	cycle_counter_reset	Reset cycle counter, excluding overflow, Read-As-Zero	WO	0
[1]	counter_reset	Reset event counters, excluding overflows, Read-As-Zero	WO	0
[0]	counter_disabled	Enable all counters using the PMCNTENSET register, event and cycle, or disable all counters.	RW	0

## 16.10 APU register summary

This section describes the APU registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-102: APU register summary**

Offset	Name	Type	Reset	Width	Description
0x00	PRBAR_LOW	RW	0x00000000	32-bit	Region Base Address Register (PRBAR) (lower 32 bits) contains the fields for the lower bound address and access attributes of the region.
0x04	PRBAR_HIGH	RW	0x00000000	32-bit	Region Base Address Register (PRBAR) (higher 32 bits) contains the fields for the lower bound address.
0x08	PRLAR_LOW	RW	0x00000000	32-bit	Region Limit Address Register (PRLAR) contains the fields for upper bound address and enables or disables control for the region.
0x0C	PRLAR_HIGH	RW	0x00000000	32-bit	Region Limit Address Register (PRLAR) contains the fields for upper bound address and enables or disables control for the region.
0x10	PRID_LOW	RW	0x00000000	32-bit	Contains the entity IDs assigned to this memory region and their respective permissions.
0x14	PRID_HIGH	RW	0x00000000	32-bit	Contains the entity IDs assigned to this memory region and their respective permissions.
0xFF8	APU_CTLR	RW	See individual bit resets.	32-bit	APU control register.
0xFFC	APU_IIDR	RO	0x00000001	32-bit	Contains information about the APU implementation.

16.10.1 APU PRBAR\_LOW register

Region Base Address Register (PRBAR) (lower 32 bits) contains the fields for the lower bound address and access attributes of the region.

Configurations

This register is only present if the APU is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x00

Type

RW

Reset value

0x00000000

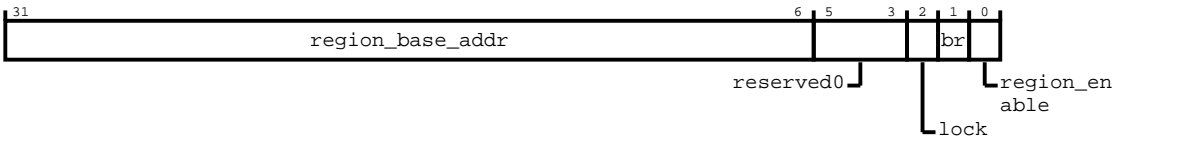
Constraints

None.

Bit descriptions

The following figure shows the PRBAR\_LOW register bit assignments.

Figure 16-93: Bit assignment diagram for the PRBAR\_LOW register



The following table shows the PRBAR\_LOW register bit descriptions.

Table 16-103: PRBAR\_LOW bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	region_base_addr	Bits [31:6] of base address of the range.	RW	0x0
[5:3]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[2]	lock	Address region is locked. Once locked, cannot be unlocked until reset	RW	0
[1]	br	Address region is background region	RW	0
[0]	region_enable	Address region is enabled	RW	0

16.10.2 APU PRBAR\_HIGH register

Region Base Address Register (PRBAR) (higher 32 bits) contains the fields for the lower bound address.

Configurations

This register is only present if the APU is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RW

Reset value

0x00000000

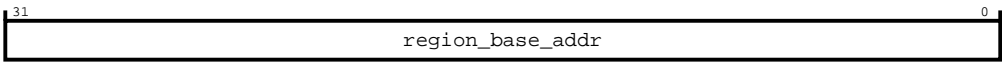
Constraints

None.

Bit descriptions

The following figure shows the PRBAR\_HIGH register bit assignments.

Figure 16-94: Bit assignment diagram for the PRBAR\_HIGH register



The following table shows the PRBAR\_HIGH register bit descriptions.

Table 16-104: PRBAR\_HIGH bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	region_base_addr	Bits [47:6] of base address of the range. Note: Unused upper address bits are RAZ/WI	RW	0x0



### 16.10.3 APU PRLAR\_LOW register

Region Limit Address Register (PRLAR) contains the fields for upper bound address and enables or disables control for the region.

#### Configurations

This register is only present if the APU is enabled on the interface.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x08

##### Type

RW

##### Reset value

0x00000000

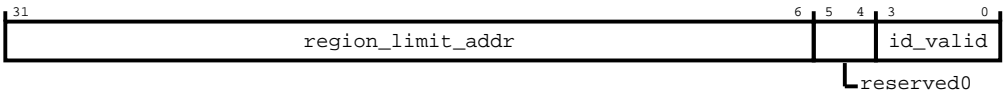
#### Constraints

None.

#### Bit descriptions

The following figure shows the PRLAR\_LOW register bit assignments.

**Figure 16-95: Bit assignment diagram for the PRLAR\_LOW register**



The following table shows the PRLAR\_LOW register bit descriptions.

**Table 16-105: PRLAR\_LOW bit descriptions**

Bits	Name	Description	Type	Reset
[31:6]	region_limit_addr	Bits [47:6] of limit address of the range	RW	0x0
[5:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[3:0]	id_valid	Valid bits for IDs in PRID register	RW	0b0000

16.10.4 APU PRLAR\_HIGH register

Region Limit Address Register (PRLAR) contains the fields for upper bound address and enables or disables control for the region.

Configurations

This register is only present if the APU is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x0C

Type

RW

Reset value

0x00000000

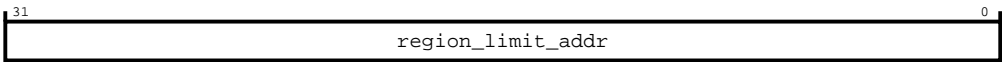
Constraints

None.

Bit descriptions

The following figure shows the PRLAR\_HIGH register bit assignments.

Figure 16-96: Bit assignment diagram for the PRLAR\_HIGH register



The following table shows the PRLAR\_HIGH register bit descriptions.

Table 16-106: PRLAR\_HIGH bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	region_limit_addr	Bits [47:6] of limit address of the range. Note: Unused upper address bits are RAZ/WI	RW	0x0

16.10.5 APU PRID\_LOW register

Contains the entity IDs assigned to this memory region and their respective permissions.

Configurations

This register is only present if the APU is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x10

Type

RW

Reset value

0x00000000

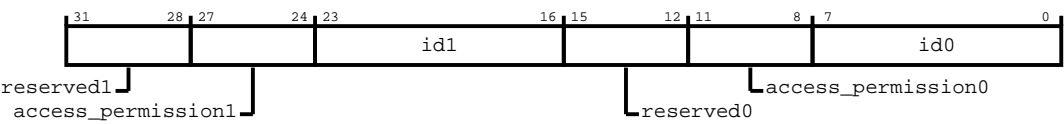
Constraints

None.

Bit descriptions

The following figure shows the PRID\_LOW register bit assignments.

Figure 16-97: Bit assignment diagram for the PRID\_LOW register



The following table shows the PRID\_LOW register bit descriptions.

Table 16-107: PRID\_LOW bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	reserved1	Bits within this register segment are reserved for future product development	RO	0b0000
[27:24]	access_permission1	Access permission for entity1 0: non-secure write 1: secure write 2: non-secure read 3: secure read 4-7: reserved	RW	0b0000
[23:16]	id1	Id of entity1	RW	0x0
[15:12]	reserved0	Bits within this register segment are reserved for future product development	RO	0b0000
[11:8]	access_permission0	Access permission for entity0 0: non-secure write 1: secure write 2: non-secure read 3: secure read 4-7: reserved	RW	0b0000
[7:0]	id0	Id of entity0	RW	0x0

16.10.6 APU PRID\_HIGH register

Contains the entity IDs assigned to this memory region and their respective permissions.

Configurations

This register is only present if the APU is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x14

Type

RW

Reset value

0x00000000

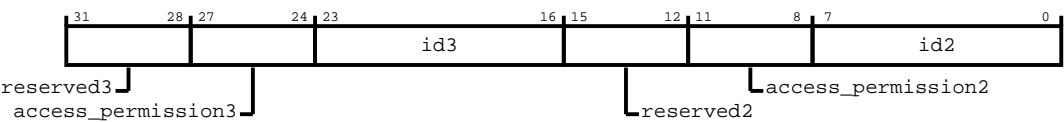
Constraints

None.

Bit descriptions

The following figure shows the PRID\_HIGH register bit assignments.

Figure 16-98: Bit assignment diagram for the PRID\_HIGH register



The following table shows the PRID\_HIGH register bit descriptions.

Table 16-108: PRID\_HIGH bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	reserved3	Bits within this register segment are reserved for future product development	RO	0b0000
[27:24]	access_permission3	Access permission for entity3	RW	0b0000
[23:16]	id3	Id of entity3	RW	0x0
[15:12]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000
[11:8]	access_permission2	Access permission for entity2 0: non-secure write 1: secure write 2: non-secure read 3: secure read 4-7: reserved	RW	0b0000
[7:0]	id2	Id of entity2	RW	0x0

16.10.7 APU APU\_CTLR register

APU control register.

Configurations

This register is only present if the APU is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFF8

Type

RW

Reset value

See individual bit resets.

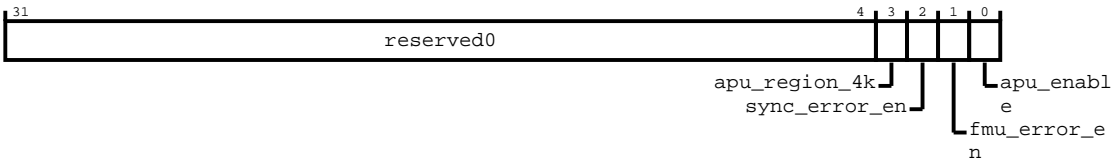
Constraints

None.

Bit descriptions

The following figure shows the APU\_CTLR register bit assignments.

Figure 16-99: Bit assignment diagram for the APU\_CTLR register



The following table shows the APU\_CTLR register bit descriptions.

Table 16-109: APU\_CTLR bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	apu_region_4k	APU address regions are a minimum 4KB granularity 0 - Disabled. APU address regions can be as small as 64 bytes 1 - Enabled. APU address regions are a minimum 4KB granularity.	RO	Configuration dependent
[2]	sync_error_en	On access permission faults provide a SLVERR response on the transaction	RW	0
[1]	fmu_error_en	On access permission fault raise a FuSa error through FMU	RW	0
[0]	apu_enable	APU enable bit 0 - APU is disabled, no permission checking is performed and transactions pass through the interconnect 1 - APU is enabled and filters transactions based on programmed permissions The reset value of apu_enable comes from a strap pin. Once apu_enable is set to 1 it cannot be written to 0. Only reset will clear it to its reset value	RW	0

### 16.10.8 APU APU\_IIDR register

Contains information about the APU implementation.

#### Configurations

This register is only present if the APU is enabled on the interface.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xFFC

##### Type

RO

##### Reset value

0x00000001

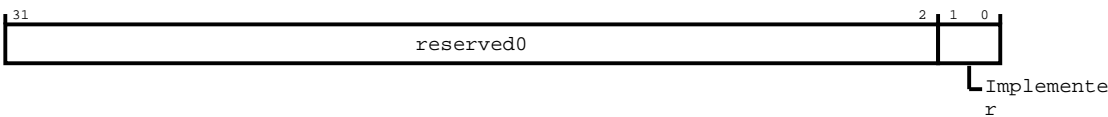
#### Constraints

None.

#### Bit descriptions

The following figure shows the APU\_IIDR register bit assignments.

**Figure 16-100: Bit assignment diagram for the APU\_IIDR register**



The following table shows the APU\_IIDR register bit descriptions.

**Table 16-110: APU\_IIDR bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1:0]	Implementer	Implementation selection 00 - RESERVED 01 - Arm implementation 10 - Customer implementation 11 - RESERVED	RO	0b01

## 16.11 FMU register summary

This section describes the FMU registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-111: FMU register summary**

Offset	Name	Type	Reset	Width	Description
0x0	FMU_ERR_FR_0	RO	0x00000000008008A2	64-bit	Error Record Feature Register for each functional block.
0x8	FMU_ERR_CTLR_0	RW	0x0000000000000001	64-bit	Error Record Control Register for each functional block.
0x10 + (0x40 x n), where n = record number	FMU_ERR_STATUS	RW	0x0000000000000000	64-bit	Error Record Primary Status Register for each functional block.
0x20 + (0x40 x n), where n = record number	FMU_ERR_MISCO	RW	0x0030000000000000	64-bit	<b>IMPLEMENTATION DEFINED</b> error syndrome register.
0x40 + (0x40 x n), where n = record number	FMU_ERR_FR	RO	0x0000000000000000	64-bit	Error Record Feature Register for each functional block.
0x48 + (0x40 x n), where n = record number	FMU_ERR_CTLR	RO	0x0000000000000000	64-bit	Error Record Control Register for each functional block.
0xE000	FMU_ERRGSR	RO	0x0000000000000000	64-bit	Error Group Status Register.
0xE100	FMU_ERRIDR	RO	0x43D0043B	32-bit	Error Record ID Register.
0xE200	FMU_KEY	RW	0x00000000	32-bit	FMU Key Register.
0xE204	FMU_SMEN	RW	0x00000000	32-bit	Safety Mechanism Enable Register.
0xE208	FMU_SMINJERR	RW	0x00000000	32-bit	Safety Mechanism Inject Error Register.
0xE210	FMU_SMINFO	RW	0x0000000000000000	64-bit	Used in conjunction with FMU_SMEN or FMU_SMINJERR register.
0xFFBC	FMU_ERRDEVARCH	RO	0x47710A00	32-bit	Device architecture register.
0xFFC8	FMU_ERRDEVID	RO	0x00000000	32-bit	Provides discovery information for the component.
0xFFE0	FMU_ERRPIDR0	RO	0x0000003d	32-bit	Peripheral Identification Register 0.
0xFFE4	FMU_ERRPIDR1	RO	0x000000b4	32-bit	Peripheral Identification Register 1.
0xFFE8	FMU_ERRPIDR2	RO	0x0000000B	32-bit	Peripheral Identification Register 2.
0xFFEC	FMU_ERRPIDR3	RO	0x00000000	32-bit	Peripheral Identification Register 3.
0xFFD0	FMU_ERRPIDR4	RO	0x00000044	32-bit	Peripheral Identification Register 4.
0xFFFF0	FMU_ERRCIDR0	RO	0x0000000D	32-bit	Component Identification Register 0.
0xFFFF4	FMU_ERRCIDR1	RO	0x000000F0	32-bit	Component Identification Register 1.
0xFFFF8	FMU_ERRCIDR2	RO	0x00000005	32-bit	Component Identification Register 2.
0xFFFFC	FMU_ERRCIDR3	RO	0x000000B1	32-bit	Component Identification Register 3.

### 16.11.1 FMU\_ERR\_FR\_0 register

Error Record Feature Register for each functional block.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

64-bit

##### Address offset

0x0

##### Type

RO

##### Reset value

0x0000000008008A2

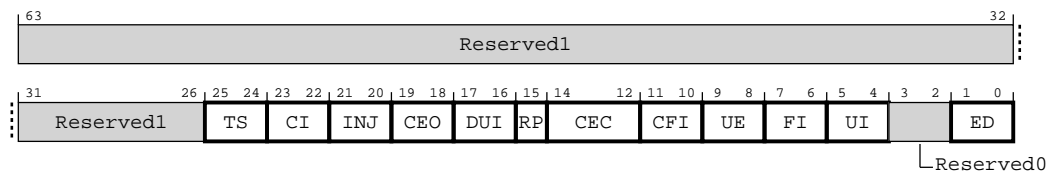
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERR\_FR\_0 register bit assignments.

**Figure 16-101: Bit assignment diagram for the FMU\_ERR\_FR\_0 register**



The following table shows the FMU\_ERR\_FR\_0 register bit descriptions.

**Table 16-112: FMU\_ERR\_FR\_0 bit descriptions**

Bits	Name	Description	Type	Reset
[63:26]	Reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[25:24]	TS	ERR_MISC3 used as timestamp register	RO	0b00
[23:22]	CI	Critical error interrupt and controls	RO	0b10
[21:20]	INJ	Fault injection mechanism	RO	0b00
[19:18]	CEO	Read as 0 if no CE counter implemented	RO	0b00



Bits	Name	Description	Type	Reset
[17:16]	DUI	Error recovery interrupts on deferred errors	RO	0b00
[15]	RP	Repeat counter	RO	0
[14:12]	CEC	Corrected error counter mechanism	RO	0b000
[11:10]	CFI	Control for enabling interrupts for corrected errors	RO	0b10
[9:8]	UE	Error reporting for in-band uncorrected error	RO	0b00
[7:6]	FI	Fault Handling Interrupt	RO	0b10
[5:4]	UI	Error Recovery Interrupt for Uncorrected Errors	RO	0b10
[3:2]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[1:0]	ED	Error reporting and logging	RO	0b10

### 16.11.2 FMU FMU\_ERR\_CTLR\_0 register

Error Record Control Register for each functional block.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

64-bit

##### Address offset

0x8

##### Type

RW

##### Reset value

0x0000000000000001

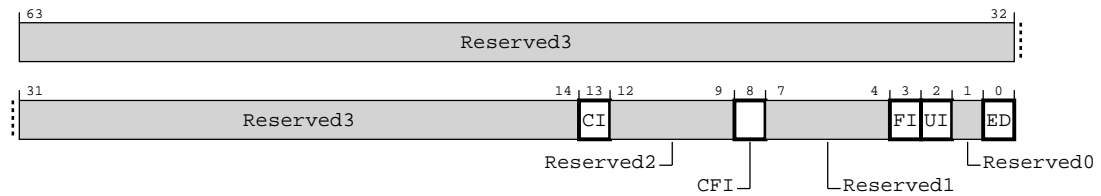
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERR\_CTLR\_0 register bit assignments.

**Figure 16-102: Bit assignment diagram for the FMU\_ERR\_CTLR\_0 register**



The following table shows the FMU\_ERR\_CTLR\_0 register bit descriptions.

**Table 16-113: FMU\_ERR\_CTLR\_0 bit descriptions**

Bits	Name	Description	Type	Reset
[63:14]	Reserved3	Bits within this register segment are reserved for future product development.	RO	0x0
[13]	CI	Critical error interrupt and controls.	RW	0
[12:9]	Reserved2	Bits within this register segment are reserved for future product development.	RO	0b0000
[8]	CFI	Fault handling interrupt for Corrected errors.	RW	0
[7:4]	Reserved1	Bits within this register segment are reserved for future product development.	RO	0b0000
[3]	FI	Fault Handling Interrupt (FHI) enable. This controls whether an FHI is generated for all detected and logged.	RW	0
[2]	UI	This controls whether an ERI is generated for all detected and logged.	RW	0
[1]	Reserved0	Bits within this register segment are reserved for future product development.	RO	0
[0]	ED	Error reporting and logging enable.	RW	1

### 16.11.3 FMU FMU\_ERR\_STATUS register

Error Record Primary Status Register for each functional block.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

64-bit

#### Address offset

$0 \times 10 + (0 \times 40 \times n)$ , where  $n$  is the error record entry number

#### Type

RW

#### Reset value

0x0000000000000000

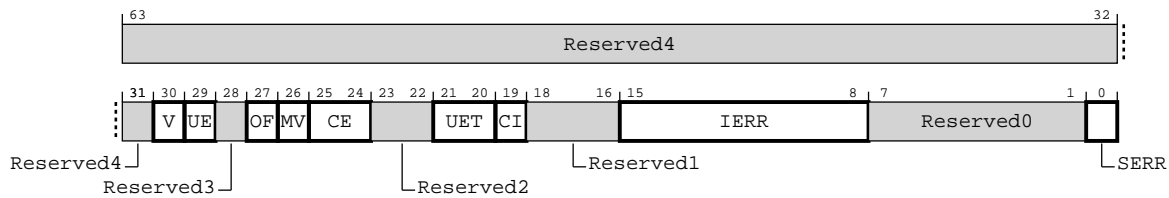
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the FMU\_ERR\_STATUS register bit assignments.

**Figure 16-103: Bit assignment diagram for the FMU\_ERR\_STATUS register**



The following table shows the FMU\_ERR\_STATUS register bit descriptions.

**Table 16-114: FMU\_ERR\_STATUS bit descriptions**

Bits	Name	Description	Type	Reset
[63:31]	Reserved4	Bits within this register segment are reserved for future product development.	RO	0x0
[30]	V	Status Register valid.  Write 1 to clear.	RW	0
[29]	UE	Uncorrected Error.  Write 1 to clear.	RW	0
[28]	Reserved3	Bits within this register segment are reserved for future product development.	RO	0
[27]	OF	Record has overflowed.  Write 1 to clear.	RW	0
[26]	MV	Additional information for error recorded.  Write 1 to clear.	RW	0
[25:24]	CE	Corrected Error bit.  Write 1 to clear.	RW	0b00
[23:22]	Reserved2	Bits within this register segment are reserved for future product development.	RO	0b00
[21:20]	UET	Uncorrected Error type.  Write 1 to clear.	RW	0b00
[19]	CI	Critical error condition indication.  Write 1 to clear.	RW	0
[18:16]	Reserved1	Bits within this register segment are reserved for future product development.	RO	0b000
[15:8]	IERR	Safety Mechanism ID code.	RW	0x0

Bits	Name	Description	Type	Reset
[7:1]	Reserved0	Bits within this register segment are reserved for future product development.	RO	0b0000000
[0]	SERR	Reads as zero if FMU_ERR(n)STATUS.V == 0. Otherwise, reads as 1.	RW	0

#### 16.11.4 FMU FMU\_ERR\_MISC0 register

**IMPLEMENTATION DEFINED** error syndrome register.

If the FMU\_ERR\_STATUS.V field in the corresponding status register for the error record is set to 0, the contents of the FMU\_ERR\_MISCO register are not valid and read as **UNKNOWN**.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

64-bit

## Address offset

0x20 + (0x40 x n), where n is the error record entry number

## Type

RW

## Reset value

```
0x0030000000000000
```

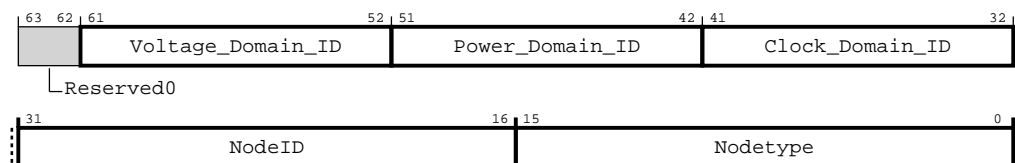
## Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the FMU\_ERR\_MISC0 register bit assignments.

**Figure 16-104: Bit assignment diagram for the FMU\_ERR\_MISC0 register**



The following table shows the FMU\_ERR\_MISC0 register bit descriptions.

**Table 16-115: FMU\_ERR\_MISC0 bit descriptions**

Bits	Name	Description	Type	Reset
[63:62]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[61:52]	Voltage_Domain_ID	Voltage domain ID of block reporting error	RO	0x3
[51:42]	Power_Domain_ID	Power domain ID of block reporting error	RO	0x0
[41:32]	Clock_Domain_ID	Clock domain ID of block reporting error	RO	0x0
[31:16]	NodeID	Node ID of block reporting error	RO	0x0
[15:0]	Nodetype	Nodetype of block reporting error	RO	0x0

### 16.11.5 FMU FMU\_ERR\_FR register

Error Record Feature Register for each functional block.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

64-bit

##### Address offset

$0x40 + (0x40 \times n)$ , where n is the error record entry number

##### Type

RO

##### Reset value

0x0000000000000000

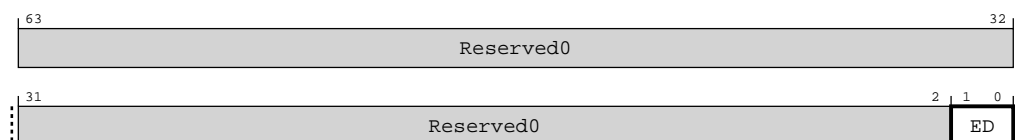
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERR\_FR register bit assignments.

**Figure 16-105: Bit assignment diagram for the FMU\_ERR\_FR register**



The following table shows the FMU\_ERR\_FR register bit descriptions.

Table 16-116: FMU\_ERR\_FR bit descriptions

Bits	Name	Description	Type	Reset
[63:2]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1:0]	ED	Error reporting and logging	RO	0b00

16.11.6 FMU FMU\_ERR\_CTLR register

Error Record Control Register for each functional block.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

64-bit

Address offset

0x48 + (0x40 x n), where n is the error record entry number

Type

RO

Reset value

0x0000000000000000

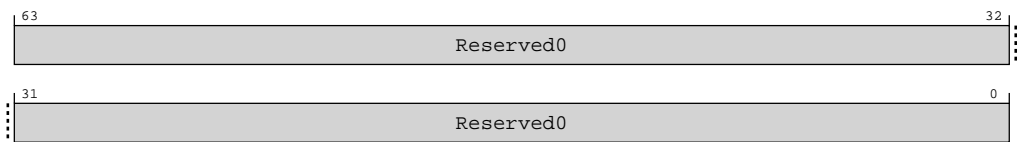
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERR\_CTLR register bit assignments.

Figure 16-106: Bit assignment diagram for the FMU\_ERR\_CTLR register



The following table shows the FMU\_ERR\_CTLR register bit descriptions.

**Table 16-117: FMU\_ERR\_CTLR bit descriptions**

Bits	Name	Description	Type	Reset
[63:0]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0

### 16.11.7 FMU FMU\_ERRGSR register

Error Group Status Register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

64-bit

##### Address offset

0xE000

##### Type

RO

##### Reset value

0x0000000000000000

#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERRGSR register bit assignments.

**Figure 16-107: Bit assignment diagram for the FMU\_ERRGSR register**



The following table shows the FMU\_ERRGSR register bit descriptions.

**Table 16-118: FMU\_ERRGSR bit descriptions**

Bits	Name	Description	Type	Reset
[63:0]	S	Indicates the status of Error Record	RO	0x0

## 16.11.8 FMU\_FMU\_ERRIDR register

Error Record ID Register.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0xE100

#### Type

RO

#### Reset value

0x43D1043B

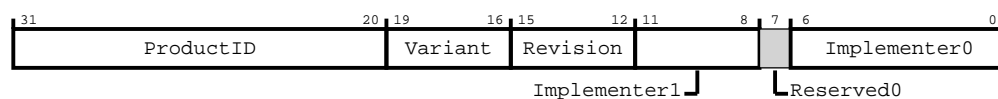
### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the FMU\_ERRIDR register bit assignments.

**Figure 16-108: Bit assignment diagram for the FMU\_ERRIDR register**



The following table shows the FMU\_ERRIDR register bit descriptions.

**Table 16-119: FMU\_ERRIDR bit descriptions**

Bits	Name	Description	Type	Reset
[31:20]	ProductID	Part number selected by designer	RO	0x43d
[19:16]	Variant	Together with Revision to differentiate revisions of component	RO	0b0001
[15:12]	Revision	Together with Variant to differentiate revisions of component	RO	0b0000
[11:8]	Implementer1	JEP106 bank identifier minus 1	RO	0b0100
[7]	Reserved0	Bits within this register segment are reserved for future product development	RO	0
[6:0]	Implementer0	JEP106 identification code for the designer of the component	RO	0b0111011



16.11.9 FMU FMU\_KEY register

FMU Key Register.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xE200

Type

RW

Reset value

0x00000000

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_KEY register bit assignments.

Figure 16-109: Bit assignment diagram for the FMU\_KEY register



The following table shows the FMU\_KEY register bit descriptions.

Table 16-120: FMU\_KEY bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	KEY	The required key to write to FMU registers	RW	0x0

16.11.10 FMU FMU\_SMEN register

Safety Mechanism Enable Register.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xE204

Type

RW

Reset value

0x00000000

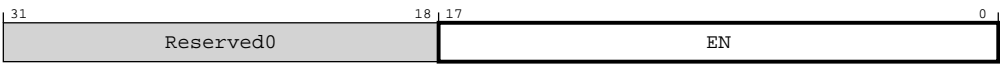
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_SMEN register bit assignments.

Figure 16-110: Bit assignment diagram for the FMU\_SMEN register



The following table shows the FMU\_SMEN register bit descriptions.

Table 16-121: FMU\_SMEN bit descriptions

Bits	Name	Description	Type	Reset
[31:18]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0

Bits	Name	Description	Type	Reset
[17:0]	EN	<p>The value of this field specifies which safety mechanisms report errors to the central Fault Management Unit (FMU). Each bit in this field corresponds to a specific safety mechanism.</p> <p><b>Bit[0]</b> DLS logic protection</p> <p><b>Bit[1]</b> External AMBA interface protection</p> <p><b>Bit[2]</b> Internal GT network CRC protection</p> <p><b>Bit[3]</b> Internal CFG_AUB network CRC protection (local)</p> <p><b>Bit[4]</b> Reset protection</p> <p><b>Bit[5]</b> External Q-channel interface protection</p> <p><b>Bit[6]</b> External P-channel interface protection</p> <p><b>Bit[7]</b> Internal CFG_AUB network CRC protection (remote)</p> <p><b>Bit[8]</b> Hang detector</p> <p><b>Bit[9]</b> Internal ERR_AUB network CRC protection</p> <p><b>Bit[10]</b> Access Protection Unit (APU)</p> <p><b>Bit[11]</b> Internal P-channel interface protection</p> <p><b>Bit[12]</b> Asynchronous signal protection</p> <p><b>Bit[13]</b> External Legacy ECC interface protection (uncorrectable error)</p> <p><b>Bit[14]</b> External Legacy ECC interface protection (correctable error)</p> <p><b>Bit[15]</b> Destination ID checker - Internal GT network</p> <p><b>Bit[16]</b> Destination ID checker - Internal CFG_AUB network (local and remote)</p> <p><b>Bit[17]</b> Destination ID checker - Internal ERR_AUB network</p>	RW	0x0

16.11.11 FMU FMU\_SMINJERR register

Safety Mechanism Inject Error Register.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xE208

Type

RW

Reset value

0x00000000

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_SMINJERR register bit assignments.

Figure 16-111: Bit assignment diagram for the FMU\_SMINJERR register



The following table shows the FMU\_SMINJERR register bit descriptions.

Table 16-122: FMU\_SMINJERR bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	SMID	Safety Mechanism identifier	WO	0b000000

### 16.11.12 FMU FMU\_SMINFO register

Used in conjunction with FMU\_SMEN or FMU\_SMINJERR register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

64-bit

##### Address offset

0xE210

##### Type

RW

##### Reset value

0x0000000000000000

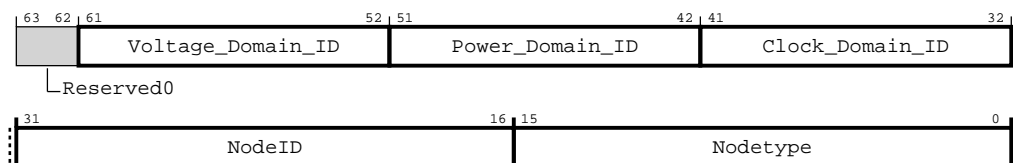
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_SMINFO register bit assignments.

**Figure 16-112: Bit assignment diagram for the FMU\_SMINFO register**



The following table shows the FMU\_SMINFO register bit descriptions.

**Table 16-123: FMU\_SMINFO bit descriptions**

Bits	Name	Description	Type	Reset
[63:62]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[61:52]	Voltage_Domain_ID	Voltage domain ID ID of block reporting error	RW	0x0
[51:42]	Power_Domain_ID	Power domain ID ID of block reporting error	RW	0x0
[41:32]	Clock_Domain_ID	Clock domain ID ID of block reporting error	RW	0x0
[31:16]	NodeID	Node ID of block reporting error	RW	0x0

Bits	Name	Description	Type	Reset
[15:0]	Nodetype	Nodetype of block reporting error	RW	0x0

### 16.11.13 FMU FMU\_ERRDEVARCH register

Device architecture register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xFFBC

##### Type

RO

##### Reset value

0x47710A00

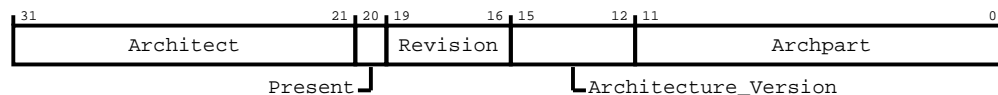
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERRDEVARCH register bit assignments.

**Figure 16-113: Bit assignment diagram for the FMU\_ERRDEVARCH register**



The following table shows the FMU\_ERRDEVARCH register bit descriptions.

**Table 16-124: FMU\_ERRDEVARCH bit descriptions**

Bits	Name	Description	Type	Reset
[31:21]	Architect	Define the architect of the component	RO	0x23b
[20]	Present	Define DEVARCH register is present	RO	1
[19:16]	Revision	Define the architecture revision of the component	RO	0b0001

Bits	Name	Description	Type	Reset
[15:12]	Architecture_Version	Define the architecture version of the component	RO	0b0000
[11:0]	Archpart	Define the architecture of the component	RO	0xa00

### 16.11.14 FMU FMU\_ERRDEVID register

Provides discovery information for the component.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xFFC8

##### Type

RO

##### Reset value

0x00000000

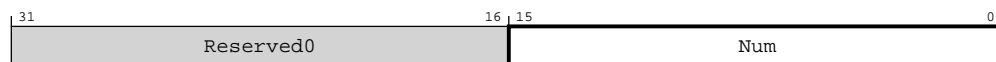
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERRDEVID register bit assignments.

**Figure 16-114: Bit assignment diagram for the FMU\_ERRDEVID register**



The following table shows the FMU\_ERRDEVID register bit descriptions.

**Table 16-125: FMU\_ERRDEVID bit descriptions**

Bits	Name	Description	Type	Reset
[31:16]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[15:0]	Num	Highest numbered index of the error records plus one	RO	0x0

16.11.15 FMU FMU\_ERRPIDR0 register

Peripheral Identification Register 0.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFE0

Type

RO

Reset value

0x0000003d

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRPIDR0 register bit assignments.

Figure 16-115: Bit assignment diagram for the FMU\_ERRPIDR0 register



The following table shows the FMU\_ERRPIDR0 register bit descriptions.

Table 16-126: FMU\_ERRPIDR0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	PART_0	Part number	RO	0x3d



16.11.16 FMU FMU\_ERRPIDR1 register

Peripheral Identification Register 1.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFE4

Type

RO

Reset value

0x000000b4

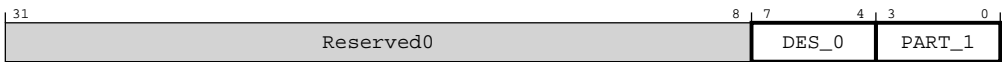
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRPIDR1 register bit assignments.

Figure 16-116: Bit assignment diagram for the FMU\_ERRPIDR1 register



The following table shows the FMU\_ERRPIDR1 register bit descriptions.

Table 16-127: FMU\_ERRPIDR1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:4]	DES_0	Designer JEP106 identification code, bits[3:0]	RO	0b1011
[3:0]	PART_1	Part number	RO	0b0100

### 16.11.17 FMU\_FMU\_ERRPIDR2 register

Peripheral Identification Register 2.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0xFFE8

##### Type

RO

##### Reset value

0x0000001B

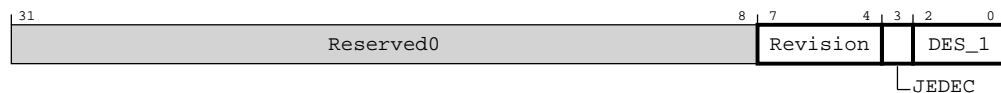
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the FMU\_ERRPIDR2 register bit assignments.

**Figure 16-117: Bit assignment diagram for the FMU\_ERRPIDR2 register**



The following table shows the FMU\_ERRPIDR2 register bit descriptions.

**Table 16-128: FMU\_ERRPIDR2 bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:4]	Revision	Component major revision	RO	0b0001
[3]	JEDEC	Implementer code	RO	1
[2:0]	DES_1	Designer, JEP106 identification code, bits[6:4]	RO	0b011

16.11.18 FMU FMU\_ERRPIDR3 register

Peripheral Identification Register 3.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFEC

Type

RO

Reset value

0x00000000

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRPIDR3 register bit assignments.

Figure 16-118: Bit assignment diagram for the FMU\_ERRPIDR3 register



The following table shows the FMU\_ERRPIDR3 register bit descriptions.

Table 16-129: FMU\_ERRPIDR3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:4]	REVAND	Component minor revision	RO	0b0000
[3:0]	CMOD	Indicate the component has been modified	RO	0b0000

16.11.19 FMU FMU\_ERRPIDR4 register

Peripheral Identification Register 4.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFD0

Type

RO

Reset value

0x00000044

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRPIDR4 register bit assignments.

Figure 16-119: Bit assignment diagram for the FMU\_ERRPIDR4 register



The following table shows the FMU\_ERRPIDR4 register bit descriptions.

Table 16-130: FMU\_ERRPIDR4 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development.	RO	0x0
[7:4]	Size	Designer. JEP106 continuation code.	RO	0b0100
[3:0]	DES_2	JEP106 band identifier minus 1.	RO	0b0100

16.11.20 FMU FMU\_ERRCIDR0 register

Component Identification Register 0.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFF0

Type

RO

Reset value

0x0000000D

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRCIDR0 register bit assignments.

Figure 16-120: Bit assignment diagram for the FMU\_ERRCIDR0 register



The following table shows the FMU\_ERRCIDR0 register bit descriptions.

Table 16-131: FMU\_ERRCIDR0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	PRMBL_0	Component identification preamble, segment 0	RO	0xd

16.11.21 FMU FMU\_ERRCIDR1 register

Component Identification Register 1.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFF4

Type

RO

Reset value

0x000000F0

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRCIDR1 register bit assignments.

Figure 16-121: Bit assignment diagram for the FMU\_ERRCIDR1 register



The following table shows the FMU\_ERRCIDR1 register bit descriptions.

Table 16-132: FMU\_ERRCIDR1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:4]	Class	Component class	RO	0b1111
[3:0]	PRMBL_1	Component identification preamble, segment 1	RO	0b0000

16.11.22 FMU FMU\_ERRCIDR2 register

Component Identification Register 2.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFF8

Type

RO

Reset value

0x00000005

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRCIDR2 register bit assignments.

Figure 16-122: Bit assignment diagram for the FMU\_ERRCIDR2 register



The following table shows the FMU\_ERRCIDR2 register bit descriptions.

Table 16-133: FMU\_ERRCIDR2 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	PRMBL_2	Component identification preamble, segment 2	RO	0x5

16.11.23 FMU FMU\_ERRCIDR3 register

Component Identification Register 3.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFFFC

Type

RO

Reset value

0x000000B1

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the FMU\_ERRCIDR3 register bit assignments.

Figure 16-123: Bit assignment diagram for the FMU\_ERRCIDR3 register



The following table shows the FMU\_ERRCIDR3 register bit descriptions.

Table 16-134: FMU\_ERRCIDR3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	PRMBL_3	Component identification preamble, segment 3	RO	0xb1



## 16.12 ASNI register summary

This section describes the ASNI registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-135: ASNI register summary**

Offset	Name	Type	Reset	Width	Description
0x0	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a node for ASNI registers.
0x4	<a href="#">node_info</a>	RO	See individual bit resets.	32-bit	This register provides node information for ASNI, such as data width.
0x08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register controls Secure access.
0x0C	<a href="#">pmusela</a>	RW	0x00000000	32-bit	This register is used to select the event values in the ASNI event crossbar.
0x10	<a href="#">pmuselb</a>	RW	0x00000000	32-bit	This register is used to select the event values in the ASNI event crossbar.
0x14	<a href="#">interface_id_0_3</a>	RO	See individual bit resets.	32-bit	Contains information about the ASNI interface IDs for interfaces 0-3.
0x24	<a href="#">num_sub_features</a>	RO	See individual bit resets.	32-bit	Number of sub features.
0x28	<a href="#">sub_feature_0_type</a>	RO	See individual bit resets.	32-bit	Sub feature 0 type.
0x2C	<a href="#">sub_feature_0_pointer</a>	RO	See individual bit resets.	32-bit	Sub feature 0 pointer.
0x44	<a href="#">burst_split_control</a>	RW	See individual bit resets.	32-bit	This register shows the Burst split value to apply and the Burst split value that is applied.
0x48	<a href="#">address_remap</a>	RW	0x00000000	32-bit	This register is used to program up to eight remap states that are supported by the address decode logic.
0x4C	<a href="#">hang_detector_ctrl</a>	RW	0x00000000	32-bit	Registers used to configure the hang detector. Fields in this register are only present if the hang detector feature is enabled.
0x80	<a href="#">silicon_debug</a>	RW	0x00000000	32-bit	This register monitors the status of completer interface channels.
0x84	<a href="#">qosctl</a>	RW	0x00000000	32-bit	This register controls the QoS settings for BQV and TSPEC and enables a QoS value on inbound transactions to be overridden.
0x88	<a href="#">wdatthrs</a>	RW	0x00000000	32-bit	This register specifies the number of write data beats to be queued before the write packet is sent.
0x8C	<a href="#">arqos_value</a>	RW	0x00000000	32-bit	This register contains controls for configuring the override value for the ARQOS signal on the ASNI.
0x90	<a href="#">awqos_value</a>	RW	0x00000000	32-bit	This register contains controls for configuring the override value for the AWQOS signal on the ASNI.
0x94	<a href="#">atqosot</a>	RW	0x00000000	32-bit	Registers used to configure and store the write the maximum number of outstanding atomic transactions for the interface.
0x98	<a href="#">arqosot</a>	RW	0x00000000	32-bit	Registers used to configure and store the write the maximum number of outstanding read transactions for the interface.
0x9C	<a href="#">awqosot</a>	RW	0x00000000	32-bit	Registers used to configure and store the write the maximum number of outstanding write transactions for the interface.

Offset	Name	Type	Reset	Width	Description
0xA0	<a href="#">axqosot</a>	RW	0x00000000	32-bit	Registers used to configure and store the maximum number of outstanding read and write transactions for the interface.
0xA4	<a href="#">qosrdpk</a>	RW	0x00000000	32-bit	This register controls the QoS peak rate for the read hard bandwidth regulation, TSPEC, of a completer interface.
0xA8	<a href="#">qosrdbur</a>	RW	0x00000000	32-bit	This register controls the QoS burstiness for the read hard bandwidth regulation, TSPEC, of a completer interface.
0xAC	<a href="#">qosrdavg</a>	RW	0x00000000	32-bit	This register controls the QoS average rate for the read hard bandwidth regulation, TSPEC, of a completer interface.
0xB0	<a href="#">qoswrpk</a>	RW	0x00000000	32-bit	This register controls the QoS peak rate for the write hard bandwidth regulation, TSPEC, of a completer interface.
0xB4	<a href="#">qoswrbur</a>	RW	0x00000000	32-bit	This register controls the QoS burstiness for the write hard bandwidth regulation, TSPEC, of a completer interface.
0xB8	<a href="#">qoswrvavg</a>	RW	0x00000000	32-bit	This register controls the QoS average rate for the write hard bandwidth regulation, TSPEC, of a completer interface.
0xBC	<a href="#">qoscompk</a>	RW	0x00000000	32-bit	This register controls the QoS peak rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.
0xC0	<a href="#">qoscombur</a>	RW	0x00000000	32-bit	This register controls the QoS burstiness allowance for combined read and write hard bandwidth regulation, TSPEC, of a completer interface.
0xC4	<a href="#">qoscomavg</a>	RW	0x00000000	32-bit	This register controls the QoS average rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.
0xC8	<a href="#">qosrdbqv</a>	RW	0x00000000	32-bit	This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for read soft bandwidth regulation, BQV, of a completer interface.
0xCC	<a href="#">qoswrbqv</a>	RW	0x00000000	32-bit	This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for write soft bandwidth regulation, BQV, of a completer interface.
0xD0	<a href="#">qoscombqv</a>	RW	0x00000000	32-bit	This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for both read and write soft bandwidth regulation, BQV, of a completer interface.
0xE0	<a href="#">read_channel_mpam_override</a>	RW	0x00000000	32-bit	This register controls the ASNI read channel MPAM override behavior.
0xE4	<a href="#">write_channel_mpam_override</a>	RW	0x00000000	32-bit	This register controls the ASNI write channel MPAM override behavior.
0x100	<a href="#">idm_device_id</a>	RO	See individual bit resets.	32-bit	This register indicates the statically configured device ID value and is implemented if IDM is enabled.
0x104	<a href="#">idm_config</a>	RW	See individual bit resets.	32-bit	This register enables transaction logging, error detection, timeout detection, access control, and reset control.
0x108	<a href="#">idm_errctlr</a>	RW	0x00000000	32-bit	This register controls how errors are handled.
0x110	<a href="#">idm_errstatus</a>	RW	0x00000000	32-bit	This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.
0x114	<a href="#">idm_erraddr_lsb</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x118	<a href="#">idm_erraddr_msb</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x128	<a href="#">idm_errmisc0</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x12C	<a href="#">idm_errmisc1</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x130	<a href="#">idm_access_control</a>	RW	0x00000000	32-bit	This register controls the state, gated or ungated, of a device.

Offset	Name	Type	Reset	Width	Description
0x134	<a href="#">idm_access_status</a>	RO	0x00000002	32-bit	This register indicates the access status for Secure transactions.
0x138	<a href="#">idm_access_readid</a>	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x13C	<a href="#">idm_access_writeid</a>	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x140	<a href="#">idm_reset_control</a>	RW	0x00000002	32-bit	This register controls the reset of a device that is attached to the interconnect.
0x144	<a href="#">idm_reset_status</a>	RO	0x00000000	32-bit	This register indicates mostly the reset status of Secure transactions. However, the <code>rst_exit_state</code> field indicates reset exit state of secure or non-secure transactions.
0x148	<a href="#">idm_reset_readid</a>	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x14C	<a href="#">idm_reset_writeid</a>	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x150	<a href="#">idm_timeout_control</a>	RW	0x00000000	32-bit	This register is present when timeout detection is configured.
0x154	<a href="#">idm_timeout_value</a>	RW	0x00000004	32-bit	This register controls the duration that is used to determine if a transaction has timed out.
0x158	<a href="#">idm_interrupt_status</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0x15C	<a href="#">idm_interrupt_mask</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0x160	<a href="#">idm_errstatus_ns</a>	RW	0x00000000	32-bit	This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.
0x164	<a href="#">idm_erraddr_lsb_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x168	<a href="#">idm_erraddr_msb_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x178	<a href="#">idm_errmisc0_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x17C	<a href="#">idm_errmisc1_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x184	<a href="#">idm_access_status_ns</a>	RO	0x00000000	32-bit	This register indicates the access status for Non-secure transactions.
0x188	<a href="#">idm_access_readid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x18C	<a href="#">idm_access_writeid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x194	<a href="#">idm_reset_status_ns</a>	RO	0x00000000	32-bit	This register indicates the reset status of Non-secure transactions.
0x198	<a href="#">idm_reset_readid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x19C	<a href="#">idm_reset_writeid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x1A8	<a href="#">idm_interrupt_status_ns</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0x1AC	<a href="#">idm_interrupt_mask_ns</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.

### 16.12.1 ASNI node\_type register

This register identifies the node type as a node for ASNI registers.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

**Width**

32-bit

**Address offset**

0x0

**Type**

RO

**Reset value**

See individual bit resets.

**Constraints**

None.

**Bit descriptions**

The following figure shows the node\_type register bit assignments.

**Figure 16-124: Bit assignment diagram for the node\_type register**



The following table shows the node\_type register bit descriptions.

**Table 16-136: node\_type bit descriptions**

Bits	Name	Description	Type	Reset
[31:16]	node_id	The ASNI ID that is assigned during network construction.	RO	Configuration dependent
[15:0]	node_type	Identifies the associated node type as a node for ASNI registers. The reset value of this field is 0x4.	RO	0x4

**16.12.2 ASNI node\_info register**

This register provides node information for ASNI, such as data width.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**

32-bit

**Address offset**

0x4

## Type

RO

## Reset value

See individual bit resets.

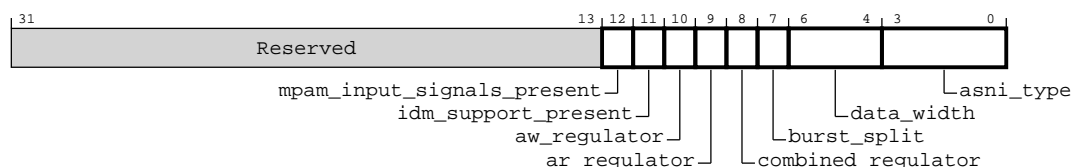
## Constraints

None.

## Bit descriptions

The following figure shows the node\_info register bit assignments.

**Figure 16-125: Bit assignment diagram for the node\_info register**



The following table shows the node\_info register bit descriptions.

**Table 16-137: node\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:13]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[12]	mpam_input_signals_present	MPAM input signals present:  <b>0</b> MPAM input signal not present  <b>1</b> MPAM input signal is present  If MPAM input signals are not present, then the MPAM value is driven from the MPAM override register, regardless of the MPAM override enable bit.	RO	Configuration dependent
[11]	idm_support_present	IDM support present  <b>0</b> IDM support logic is not present  <b>1</b> IDM support logic is present	RO	Configuration dependent
[10]	aw_regulator	AW regulator is present:  <b>0</b> AW regulator logic not present  <b>1</b> AW regulator logic is present	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[9]	ar_regulator	AR regulator is present: <b>0</b> AR regulator logic not present <b>1</b> AR regulator logic is present	RO	Configuration dependent
[8]	combined_regulator	Combined AR and AW regulator present: <b>0</b> Combined AR and AW regulator logic is not present <b>1</b> Combined AR and AW regulator logic is present	RO	Configuration dependent
[7]	burst_split	Burst split present: <b>0</b> Burst split logic is not present <b>1</b> Burst split logic is present	RO	Configuration dependent
[6:4]	data_width	Data width, AxSIZE encoded: <b>0b000</b> Reserved <b>0b001</b> Reserved <b>0b010</b> 4 bytes <b>0b011</b> 8 bytes <b>0b100</b> 16 bytes <b>0b101</b> 32 bytes <b>0b110</b> 64 bytes <b>0b111</b> 128 bytes	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[3:0]	asni_type	ASNI type:  <b>0b0000</b> Reserved  <b>0b0001</b> Reserved  <b>0b0010</b> AXI Issue F  <b>0b0011</b> ACE-Lite  <b>0b0100</b> AXI Issue G  <b>0b0101</b> AXI Issue H  <b>0b0110-0b1111</b> Reserved	RO	Configuration dependent

### 16.12.3 ASNI secure\_access register

This register controls Secure access.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x08

##### Type

RW

##### Reset value

0x00000000

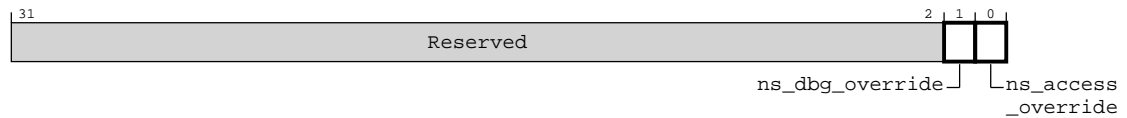
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the secure\_access register bit assignments.

**Figure 16-126: Bit assignment diagram for the secure\_access register**



The following table shows the secure\_access register bit descriptions.

**Table 16-138: secure\_access bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[1]	ns_dbg_override	Enables/Disables non-secure access to clock domain PMU and interface registers	RW	0
[0]	ns_access_override	Enables/Disables non-secure access to clock domain registers	RW	0

## 16.12.4 ASNI pmusela register

This register is used to select the event values in the ASNI event crossbar.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x0C

#### Type

RW

#### Reset value

0x00000000

### Constraints

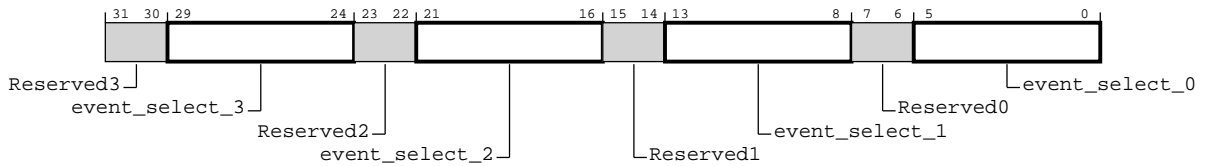
Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the pmusela register bit assignments.



**Figure 16-127: Bit assignment diagram for the pmusela register**



The following table shows the pmusela register bit descriptions.

**Table 16-139: pmusela bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

### 16.12.5 ASNI pmusela register

This register is used to select the event values in the ASNI event crossbar.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x10

##### Type

RW

##### Reset value

0x00000000

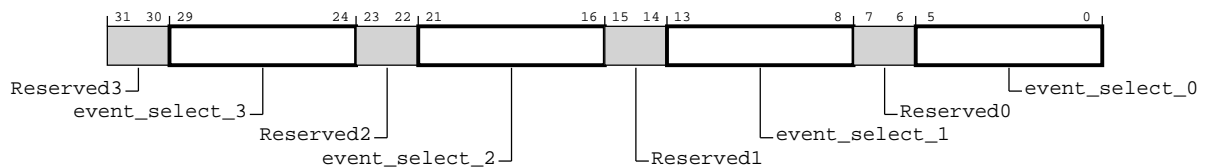
## Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmuselib register bit assignments.

**Figure 16-128: Bit assignment diagram for the pmuselib register**



The following table shows the pmuselib register bit descriptions.

**Table 16-140: pmuselib bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

### 16.12.6 ASNI interface\_id\_0\_3 register

Contains information about the ASNI interface IDs for interfaces 0-3.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

### Width

32-bit

### Address offset

0x14

Type

RO

Reset value

See individual bit resets.

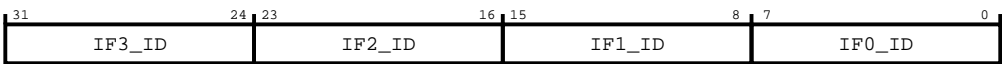
Constraints

None.

Bit descriptions

The following figure shows the interface\_id\_0\_3 register bit assignments.

Figure 16-129: Bit assignment diagram for the interface\_id\_0\_3 register



The following table shows the interface\_id\_0\_3 register bit descriptions.

Table 16-141: interface\_id\_0\_3 bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	IF3_ID	Reserved	RO	Configuration dependent
[23:16]	IF2_ID	Reserved	RO	Configuration dependent
[15:8]	IF1_ID	Reserved	RO	Configuration dependent
[7:0]	IF0_ID	ASNI interface ID 0	RO	Configuration dependent

16.12.7 ASNI num\_sub\_features register

Number of sub features.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x24

Type

RO

**Reset value**

See individual bit resets.

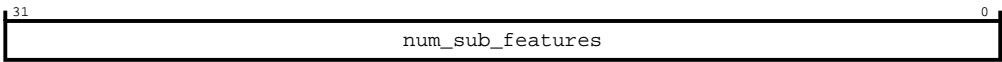
**Constraints**

None.

**Bit descriptions**

The following figure shows the num\_sub\_features register bit assignments.

**Figure 16-130: Bit assignment diagram for the num\_sub\_features register**



The following table shows the num\_sub\_features register bit descriptions.

**Table 16-142: num\_sub\_features bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	num_sub_features	Number of sub features	RO	Configuration dependent

**16.12.8 ASNI sub\_feature\_0\_type register**

Sub feature 0 type.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**

32-bit

**Address offset**

0x28

**Type**

RO

**Reset value**

See individual bit resets.

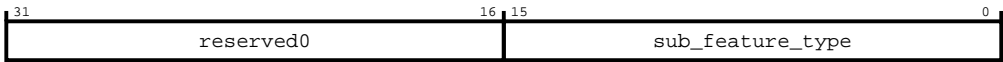
**Constraints**

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_type register bit assignments.

Figure 16-131: Bit assignment diagram for the sub\_feature\_0\_type register



The following table shows the sub\_feature\_0\_type register bit descriptions.

Table 16-143: sub\_feature\_0\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[15:0]	sub_feature_type	Sub feature 0 type	RO	Configuration dependent

16.12.9 ASNI sub\_feature\_0\_pointer register

Sub feature 0 pointer.

Configurations

The number of registers of this type that are present depends on the number of subfeatures in the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x2C

Type

RO

Reset value

See individual bit resets.

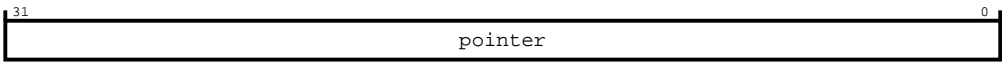
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_pointer register bit assignments.

Figure 16-132: Bit assignment diagram for the sub\_feature\_0\_pointer register



The following table shows the sub\_feature\_0\_pointer register bit descriptions.

Table 16-144: sub\_feature\_0\_pointer bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	pointer	Sub feature 0 pointer	RO	Configuration dependent

16.12.10 ASNI burst\_split\_control register

This register shows the Burst split value to apply and the Burst split value that is applied.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x44

Type

RW

Reset value

See individual bit resets.

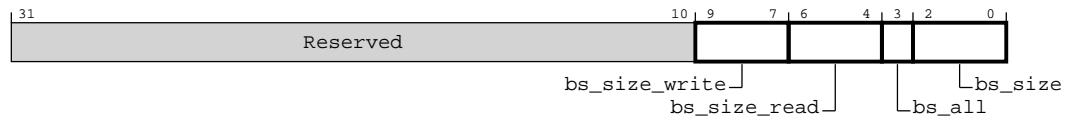
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the burst\_split\_control register bit assignments.

**Figure 16-133: Bit assignment diagram for the burst\_split\_control register**



The following table shows the burst\_split\_control register bit descriptions.

**Table 16-145: burst\_split\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:10]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[9:7]	bs_size_write	The value of Burst split size that is applied on the write channel. The value is based on the size of the address stripe. This field indicates the applied Burst size. The values are the lower of: <ul style="list-style-type: none"> <li>The configured minimum address stripe size, entered through the address map * Bits [2:0] of this register</li> </ul> This field is read only.	RO	Configuration dependent
[6:4]	bs_size_read	The value of Burst split size that is applied on the read channel. The value is based on the size of the address stripe. This field indicates the applied Burst size. The values are the lower of: <ul style="list-style-type: none"> <li>The configured minimum address stripe size, entered through the address map * Bits [2:0] of this register</li> </ul> This field is read only.	RO	Configuration dependent
[3]	bs_all	Burst split all. If set, modifiable Bursts to non-striped regions are also split. This field is read/write.	RW	0
[2:0]	bs_size	The value of Burst split size to apply. The supported encodings are: <p><b>0b000-0b001</b> Reserved</p> <p><b>0b010</b> 128 bytes</p> <p><b>0b011</b> 256 bytes</p> <p><b>0b100</b> 512 bytes</p> <p><b>0b101</b> 1024 bytes</p> <p><b>0b110</b> 2048 bytes</p> <p><b>0b111</b> 4096 bytes, no Burst split</p> This field is read/write.	RW	0b111

16.12.11 ASNI address\_remap register

This register is used to program up to eight remap states that are supported by the address decode logic.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x48

Type

RW

Reset value

0x00000000

Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the address\_remap register bit assignments.

Figure 16-134: Bit assignment diagram for the address\_remap register



The following table shows the address\_remap register bit descriptions.

Table 16-146: address\_remap bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	remap	If multiple bits are set, the bit for each remap with the lowest bit set is taken.	RW	0x0



16.12.12 ASNI hang\_detector\_ctrl register

Registers used to configure the hang detector. Fields in this register are only present if the hang detector feature is enabled.

Configurations

This register is only present if timeout detection is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x4c

Type

RW

Reset value

0x00000000

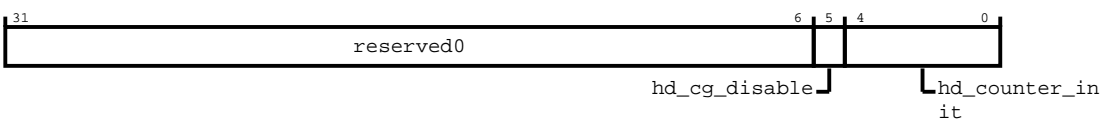
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the hang\_detector\_ctrl register bit assignments.

Figure 16-135: Bit assignment diagram for the hang\_detector\_ctrl register



The following table shows the hang\_detector\_ctrl register bit descriptions.

Table 16-147: hang\_detector\_ctrl bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0

Bits	Name	Description	Type	Reset
[5]	hd_cg_disable	Hang detector clock gate disable: <b>0</b> clock gate in the hang detector is enabled <b>1</b> clock gate in the hang detector is disabled, which disables the hang detector	RW	0

Bits	Name	Description	Type	Reset
[4:0]	hd_counter_init	<p>Timeout setting for the hang detector. Each 5'dx encoding encodes a timeout range, shown in both clock cycles and duration for a 1GHz clock.</p> <p><b>5'd0</b> Timeout range in clock cycles = <math>3 \times 2^{30}</math> to <math>4 \times 2^{30}</math>. Timeout range in duration at 1GHz = 3.0s to 4s.</p> <p><b>5'd1</b> Timeout range in clock cycles = <math>3 \times 2^{29}</math> to <math>4 \times 2^{29}</math>. Timeout range in duration at 1GHz = 1.6s to 2s.</p> <p><b>5'd2</b> Timeout range in clock cycles = <math>3 \times 2^{28}</math> to <math>4 \times 2^{28}</math>. Timeout range in duration at 1GHz = 805ms to 1s.</p> <p><b>5'd3</b> Timeout range in clock cycles = <math>3 \times 2^{27}</math> to <math>4 \times 2^{27}</math>. Timeout range in duration at 1GHz = 403ms to 537ms.</p> <p><b>5'd4</b> Timeout range in clock cycles = <math>3 \times 2^{26}</math> to <math>4 \times 2^{26}</math>. Timeout range in duration at 1GHz = 201ms to 268ms.</p> <p><b>5'd5</b> Timeout range in clock cycles = <math>3 \times 2^{25}</math> to <math>4 \times 2^{25}</math>. Timeout range in duration at 1GHz = 101ms to 134ms.</p> <p><b>5'd6</b> Timeout range in clock cycles = <math>3 \times 2^{24}</math> to <math>4 \times 2^{24}</math>. Timeout range in duration at 1GHz = 50ms to 67ms.</p> <p><b>5'd7</b> Timeout range in clock cycles = <math>3 \times 2^{23}</math> to <math>4 \times 2^{23}</math>. Timeout range in duration at 1GHz = 25ms to 34ms.</p> <p><b>5'd8</b> Timeout range in clock cycles = <math>3 \times 2^{22}</math> to <math>4 \times 2^{22}</math>. Timeout range in duration at 1GHz = 12.6ms to 17ms.</p> <p><b>5'd9</b> Timeout range in clock cycles = <math>3 \times 2^{21}</math> to <math>4 \times 2^{21}</math>. Timeout range in duration at 1GHz = 6.3ms to 8.4ms.</p> <p><b>5'd10</b> Timeout range in clock cycles = <math>3 \times 2^{20}</math> to <math>4 \times 2^{20}</math>. Timeout range in duration at 1GHz = 3.15ms to 4.2ms.</p> <p><b>5'd11</b> Timeout range in clock cycles = <math>3 \times 2^{19}</math> to <math>4 \times 2^{19}</math>. Timeout range in duration at 1GHz = 1.6ms to 2.1ms.</p> <p><b>5'd12</b> Timeout range in clock cycles = <math>3 \times 2^{18}</math> to <math>4 \times 2^{18}</math>. Timeout range in duration at 1GHz = 786us to 1.0ms.</p> <p><b>5'd13</b> Timeout range in clock cycles = <math>3 \times 2^{17}</math> to <math>4 \times 2^{17}</math>. Timeout range in duration at 1GHz = 393us to 524us.</p> <p><b>5'd14</b> Timeout range in clock cycles = <math>3 \times 2^{16}</math> to <math>4 \times 2^{16}</math>. Timeout range in duration at 1GHz = 196us to 262us.</p>	RW	0b00000

Bits	Name	Description	Type	Reset
[4:0]	hd_counter_init	<p><b>5'd15</b> Timeout range in clock cycles = <math>3 \times 2^{15}</math> to <math>4 \times 2^{15}</math>. Timeout range in duration at 1GHz = 98us to 131us.</p> <p><b>5'd16</b> Timeout range in clock cycles = <math>3 \times 2^{14}</math> to <math>4 \times 2^{14}</math>. Timeout range in duration at 1GHz = 49us to 65us.</p> <p><b>5'd17</b> Timeout range in clock cycles = <math>3 \times 2^{13}</math> to <math>4 \times 2^{13}</math>. Timeout range in duration at 1GHz = 25us to 33us.</p> <p><b>5'd18</b> Timeout range in clock cycles = <math>3 \times 2^{12}</math> to <math>4 \times 2^{12}</math>. Timeout range in duration at 1GHz = 12us to 16us.</p> <p><b>5'd19</b> Timeout range in clock cycles = <math>3 \times 2^{11}</math> to <math>4 \times 2^{11}</math>. Timeout range in duration at 1GHz = 6us to 8.2us.</p> <p><b>5'd20</b> Timeout range in clock cycles = <math>3 \times 2^{10}</math> to <math>4 \times 2^{10}</math>. Timeout range in duration at 1GHz = 3us to 4.1us.</p> <p><b>5'd21</b> Timeout range in clock cycles = <math>3 \times 2^9</math> to <math>4 \times 2^9</math>. Timeout range in duration at 1GHz = 1.5us to 2.0us.</p> <p><b>5'd22</b> Timeout range in clock cycles = <math>3 \times 2^8</math> to <math>4 \times 2^8</math>. Timeout range in duration at 1GHz = 768ns to 1.0us.</p> <p><b>5'd23</b> Timeout range in clock cycles = <math>3 \times 2^7</math> to <math>4 \times 2^7</math>. Timeout range in duration at 1GHz = 384ns to 512ns.</p> <p><b>5'd24</b> Timeout range in clock cycles = <math>3 \times 2^6</math> to <math>4 \times 2^6</math>. Timeout range in duration at 1GHz = 192ns to 256ns.</p> <p><b>5'd25</b> Timeout range in clock cycles = <math>3 \times 2^5</math> to <math>4 \times 2^5</math>. Timeout range in duration at 1GHz = 96ns to 128ns.</p> <p><b>5'd26</b> Timeout range in clock cycles = <math>3 \times 2^4</math> to <math>4 \times 2^4</math>. Timeout range in duration at 1GHz = 48ns to 64ns.</p> <p><b>5'd27</b> Timeout range in clock cycles = <math>3 \times 2^3</math> to <math>4 \times 2^3</math>. Timeout range in duration at 1GHz = 24ns to 32ns.</p> <p><b>5'd28</b> Illegal</p> <p><b>5'd29</b> Illegal</p> <p><b>5'd30</b> Illegal</p>	RW	0b00000

Bits	Name	Description	Type	Reset
[4:0]	hd_counter_init	5'd31 Illegal	RW	0b00000

16.12.13 ASNI silicon\_debug register

This register monitors the status of completer interface channels.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x80

Type

RW

Reset value

0x00000000

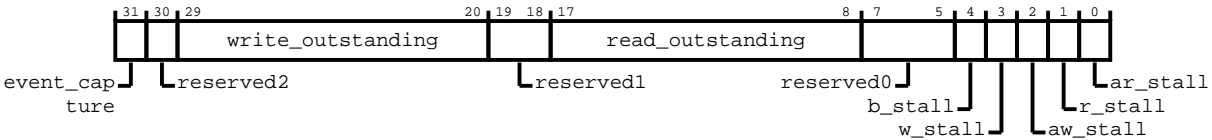
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the silicon\_debug register bit assignments.

Figure 16-136: Bit assignment diagram for the silicon\_debug register



The following table shows the silicon\_debug register bit descriptions.

Table 16-148: silicon\_debug bit descriptions

Bits	Name	Description	Type	Reset
[31]	event_capture	Enable capture	RW	0

Bits	Name	Description	Type	Reset
[30]	reserved2	Bits within this register segment are reserved for future product development	RO	0
[29:20]	write_outstanding	Indicates that the interface has writes that are outstanding.	RO	0x0
[19:18]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[17:8]	read_outstanding	Indicates that the interface has reads that are outstanding.	RO	0x0
[7:5]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[4]	b_stall	Indicates that the B channel is stalled.	RO	0
[3]	w_stall	Indicates that the W channel is stalled.	RO	0
[2]	aw_stall	Indicates that the AW channel is stalled.	RO	0
[1]	r_stall	Indicates that the R channel is stalled.	RO	0
[0]	ar_stall	Indicates that the AR channel is stalled.	RO	0

### 16.12.14 ASNI qosctl register

This register controls the QoS settings for BQV and TSPEC and enables a QoS value on inbound transactions to be overridden.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x84

##### Type

RW

##### Reset value

0x00000000

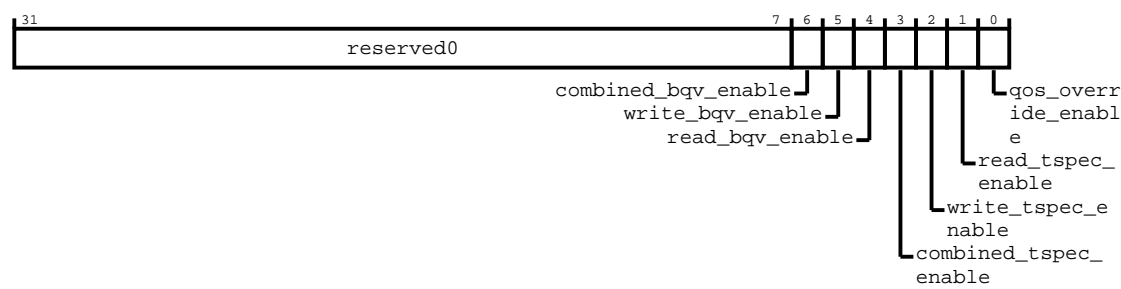
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the qosctl register bit assignments.

Figure 16-137: Bit assignment diagram for the qosctl register



The following table shows the qosctl register bit descriptions.

Table 16-149: qosctl bit descriptions

Bits	Name	Description	Type	Reset
[31:7]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[6]	combined_bqv_enable	Enables combined BQV	RW	0
[5]	write_bqv_enable	Enables Write BQV	RW	0
[4]	read_bqv_enable	Enables Read BQV	RW	0
[3]	combined_tspec_enable	Enables combined TSPEC	RW	0
[2]	write_tspec_enable	Enables Write TSPEC	RW	0
[1]	read_tspec_enable	Enables Read TSPEC	RW	0
[0]	qos_override_enable	When set, this bit enables a QoS value on inbound transactions to be overridden	RW	0

16.12.15 ASNI wdatthrs register

This register specifies the number of write data beats to be queued before the write packet is sent.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x88

Type

RW

Reset value

0x00000000

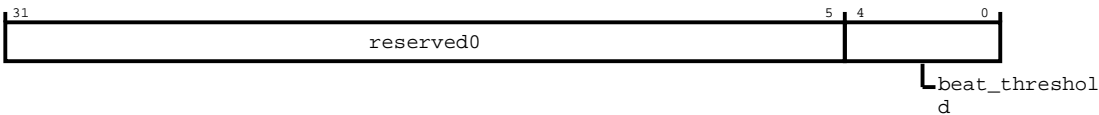
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the wdatthrs register bit assignments.

Figure 16-138: Bit assignment diagram for the wdatthrs register



The following table shows the wdatthrs register bit descriptions.

Table 16-150: wdatthrs bit descriptions

Bits	Name	Description	Type	Reset
[31:5]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[4:0]	beat_threshold	Write data threshold decimal value. Specifies the number of write data beats to be buffered before the write data packet is sent.	RW	0b00000

16.12.16 ASNI arqos\_value register

This register contains controls for configuring the override value for the ARQOS signal on the ASNI.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x8C

Type

RW

Reset value

0x00000000



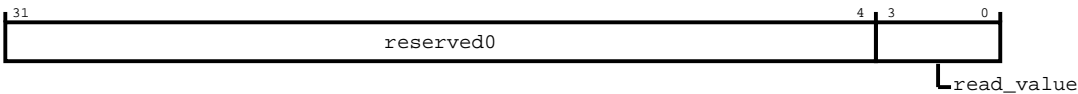
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the arqos\_value register bit assignments.

Figure 16-139: Bit assignment diagram for the arqos\_value register



The following table shows the arqos\_value register bit descriptions.

Table 16-151: arqos\_value bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3:0]	read_value	ARQOS value override for the completer interface. This value is applied to transactions at this interface if the following configuration scenario is present: <ul style="list-style-type: none"><li>The QOSOVERRIDE input signal bit is HIGH or if the QOS_OVERRIDE_ENABLE bit of ASNI_QOSCTL register is HIGH</li><li>The BQV enable bits of ASNI_QOSCTL register are not set</li></ul>	RW	0b0000

16.12.17 ASNI awqos\_value register

This register contains controls for configuring the override value for the AWQOS signal on the ASNI.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x90

Type

RW

Reset value

0x00000000

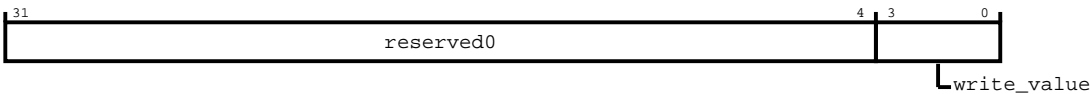
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the awqos\_value register bit assignments.

Figure 16-140: Bit assignment diagram for the awqos\_value register



The following table shows the awqos\_value register bit descriptions.

Table 16-152: awqos\_value bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3:0]	write_value	AWQOS value override for the completer interface. This value is applied to transactions at this interface if the following configuration scenario is present: <ul style="list-style-type: none"><li>The QOSOVERRIDE input signal bit is HIGH or if the QOS_OVERRIDE_ENABLE bit of ASNI_QOSCTL register is HIGH</li><li>The BQV enable bits of ASNI_QOSCTL register are not set</li></ul>	RW	0b0000

16.12.18 ASNI atqosot register

Registers used to configure and store the write the maximum number of outstanding atomic transactions for the interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x94

Type

RW

Reset value

0x00000000

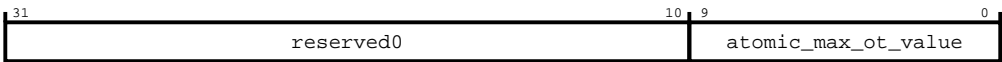
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the atqosot register bit assignments.

Figure 16-141: Bit assignment diagram for the atqosot register



The following table shows the atqosot register bit descriptions.

Table 16-153: atqosot bit descriptions

Bits	Name	Description	Type	Reset
[31:10]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[9:0]	atomic_max_ot_value	Configurable register to set the maximum outstanding atomic transactions for the interface	RW	0x0

16.12.19 ASNl arqosot register

Registers used to configure and store the write the maximum number of outstanding read transactions for the interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x98

Type

RW

Reset value

0x00000000

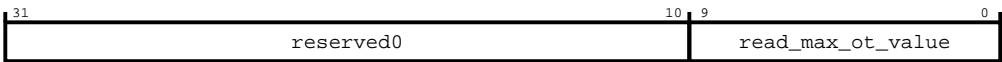
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the arqosot register bit assignments.

Figure 16-142: Bit assignment diagram for the arqosot register



The following table shows the arqosot register bit descriptions.

Table 16-154: arqosot bit descriptions

Bits	Name	Description	Type	Reset
[31:10]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[9:0]	read_max_ot_value	Configurable register to set the maximum outstanding read transactions for the interface	RW	0x0

16.12.20 ASNI awqosot register

Registers used to configure and store the write the maximum number of outstanding write transactions for the interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x9C

Type

RW

Reset value

0x00000000

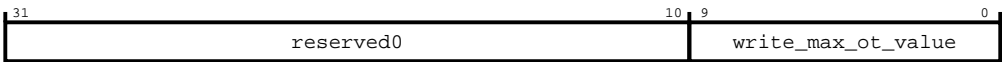
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the awqosot register bit assignments.

Figure 16-143: Bit assignment diagram for the awqosot register



The following table shows the awqosot register bit descriptions.

Table 16-155: awqosot bit descriptions

Bits	Name	Description	Type	Reset
[31:10]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[9:0]	write_max_ot_value	Configurable register to set the maximum outstanding write transactions for the interface	RW	0x0

16.12.21 ASNI axqosot register

Registers used to configure and store the maximum number of outstanding read and write transactions for the interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xA0

Type

RW

Reset value

0x00000000

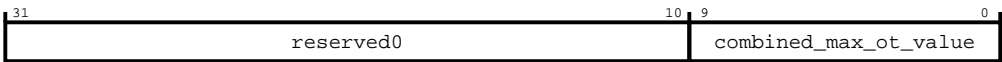
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the axqosot register bit assignments.

Figure 16-144: Bit assignment diagram for the axqosot register



The following table shows the axqosot register bit descriptions.

Table 16-156: axqosot bit descriptions

Bits	Name	Description	Type	Reset
[31:10]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[9:0]	combined_max_ot_value	Configurable register to set the maximum outstanding read and write transactions for the interface	RW	0x0

16.12.22 ASNI qosrdpk register

This register controls the QoS peak rate for the read hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xA4

Type

RW

Reset value

0x00000000

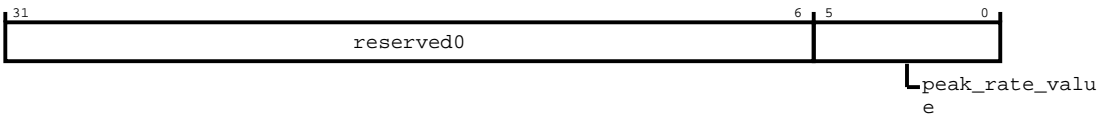
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qosrdpk register bit assignments.

Figure 16-145: Bit assignment diagram for the qosrdpk register



The following table shows the qosrdpk register bit descriptions.

Table 16-157: qosrdpk bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	peak_rate_value	Read channel peak rate value. The value is a binary fraction of the peak number of read transfers for each cycle.	RW	0b000000

16.12.23 ASNI qosrdbur register

This register controls the QoS burstiness for the read hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xA8

Type

RW

Reset value

0x00000000

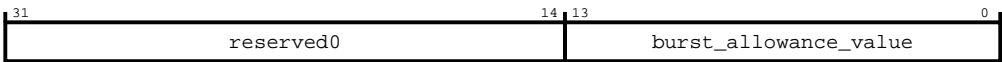
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qosrdbur register bit assignments.

Figure 16-146: Bit assignment diagram for the qosrdbur register



The following table shows the qosrdbur register bit descriptions.

Table 16-158: qosrdbur bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[13:0]	burst_allowance_value	Read channel QoS burstiness allowance value. The value is the number of read transfers that is permitted in a transaction.	RW	0x0

16.12.24 ASNI qosrdavg register

This register controls the QoS average rate for the read hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xAC

Type

RW

Reset value

0x00000000



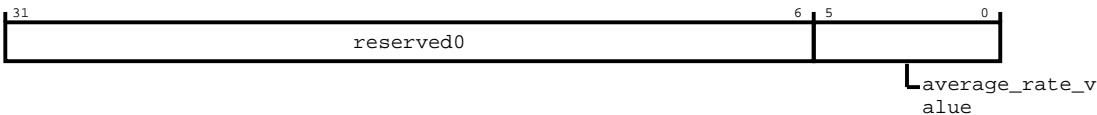
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qosrdavg register bit assignments.

Figure 16-147: Bit assignment diagram for the qosrdavg register



The following table shows the qosrdavg register bit descriptions.

Table 16-159: qosrdavg bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	average_rate_value	Read channel QoS average rate value. The value is a binary fraction of the average number of read transfers for each cycle.	RW	0b000000

16.12.25 ASNI qoswrpk register

This register controls the QoS peak rate for the write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xB0

Type

RW

Reset value

0x00000000

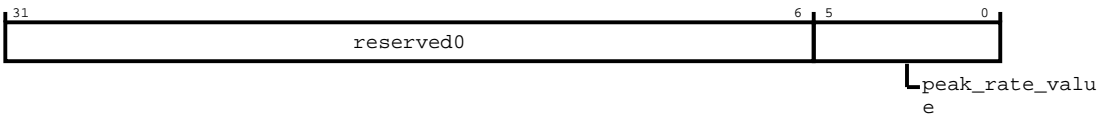
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoswrpk register bit assignments.

Figure 16-148: Bit assignment diagram for the qoswrpk register



The following table shows the qoswrpk register bit descriptions.

Table 16-160: qoswrpk bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	peak_rate_value	Write channel peak rate value. The value is a binary fraction of the peak number of write transfers for each cycle.	RW	0b000000

16.12.26 ASNI qoswrbur register

This register controls the QoS burstiness for the write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xB4

Type

RW

Reset value

0x00000000

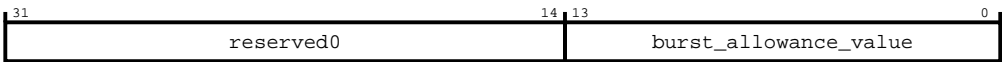
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoswrbur register bit assignments.

Figure 16-149: Bit assignment diagram for the qoswrbur register



The following table shows the qoswrbur register bit descriptions.

Table 16-161: qoswrbur bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[13:0]	burst_allowance_value	Write channel QoS burstiness allowance value. The value is the number of write transfers that are permitted in a transaction.	RW	0x0

16.12.27 ASNI qoswragv register

This register controls the QoS average rate for the write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xB8

Type

RW

Reset value

0x00000000

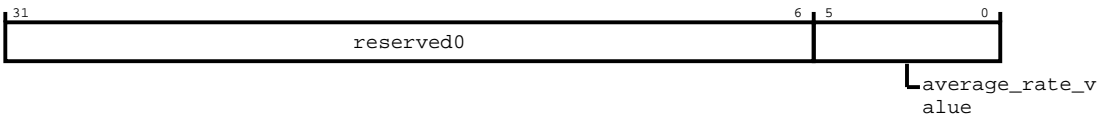
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoswavg register bit assignments.

Figure 16-150: Bit assignment diagram for the qoswavg register



The following table shows the qoswavg register bit descriptions.

Table 16-162: qoswavg bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	average_rate_value	Write channel QoS average rate value. The value is a binary fraction of the average number of write transfers for each cycle.	RW	0b000000

16.12.28 ASNI qoscompk register

This register controls the QoS peak rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xBC

Type

RW

Reset value

0x00000000

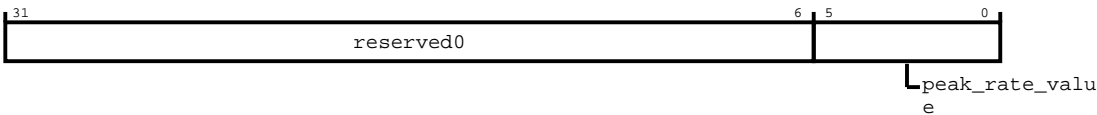
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscompk register bit assignments.

Figure 16-151: Bit assignment diagram for the qoscompk register



The following table shows the qoscompk register bit descriptions.

Table 16-163: qoscompk bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	peak_rate_value	The QoS peak rate value of both read and write channels. The value is a binary fraction of the peak number of both read and write transfers for each cycle.	RW	0b000000

16.12.29 ASNI qoscombur register

This register controls the QoS burstiness allowance for combined read and write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC0

Type

RW

Reset value

0x00000000

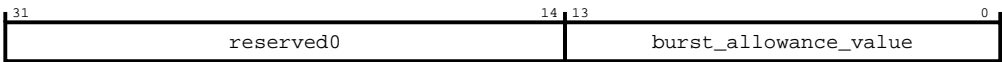
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscombur register bit assignments.

Figure 16-152: Bit assignment diagram for the qoscombur register



The following table shows the qoscombur register bit descriptions.

Table 16-164: qoscombur bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[13:0]	burst_allowance_value	Specifies the combined read and write TSPEC burstiness allowance	RW	0x0

16.12.30 ASNI qoscomavg register

This register controls the QoS average rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC4

Type

RW

Reset value

0x00000000

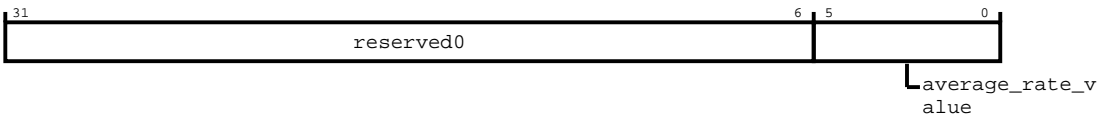
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscomavg register bit assignments.

Figure 16-153: Bit assignment diagram for the qoscomavg register



The following table shows the qoscomavg register bit descriptions.

Table 16-165: qoscomavg bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	average_rate_value	The QoS average rate value of both read and write channels. The value is a binary fraction of the average number of both read and write transfers for each cycle.	RW	0b000000

16.12.31 ASNI qosrdbqv register

This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for read soft bandwidth regulation, BQV, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC8

Type

RW

Reset value

0x00000000

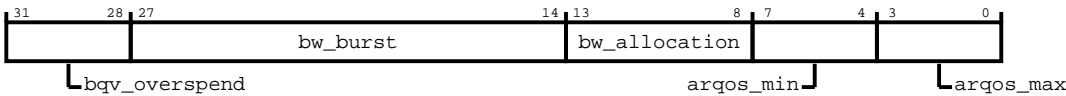
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qosrdbqv register bit assignments.

Figure 16-154: Bit assignment diagram for the qosrdbqv register



The following table shows the qosrdbqv register bit descriptions.

Table 16-166: qosrdbqv bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	bqv_overspend	BQV overspend. The excess number of full data bus transfers permitted.	RW	0b0000
[27:14]	bw_burst	Bandwidth burstiness. The excess number of full data bus transfers to permit as burstiness allowance.	RW	0x0
[13:8]	bw_allocation	Bandwidth allocation. The proportion of data bus width for bandwidth allocation.	RW	0b000000
[7:4]	arqos_min	BQV minimum QoS value. The minimum value of ARQOS.	RW	0b0000
[3:0]	arqos_max	BQV maximum QoS value. The maximum value of ARQOS.	RW	0b0000

16.12.32 ASNI qoswrbqv register

This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for write soft bandwidth regulation, BQV, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xCC

Type

RW



Reset value

0x00000000

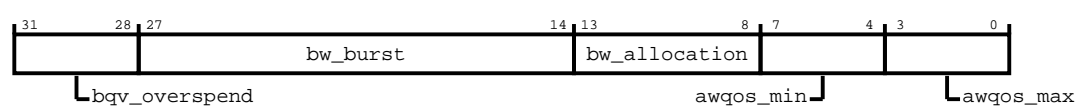
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoswrbqv register bit assignments.

Figure 16-155: Bit assignment diagram for the qoswrbqv register



The following table shows the qoswrbqv register bit descriptions.

Table 16-167: qoswrbqv bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	bqv_overspend	BQV overspend. The excess number of full data bus transfers permitted.	RW	0b0000
[27:14]	bw_burst	Bandwidth burstiness. The excess number of full data bus transfers to permit as burstiness allowance.	RW	0x0
[13:8]	bw_allocation	Bandwidth allocation. The proportion of data bus width for bandwidth allocation.	RW	0b000000
[7:4]	awqos_min	BQV minimum QoS value. The minimum value of AWQOS.	RW	0b0000
[3:0]	awqos_max	BQV maximum QoS value. The maximum value of AWQOS.	RW	0b0000

16.12.33 ASNI qoscombqv register

This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for both read and write soft bandwidth regulation, BQV, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xD0

**Type**  
RW

**Reset value**  
0x00000000

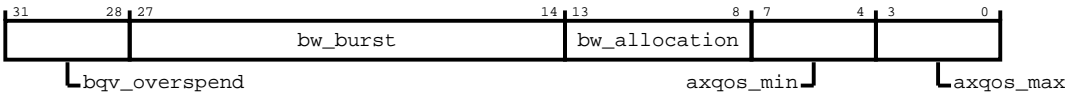
**Constraints**

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the qoscombqv register bit assignments.

**Figure 16-156: Bit assignment diagram for the qoscombqv register**



The following table shows the qoscombqv register bit descriptions.

**Table 16-168: qoscombqv bit descriptions**

Bits	Name	Description	Type	Reset
[31:28]	bqv_overspend	BQV overspend. The excess number of full data bus transfers permitted.	RW	0b0000
[27:14]	bw_burst	Bandwidth burstiness. The excess number of full data bus transfers to permit as burstiness allowance.	RW	0x0
[13:8]	bw_allocation	Bandwidth allocation. The proportion of data bus width for bandwidth allocation.	RW	0b000000
[7:4]	axqos_min	BQV minimum QoS value. The minimum value of AxQoS.	RW	0b0000
[3:0]	axqos_max	BQV maximum QoS value. The maximum value of AxQoS.	RW	0b0000

16.12.34 ASNI read\_channel\_mpam\_override register

This register controls the ASNI read channel MPAM override behavior.

**Configurations**

This register is only present if support for MPAM is enabled on the interface.

**Attributes**

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0xE0

**Type**  
RW

**Reset value**  
0x00000000

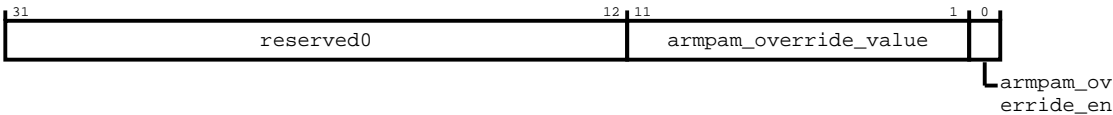
**Constraints**

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the read\_channel\_mpam\_override register bit assignments.

**Figure 16-157: Bit assignment diagram for the read\_channel\_mpam\_override register**



The following table shows the read\_channel\_mpam\_override register bit descriptions.

**Table 16-169: read\_channel\_mpam\_override bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[11:1]	armpam_override_value	ARMPAM override value	RW	0x0
[0]	armpam_override_en	When set, the ARMPAM value driven into the interconnect is driven from the MPAM override value in this register. If MPAM_SUPPORT = 0 for this specific interface, but GT_MPAM_SUPPORT is enabled, then this register drives the ARMPAM values for this ASNI irrespective of the value of the override bit.	RW	0

16.12.35 ASNI write\_channel\_mpam\_override register

This register controls the ASNI write channel MPAM override behavior.

**Configurations**

This register is only present if support for MPAM is enabled on the interface.

**Attributes**

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0xE4

**Type**  
RW

**Reset value**  
0x00000000

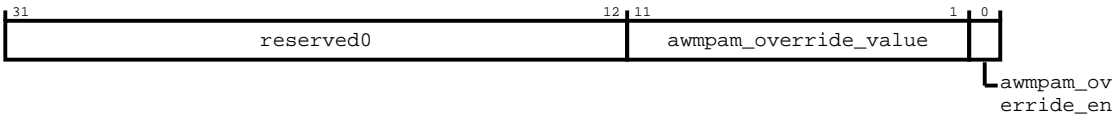
**Constraints**

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the write\_channel\_mpam\_override register bit assignments.

**Figure 16-158: Bit assignment diagram for the write\_channel\_mpam\_override register**



The following table shows the write\_channel\_mpam\_override register bit descriptions.

**Table 16-170: write\_channel\_mpam\_override bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[11:1]	awmpam_override_value	AWMPAM override value	RW	0x0
[0]	awmpam_override_en	When set, the AWMPAM value driven into the interconnect is driven from the MPAM override value in this register. If MPAM_SUPPORT = 0 for this specific interface, but GT_MPAM_SUPPORT is enabled, then this register drives the AWMPAM values for this ASNI irrespective of the value of the override bit.	RW	0

16.12.36 ASNI idm\_device\_id register

This register indicates the statically configured device ID value and is implemented if IDM is enabled.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**  
32-bit

**Address offset**

0x100

**Type**

RO

**Reset value**

See individual bit resets.

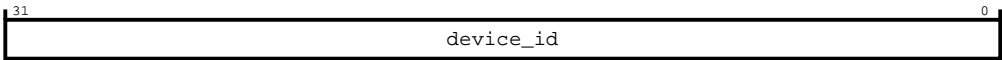
**Constraints**

None.

**Bit descriptions**

The following figure shows the `idm_device_id` register bit assignments.

**Figure 16-159: Bit assignment diagram for the `idm_device_id` register**



The following table shows the `idm_device_id` register bit descriptions.

**Table 16-171: `idm_device_id` bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	device_id	Returns statically configured ID value	RO	Configuration dependent

**16.12.37 ASNI `idm_config` register**

This register enables transaction logging, error detection, timeout detection, access control, and reset control.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**

32-bit

**Address offset**

0x104

**Type**

RW

## Reset value

See individual bit resets.

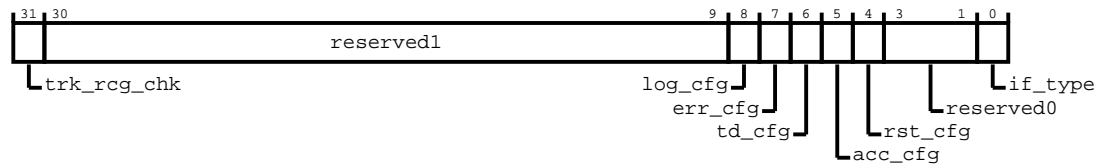
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_config` register bit assignments.

**Figure 16-160: Bit assignment diagram for the idm\_config register**



The following table shows the `idm_config` register bit descriptions.

### Table 16-172: idm\_config bit descriptions

Bits	Name	Description	Type	Reset
[31]	trk_rcg_chk	Tracker Regional Clock Gating (RCG) chicken bit	RW	0
[30:9]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[8]	log_cfg	Transaction logging present	RO	1
[7]	err_cfg	Error detection present	RO	1
[6]	td_cfg	Timeout detection present	RO	1
[5]	acc_cfg	Access control present	RO	1
[4]	rst_cfg	Reset control present	RO	1
[3:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[0]	if_type	Interface type  <div> <div>0</div> <div>Completer</div> </div> <div> <div>1</div> <div>Requester</div> </div>	RO	Configuration dependent

### 16.12.38 ASNI idm\_errctlr register

This register controls how errors are handled.

## Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x108

Type

RW

Reset value

0x00000000

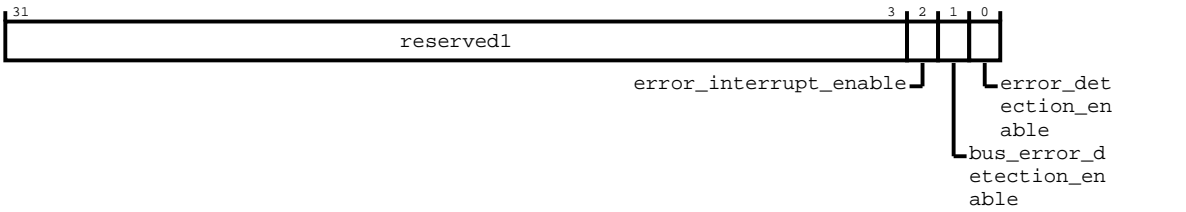
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_errctlr register bit assignments.

Figure 16-161: Bit assignment diagram for the idm\_errctlr register



The following table shows the idm\_errctlr register bit descriptions.

Table 16-173: idm\_errctlr bit descriptions

Bits	Name	Description	Type	Reset
[31:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[2]	error_interrupt_enable	Enable error interrupt for uncorrected error as indicated by IDM_ERRSTATUS.UE fields	RW	0
[1]	bus_error_detection_enable	Enable bus error detection  0 Disabled  1 Enabled when an error is detected and idm_errctlr [ed] is enabled. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set. An error interrupt event is generated (unless masked).	RW	0

Bits	Name	Description	Type	Reset
[0]	error_detection_enable	<p>Error detection global enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when an error is detected. In other words, a timeout error or bus error is detected and its respective detection enable register bit, Timeout_control[TD_EN], or idm_errctrlr[be] is also set. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>An error interrupt event is generated, unless masked.</p>	RW	0

### 16.12.39 ASNI idm\_errstatus register

This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x110

##### Type

RW

##### Reset value

0x00000000

#### Constraints

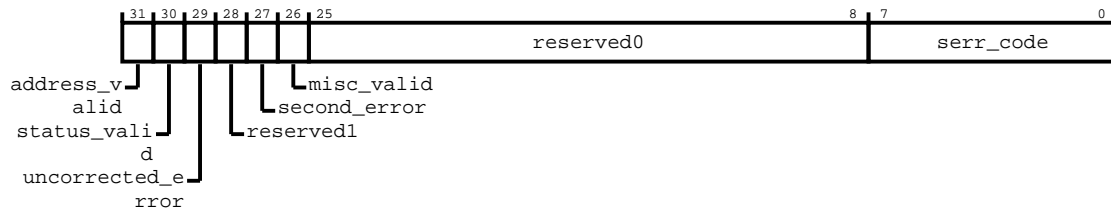
Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_errstatus register bit assignments.



**Figure 16-162: Bit assignment diagram for the idm\_errstatus register**



The following table shows the idm\_errstatus register bit descriptions.

**Table 16-174: idm\_errstatus bit descriptions**

Bits	Name	Description	Type	Reset
[31]	address_valid	<p>Address valid. The values are:</p> <p><b>0</b></p> <p>ERRADDR is not valid.</p> <p><b>1</b></p> <p>ERRADDR contains an address that is associated with the highest priority error which this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1 and is not cleared to zero in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[30]	status_valid	<p>Status register is valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRSTATUS not valid</p> <p><b>1</b></p> <p>IDM_ERRSTATUS valid. At least one error has been recorded.</p> <p>This bit ignores writes if any of the following fields is set to 1 and is not being cleared to zero in the same write:</p> <ul style="list-style-type: none"> <li>IDM_ERRSTATUS.UE</li> <li>IDM_ERRSTATUS.AV</li> <li>IDM_ERRSTATUS.OF * IDM_ERRSTATUS.MV</li> </ul> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred</p> <p>This bit ignores writes if IDM_ERRSTATUS.OF is set to 1 and is not being cleared to zero in the same write. This bit is not valid and reads <b>UNKNOWN</b> if IDM_ERRSTATUS.V is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISC0 and IDM_ERRMISC1 not valid</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISC0 and IDM_ERRMISC1 registers contains additional information for an error that this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1, and is not being cleared to 0 in the same write. This bit is a read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	<p>Primary error code. Indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0

16.12.40 ASNI idm\_erraddr\_lsb register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x114

Type

RO

Reset value

0x00000000

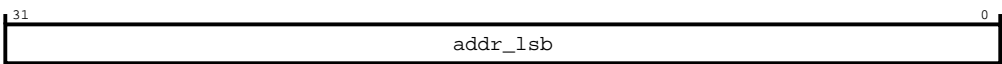
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb register bit assignments.

Figure 16-163: Bit assignment diagram for the idm\_erraddr\_lsb register



The following table shows the idm\_erraddr\_lsb register bit descriptions.

Table 16-175: idm\_erraddr\_lsb bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.12.41 ASNI idm\_erraddr\_msb register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x118

Type

RO

Reset value

0x00000000

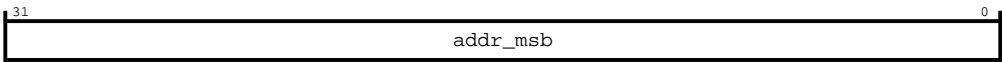
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_msb` register bit assignments.

Figure 16-164: Bit assignment diagram for the `idm_erraddr_msb` register



The following table shows the `idm_erraddr_msb` register bit descriptions.

Table 16-176: `idm_erraddr_msb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.12.42 ASNI `idm_errmisc0` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x128

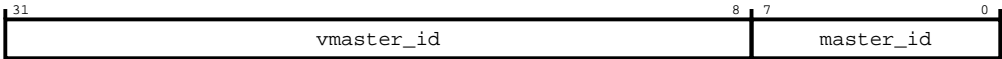
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions.

**Bit descriptions**  
The following figure shows the idm\_errmisc0 register bit assignments.

**Figure 16-165: Bit assignment diagram for the idm\_errmisc0 register**



The following table shows the idm\_errmisc0 register bit descriptions.

**Table 16-177: idm\_errmisc0 bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming AXI AxID into ASNI of the transaction causing an error. The assumption here is there is no manipulation of incoming AXI AxID in ASNI.	RO	0x0
[7:0]	master_id	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.12.43 ASNI idm\_errmisc1 register

This register is the error log of Secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x12C

**Type**  
RO

**Reset value**  
0x00000000

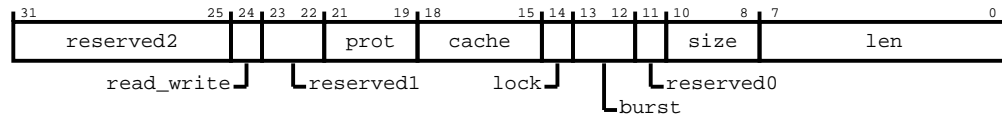
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_errmisc1` register bit assignments.

**Figure 16-166: Bit assignment diagram for the `idm_errmisc1` register**



The following table shows the `idm_errmisc1` register bit descriptions.

**Table 16-178: `idm_errmisc1` bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	The AXI read or write information of a transaction causing an error  1 Write  0 Read	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	The AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	The AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	The AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	The AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	The AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	The AXI len information of a transaction causing an error.	RO	0x0

## 16.12.44 ASNI `idm_access_control` register

This register controls the state, gated or ungated, of a device.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

### Width

32-bit

Address offset

0x130

Type

RW

Reset value

0x00000000

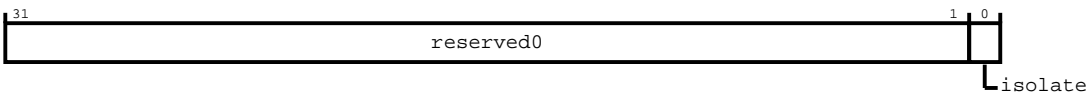
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_access\_control register bit assignments.

Figure 16-167: Bit assignment diagram for the idm\_access\_control register



The following table shows the idm\_access\_control register bit descriptions.

Table 16-179: idm\_access\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	isolate	Perform gating off a device. Reading 1 indicates that the completer device is gated or isolated. Reading 0 indicates that the completer device is ungated or de-isolated. Write 1 to enter gated state. Write 0 to exit gated state. There is some delay to updating this field with the intended write value. Exit from gated state is only successful if there are no outstanding transactions and all error status register bits are cleared. Entry into gated state is only successful if there are no outstanding transactions. While in pending isolation entry state or in active isolation state, a write of 1 to this bit causes reentry to isolation state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 0 is ignored. While in pending isolation exit state, a write of 0 to this bit causes a re-exit to the exit state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS, and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 1 is ignored.	RW	0

16.12.45 ASNI idm\_access\_status register

This register indicates the access status for Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x134

Type

RO

Reset value

0x00000002

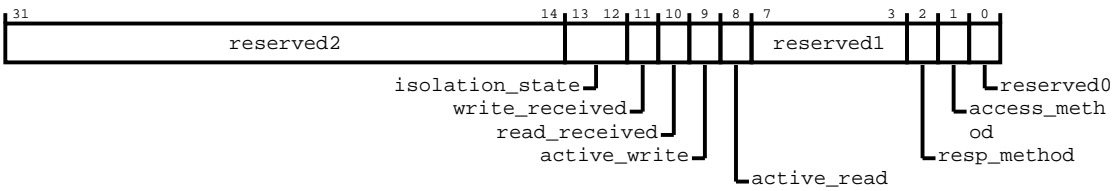
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_access\_status register bit assignments.

Figure 16-168: Bit assignment diagram for the idm\_access\_status register



The following table shows the idm\_access\_status register bit descriptions.

Table 16-180: idm\_access\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved2	Bits within this register segment are reserved for future product development	RO	0x0
[13:12]	isolation_state	Isolation status:  00 Isolation exit or entry is successful or not in gated or isolation state  01 Isolation exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  10 Isolation entry is unsuccessful or pending because of outstanding transactions  11 Reserved	RO	0b00



Bits	Name	Description	Type	Reset
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00000
[2]	resp_method	Indicates device generates errors in gated access	RO	0
[1]	access_method	Wait for all outstanding to complete, then block input	RO	1
[0]	reserved0	Bits within this register segment are reserved for future product development	RO	0

## 16.12.46 ASNI idm\_access\_readid register

This register is the access log of Secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x138

#### Type

RO

#### Reset value

0x00000000

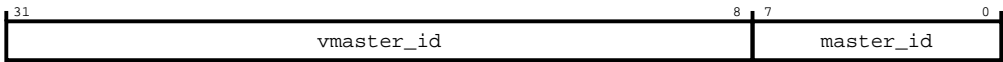
### Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_access_readid` register bit assignments.

Figure 16-169: Bit assignment diagram for the `idm_access_readid` register



The following table shows the `idm_access_readid` register bit descriptions.

Table 16-181: `idm_access_readid` bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

16.12.47 ASNI `idm_access_writeid` register

This register is the access log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x13C

Type

RO

Reset value

0x00000000

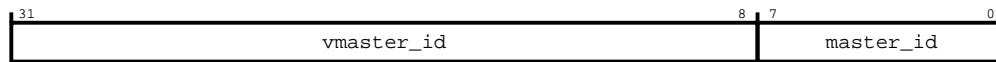
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_access_writeid` register bit assignments.

**Figure 16-170: Bit assignment diagram for the `idm_access_writeid` register**



The following table shows the `idm_access_writeid` register bit descriptions.

**Table 16-182: `idm_access_writeid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AWID signal into the endpoint of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

## 16.12.48 ASNI `idm_reset_control` register

This register controls the reset of a device that is attached to the interconnect.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x140

#### Type

RW

#### Reset value

0x00000002

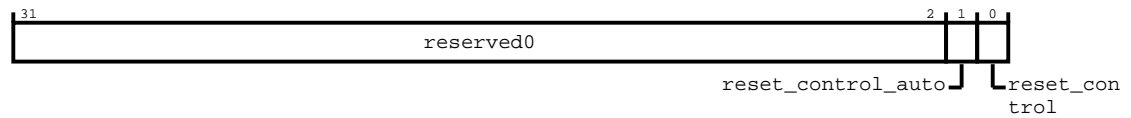
### Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the `idm_reset_control` register bit assignments.

**Figure 16-171: Bit assignment diagram for the `idm_reset_control` register**



The following table shows the `idm_reset_control` register bit descriptions.

### Table 16-183: idm\_reset\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	reset_control_auto	<p>Configures the device for auto or internal reset mode. For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. There are several constraints on this field:</p> <ul style="list-style-type: none"> <li>You can only change this field during initialization or when the interface is fully quiesced. * Arm does not support changing this field while the interface is active. If you change this field during runtime, behavior is <b>UNPREDICTABLE</b>.</li> </ul> <p>Reads have the following effect:</p> <p><b>1</b></p> <p>A read of 1 indicates that the device is in auto or internal reset mode.</p> <p><b>0</b></p> <p>A read of 0 indicates that the device is not in auto or internal reset mode.</p> <p>Writes have the following effect:</p> <p><b>1</b></p> <p>A write of 1 configures the device for auto or internal reset mode.</p> <p><b>0</b></p> <p>A write of 0 disables auto or internal reset mode.</p> <p>For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. Bit[1] of the IDM_RESET_CONTROL register is 1 out of reset. This bit enables internal recovery mode out of reset. When not in auto reset mode and a timeout is detected, a write of 1 to the IDM_RESET_CONTROL.reset field initiates internal recovery mode. Changing this bit while the interface is not in idle mode results in <b>UNPREDICTABLE</b> behavior.</p>	RO	1

Bits	Name	Description	Type	Reset
[0]	reset_control	<p>Performs soft reset of attached device. If the auto bit is set to 1 the network interface gates the external interface, however the soft reset pin is not activated. If the auto bit is 0, the interfaces are not gated until there is a write to bit[0]. In this case, the soft reset pin is activated. Writes have the following effect:</p> <p><b>1</b></p> <p>Request the attached device to enter reset. If the write occurs before soft reset exit has occurred, the write is ignored.</p> <p><b>0</b></p> <p>Request the attached device to exit reset. If the write occurs before soft reset entry has occurred, the write is ignored.</p> <p>Software polls this register to determine if soft reset entry or exit has occurred, using the following values:</p> <p><b>1</b></p> <p>Indicates that the device is in reset.</p> <p><b>0</b></p> <p>Indicates that the device is not in reset.</p> <p>This register value updates to reflect a request for reset entry or reset exit, but the update can only occur after required internal conditions are met. Until these conditions are met, a read to this register returns the old value. For example, outstanding transactions currently being handled must complete before this register value updates. To ensure reset propagation within the device, it is the responsibility of the software to permit enough cycles after soft reset assertion is reflected in the IDM_RESET_CONTROL register before exiting soft reset by triggering a write of 0. If this responsibility is not met, the behavior is <b>UNDEFINED</b> or <b>UNPREDICTABLE</b>. When this register value is 1, the external soft reset pin that connects to the attached AXI requester or completer device is asserted, using the correct polarity of the reset pin. When this register value is 0, the external soft reset pin that connects to the attached AXI requester or completer device is deasserted, using the correct polarity of the reset pin. When in pending soft reset entry state or in active soft reset state, a write of 1 to this bit causes reentry to soft reset state. This write causes the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers to be cleared. A write of 0 is ignored. While in pending soft reset exit state, a write of 0 to this bit causes re-exit to exit state. A write of 0 also clears the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers. A write of 1 is ignored.</p>	RW	0

### 16.12.49 ASNI idm\_reset\_status register

This register indicates mostly the reset status of Secure transactions. However, the rst\_exit\_state field indicates reset exit state of secure or non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

Address offset

0x144

Type

RO

Reset value

0x00000000

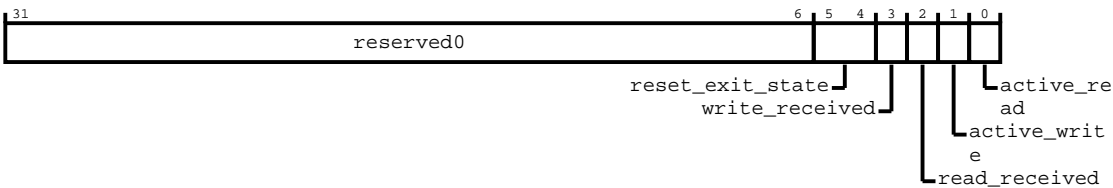
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_reset\_status register bit assignments.

Figure 16-172: Bit assignment diagram for the idm\_reset\_status register



The following table shows the idm\_reset\_status register bit descriptions.

Table 16-184: idm\_reset\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:4]	reset_exit_state	Reset exit state  00 Reset exit or entry is successful or not in reset state  01 Reset exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  10 Reset exit is unsuccessful or pending because of outstanding transactions  11 Reset exit is unsuccessful or pending because of both uncleared error status bits and outstanding transactions	RO	0b00
[3]	write_received	A 1 indicates that an active Secure write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"><li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li><li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li></ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0

### 16.12.50 ASNI idm\_reset\_readid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x148

##### Type

RO

##### Reset value

0x00000000

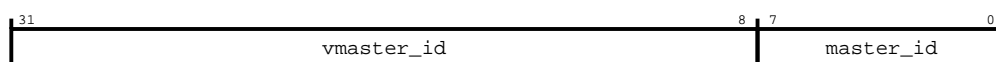
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_readid register bit assignments.

**Figure 16-173: Bit assignment diagram for the idm\_reset\_readid register**



The following table shows the idm\_reset\_readid register bit descriptions.

**Table 16-185: idm\_reset\_readid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.12.51 ASNI idm\_reset\_writeid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x14C

##### Type

RO

##### Reset value

0x00000000

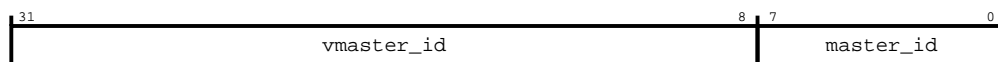
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid register bit assignments.

**Figure 16-174: Bit assignment diagram for the idm\_reset\_writeid register**



The following table shows the idm\_reset\_writeid register bit descriptions.



**Table 16-186: idm\_reset\_writeid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

## 16.12.52 ASNI idm\_timeout\_control register

This register is present when timeout detection is configured.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x150

#### Type

RW

#### Reset value

0x00000000

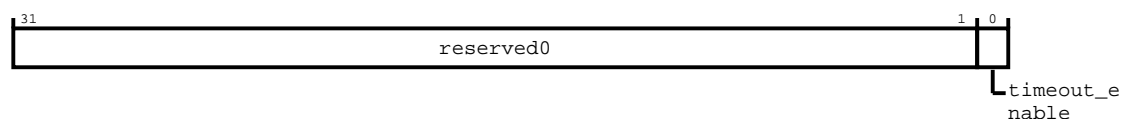
### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the idm\_timeout\_control register bit assignments.

**Figure 16-175: Bit assignment diagram for the idm\_timeout\_control register**



The following table shows the idm\_timeout\_control register bit descriptions.

**Table 16-187: idm\_timeout\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	timeout_enable	<p>Timeout detection enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when a timeout is detected. The timeout is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>A timeout interrupt event is generated, unless it is masked.</p>	RW	0

### 16.12.53 ASNI idm\_timeout\_value register

This register controls the duration that is used to determine if a transaction has timed out.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x154

##### Type

RW

##### Reset value

0x00000004

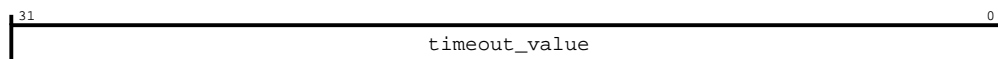
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the idm\_timeout\_value register bit assignments.

**Figure 16-176: Bit assignment diagram for the idm\_timeout\_value register**



The following table shows the `idm_timeout_value` register bit descriptions.

**Table 16-188: `idm_timeout_value` bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	<code>timeout_value</code>	Controls the duration that is used to determine if a transaction has timed out. The actual duration is $2^{\text{timeout\_exponent}}$ cycles. The minimum value is 4. Values of 0, 1, 2, or 3 are treated as 4. The maximum value is 30. Values greater than 30 are treated as 30.	RW	0x4

## 16.12.54 ASNI `idm_interrupt_status` register

This register indicates the interrupt status of Secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x158

#### Type

RW

#### Reset value

0x00000000

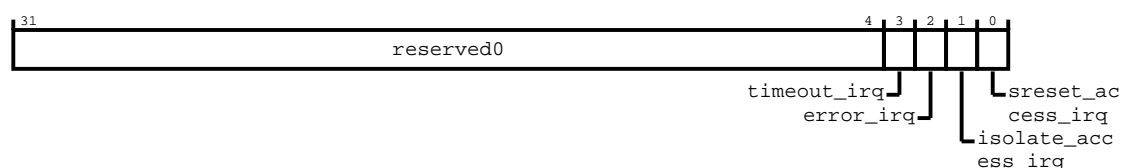
### Constraints

Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the `idm_interrupt_status` register bit assignments.

**Figure 16-177: Bit assignment diagram for the `idm_interrupt_status` register**



The following table shows the `idm_interrupt_status` register bit descriptions.

**Table 16-189: idm\_interrupt\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.12.55 ASNI idm\_interrupt\_mask register

This register is the interrupt mask of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x15C

##### Type

RW

##### Reset value

0x00000000

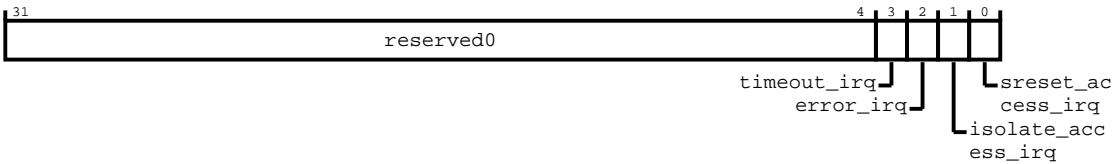
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask register bit assignments.

Figure 16-178: Bit assignment diagram for the `idm_interrupt_mask` register



The following table shows the `idm_interrupt_mask` register bit descriptions.

Table 16-190: `idm_interrupt_mask` bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

16.12.56 ASNI `idm_errstatus_ns` register

This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x160

Type

RW

Reset value

0x00000000

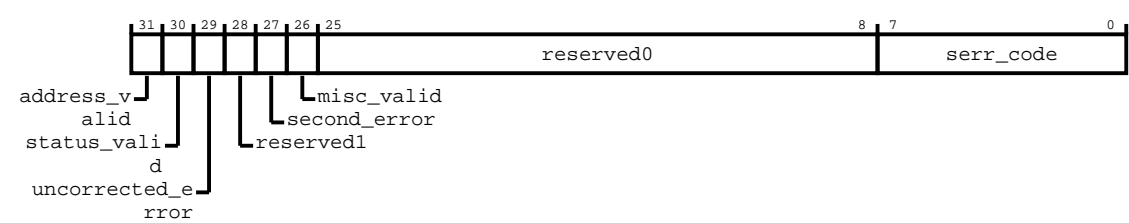
Constraints

None.

Bit descriptions

The following figure shows the `idm_errstatus_ns` register bit assignments.

Figure 16-179: Bit assignment diagram for the `idm_errstatus_ns` register



The following table shows the `idm_errstatus_ns` register bit descriptions.

Table 16-191: `idm_errstatus_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31]	<code>address_valid</code>	Address valid. The values are: <b>0</b> ERRADDR is not valid. <b>1</b> ERRADDR contains an address that is associated with the highest priority error that this record captures.  This bit ignores writes if the <code>ue</code> field of the <code>IDM_ERRSTATUS_NS</code> register is set to 1 and is not cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0
[30]	<code>status_valid</code>	Status register valid. The values are: <b>0</b> <code>IDM_ERRSTATUS_NS</code> is not valid. <b>1</b> <code>IDM_ERRSTATUS_NS</code> is valid. At least one error has been recorded.  This bit ignores writes if the <code>ue</code> field of the <code>IDM_ERRSTATUS_NS</code> register is set to 1 and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred.</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred.</p> <p>This bit ignores writes if the oe field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is not valid and reads <b>UNKNOWN</b> if the v field of the IDM_ERRSTATUS_NS register is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISCO_NS and IDM_ERRMISC1_NS are not valid.</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISCO_NS and IDM_ERRMISC1_NS registers contains additional information for an error that this record captures.</p> <p>This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1, and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	<p>Primary error code, indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0

16.12.57 ASNI idm\_erraddr\_lsb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x164

Type

RO

Reset value

0x00000000

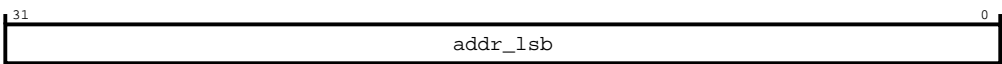
Constraints

None.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb\_ns register bit assignments.

Figure 16-180: Bit assignment diagram for the idm\_erraddr\_lsb\_ns register



The following table shows the idm\_erraddr\_lsb\_ns register bit descriptions.

Table 16-192: idm\_erraddr\_lsb\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.12.58 ASNI idm\_erraddr\_msb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.



Attributes

Its characteristics are:

Width

32-bit

Address offset

0x168

Type

RO

Reset value

0x00000000

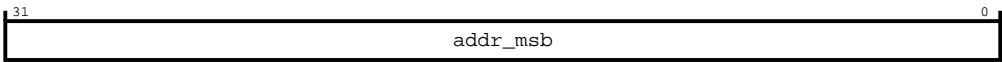
Constraints

None.

Bit descriptions

The following figure shows the `idm_erraddr_msb_ns` register bit assignments.

Figure 16-181: Bit assignment diagram for the `idm_erraddr_msb_ns` register



The following table shows the `idm_erraddr_msb_ns` register bit descriptions.

Table 16-193: `idm_erraddr_msb_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.12.59 ASNI `idm_errmisc0_ns` register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x178

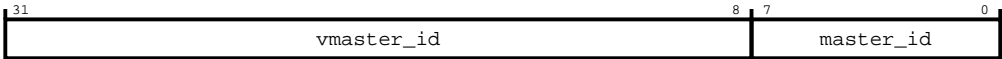
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the idm\_errmisc0\_ns register bit assignments.

**Figure 16-182: Bit assignment diagram for the idm\_errmisc0\_ns register**



The following table shows the idm\_errmisc0\_ns register bit descriptions.

**Table 16-194: idm\_errmisc0\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming AXI AxID into ASNI of the transaction causing an error. The assumption is no manipulation of incoming AXI AxID in ASNI.	RO	0x0
[7:0]	master_id	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.12.60 ASNI idm\_errmisc1\_ns register

This register is the error log of Non-secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x17C

**Type**  
RO

**Reset value**  
0x00000000

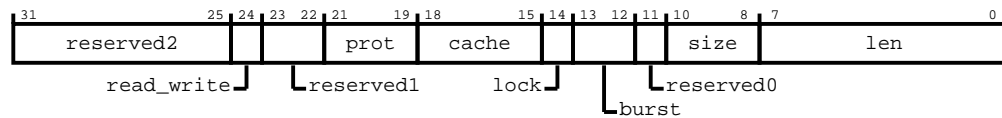
## Constraints

None.

## Bit descriptions

The following figure shows the idm\_errmisc1\_ns register bit assignments.

**Figure 16-183: Bit assignment diagram for the idm\_errmisc1\_ns register**



The following table shows the idm\_errmisc1\_ns register bit descriptions.

**Table 16-195: idm\_errmisc1\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	Returns the AXI read or write information of a transaction causing an error:  1 Write  0 Read	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	Returns the AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	Returns the AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	Returns the AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	Returns the AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	Returns the AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	Returns the AXI len information of a transaction causing an error.	RO	0x0

### 16.12.61 ASNI idm\_access\_status\_ns register

This register indicates the access status for Non-secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

### Width

32-bit

## Address offset

0x184

## Type

RO

## Reset value

0x00000000

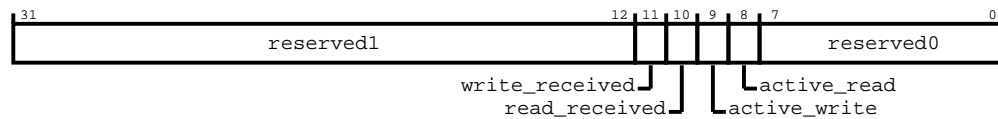
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_access_status_ns` register bit assignments.

**Figure 16-184: Bit assignment diagram for the `idm_access_status_ns` register**



The following table shows the `idm_access_status_ns` register bit descriptions.

**Table 16-196: `idm_access_status_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved1	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:0]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0

## 16.12.62 ASNI idm\_access\_readid\_ns register

This register is the access log of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x188

#### Type

RO

#### Reset value

0x00000000

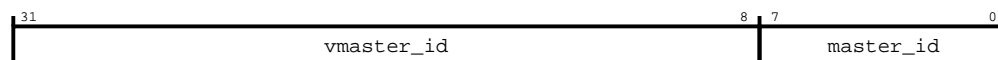
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_access\_readid\_ns register bit assignments.

**Figure 16-185: Bit assignment diagram for the idm\_access\_readid\_ns register**



The following table shows the idm\_access\_readid\_ns register bit descriptions.

**Table 16-197: idm\_access\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

### 16.12.63 ASNI idm\_access\_writeid\_ns register

This register is the access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x18C

##### Type

RO

##### Reset value

0x00000000

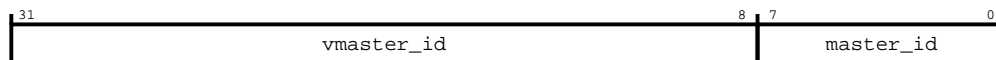
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_access\_writeid\_ns register bit assignments.

**Figure 16-186: Bit assignment diagram for the idm\_access\_writeid\_ns register**



The following table shows the idm\_access\_writeid\_ns register bit descriptions.

**Table 16-198: idm\_access\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the IDM_ACCESS_STATUS_NS register field active_write is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

## 16.12.64 ASNI idm\_reset\_status\_ns register

This register indicates the reset status of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x194

#### Type

RO

#### Reset value

0x00000000

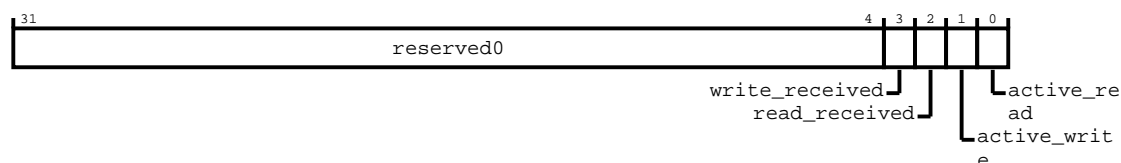
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_reset\_status\_ns register bit assignments.

**Figure 16-187: Bit assignment diagram for the idm\_reset\_status\_ns register**



The following table shows the idm\_reset\_status\_ns register bit descriptions.

**Table 16-199: idm\_reset\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[3]	write_received	<p>A 1 indicates that an active write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on:</p> <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to 0 on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates that there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates that there is at least one read transaction currently in progress.	RO	0

### 16.12.65 ASNI idm\_reset\_readid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x198

##### Type

RO

##### Reset value

0x00000000

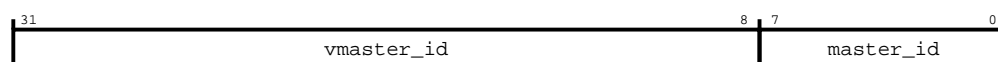
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_readid\_ns register bit assignments.

**Figure 16-188: Bit assignment diagram for the idm\_reset\_readid\_ns register**



The following table shows the idm\_reset\_readid\_ns register bit descriptions.



**Table 16-200: idm\_reset\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

### 16.12.66 ASNI idm\_reset\_writeid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x19C

##### Type

RO

##### Reset value

0x00000000

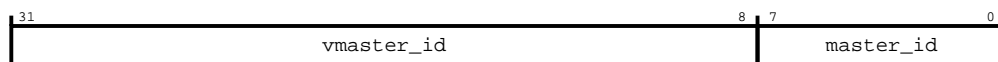
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid\_ns register bit assignments.

**Figure 16-189: Bit assignment diagram for the idm\_reset\_writeid\_ns register**



The following table shows the idm\_reset\_writeid\_ns register bit descriptions.

**Table 16-201: idm\_reset\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when active_write field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

## 16.12.67 ASNI idm\_interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1A8

#### Type

RW

#### Reset value

0x00000000

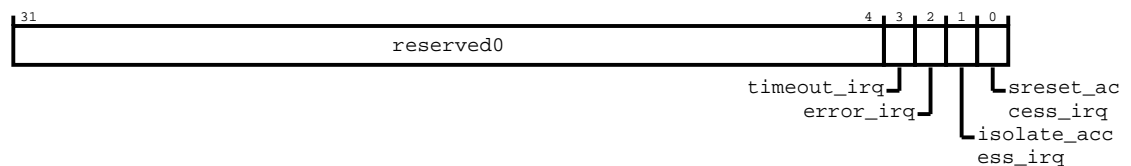
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_status\_ns register bit assignments.

**Figure 16-190: Bit assignment diagram for the idm\_interrupt\_status\_ns register**



The following table shows the idm\_interrupt\_status\_ns register bit descriptions.

**Table 16-202: idm\_interrupt\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

## 16.12.68 ASNI idm\_interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1AC

#### Type

RW

#### Reset value

0x00000000

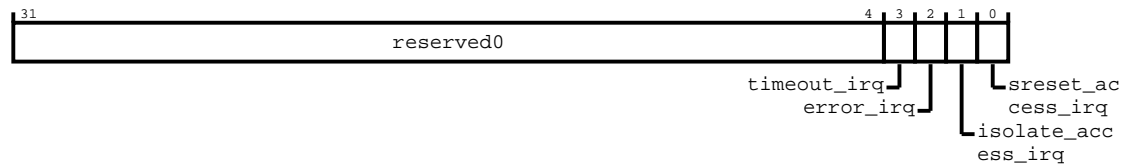
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_mask\_ns register bit assignments.

**Figure 16-191: Bit assignment diagram for the idm\_interrupt\_mask\_ns register**



The following table shows the idm\_interrupt\_mask\_ns register bit descriptions.

**Table 16-203: idm\_interrupt\_mask\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

## 16.13 AMNI register summary

This section describes the AMNI registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-204: AMNI register summary**

Offset	Name	Type	Reset	Width	Description
0x00	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies node type as AMBA requester network interface register.
0x04	<a href="#">node_info</a>	RO	See individual bit resets.	32-bit	Contains information about the AMNI node features and configuration.
0x08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	Contains register bits used for configuring the secure access behavior of the AMNI node.
0x0C	<a href="#">pmusela</a>	RW	0x00000000	32-bit	This register is used to select the event values in the AMNI event crossbar.
0x10	<a href="#">pmuselb</a>	RW	0x00000000	32-bit	This register is used to select the event values in the AMNI event crossbar.
0x14	<a href="#">interface_id_0_3</a>	RO	See individual bit resets.	32-bit	Contains information about the AMNI interface IDs for interfaces 0-3.
0x24	<a href="#">num_sub_features</a>	RO	See individual bit resets.	32-bit	Contains information about the number of sub features AMNI has.
0x28	<a href="#">sub_feature_0_type</a>	RO	See individual bit resets.	32-bit	Sub feature 0 type.
0x2C	<a href="#">sub_feature_0_pointer</a>	RO	See individual bit resets.	32-bit	Sub feature 0 pointer.

Offset	Name	Type	Reset	Width	Description
0x40	<a href="#">node_features</a>	RO	0x00000000	32-bit	Contains information about the supported features for this AMNI node.
0x80	<a href="#">silicon_debug</a>	RW	0x00000000	32-bit	This register monitors the status of NoC requester interface channels.
0x84	<a href="#">qos_accept_control</a>	RW	0x00000000	32-bit	Contains registers used for configuring QoS acceptance behavior for the AMNI node.
0x88	<a href="#">cmoovrd</a>	RW	0x00000000	32-bit	This register selects between SLVERR or OKAY responses to handle CMOs when downstream completers do not support them.
0x8C	<a href="#">rddata_agg_control</a>	RW	0x00000000	32-bit	This register contains controls for AMNI read data aggregation. Register value does not take effect if read_data_aggregation_enable in AMNI node_info is zero.
0xF0	<a href="#">interrupt_status</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transaction.
0xF4	<a href="#">interrupt_mask</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0xF8	<a href="#">interrupt_status_ns</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0xFC	<a href="#">interrupt_mask_ns</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.
0x100	<a href="#">idm_device_id</a>	RO	See individual bit resets.	32-bit	This register indicates the statically configured device ID value and is implemented if IDM is enabled.
0x104	<a href="#">idm_config</a>	RW	See individual bit resets.	32-bit	This register enables transaction logging, error detection, timeout detection, access control, and reset control.
0x108	<a href="#">idm_errctrl</a>	RW	0x00000000	32-bit	This register controls how errors are handled.
0x110	<a href="#">idm_errstatus</a>	RW	0x00000000	32-bit	This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.
0x114	<a href="#">idm_erraddr_lsb</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x118	<a href="#">idm_erraddr_msb</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x128	<a href="#">idm_errmisc0</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x12C	<a href="#">idm_errmisc1</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x130	<a href="#">idm_access_control</a>	RW	0x00000000	32-bit	This register controls the state, gated or ungated, of a device.
0x134	<a href="#">idm_access_status</a>	RO	0x00000002	32-bit	This register indicates the access status for Secure transactions.
0x138	<a href="#">idm_access_readid</a>	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x13C	<a href="#">idm_access_writeid</a>	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x140	<a href="#">idm_reset_control</a>	RW	0x00000002	32-bit	This register controls the reset of a device that is attached to the interconnect.
0x144	<a href="#">idm_reset_status</a>	RO	0x00000000	32-bit	This register indicates mostly the reset status of Secure transactions. However, the rst_exit_state field indicates reset exit state of secure or non-secure transactions.
0x148	<a href="#">idm_reset_readid</a>	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x14C	<a href="#">idm_reset_writeid</a>	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x150	<a href="#">idm_timeout_control</a>	RW	0x00000000	32-bit	This register is present when timeout detection is configured.
0x154	<a href="#">idm_timeout_value</a>	RW	0x00000004	32-bit	This register controls the duration that is used to determine if a transaction has timed out.
0x158	<a href="#">idm_interrupt_status</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0x15C	<a href="#">idm_interrupt_mask</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0x160	<a href="#">idm_errstatus_ns</a>	RW	0x00000000	32-bit	This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Offset	Name	Type	Reset	Width	Description
0x164	idm_erraddr_lsb_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x168	idm_erraddr_msb_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x178	idm_errmisc0_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x17C	idm_errmisc1_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x184	idm_access_status_ns	RO	0x00000000	32-bit	This register indicates the access status for Non-secure transactions.
0x188	idm_access_readid_ns	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x18C	idm_access_writeid_ns	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x194	idm_reset_status_ns	RO	0x00000000	32-bit	This register indicates the reset status of Non-secure transactions.
0x198	idm_reset_readid_ns	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x19C	idm_reset_writeid_ns	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x1A8	idm_interrupt_status_ns	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0x1AC	idm_interrupt_mask_ns	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.

### 16.13.1 AMNI node\_type register

This register identifies node type as AMBA requester network interface register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x00

##### Type

RO

##### Reset value

See individual bit resets.

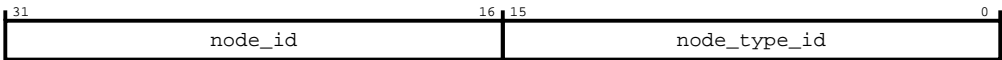
#### Constraints

None.

#### Bit descriptions

The following figure shows the node\_type register bit assignments.

Figure 16-192: Bit assignment diagram for the node\_type register



The following table shows the node\_type register bit descriptions.

Table 16-205: node\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	node_id	The AMNI ID assigned during network construction.	RO	Configuration dependent
[15:0]	node_type_id	The value of this field is 0x0005, and identifies the associated node_type as a requester interface for NoC AMNI registers.	RO	0x5

16.13.2 AMNI node\_info register

Contains information about the AMNI node features and configuration.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RO

Reset value

See individual bit resets.

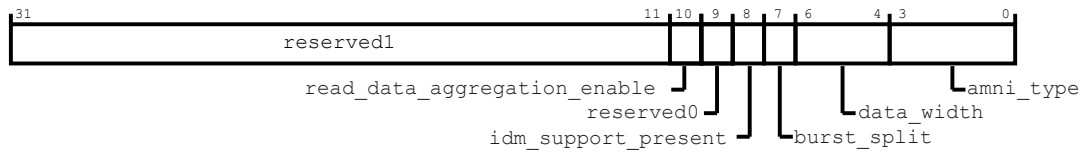
Constraints

None.

Bit descriptions

The following figure shows the node\_info register bit assignments.

**Figure 16-193: Bit assignment diagram for the node\_info register**



The following table shows the node\_info register bit descriptions.

**Table 16-206: node\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:11]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[10]	read_data_aggregation_enable	The value of this field specifies whether the AMNI support read data aggregation functionality.  0 read data aggregation not included 1 read data aggregation included	RO	Configuration dependent
[9]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[8]	idm_support_present	IDM support present 0 IDM support logic is not present 1 IDM support logic is present	RO	Configuration dependent
[7]	burst_split	Burst split present 0 Burst split logic is not present 1 Burst split logic is present	RO	Configuration dependent
[6:4]	data_width	Data width, AxSIZE encode 0b000 Reserved 0b001 Reserved 0b010 4 bytes 0b011 8 bytes 0b100 16 bytes 0b101 32 bytes 0b110 64 bytes 0b111 128 bytes	RO	Configuration dependent
[3:0]	amni_type	AMNI type 0b0000 Reserved 0b0001 AXI3 0b0010 AXI Issue F 0b0011 ACE-Lite 0b0100 AXI Issue G 0b0101 AXI Issue H 0b0110-0b1111 Reserved	RO	Configuration dependent

### 16.13.3 AMNI secure\_access register

Contains register bits used for configuring the secure access behavior of the AMNI node.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:



**Width**  
32-bit

**Address offset**  
0x08

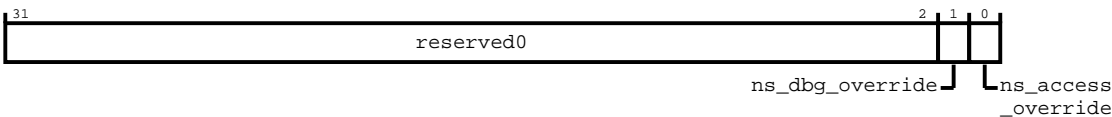
**Type**  
RW

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions.

**Bit descriptions**  
The following figure shows the secure\_access register bit assignments.

**Figure 16-194: Bit assignment diagram for the secure\_access register**



The following table shows the secure\_access register bit descriptions.

**Table 16-207: secure\_access bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	ns_dbg_override	Enables/Disables non-secure access to clock domain PMU and interface registers	RW	0
[0]	ns_access_override	Enables/Disables non-secure access to clock domain registers	RW	0

16.13.4 AMNI pmusela register

This register is used to select the event values in the AMNI event crossbar.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

## Address offset

0x0C

## Type

RW

## Reset value

0x00000000

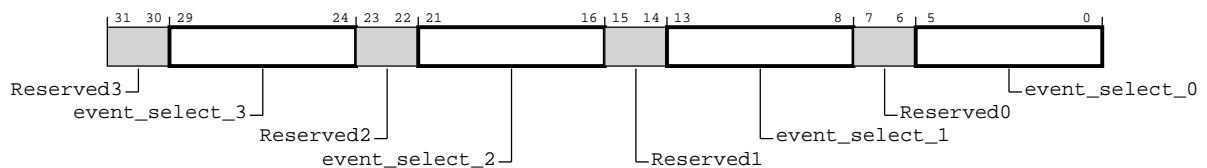
## Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmusela register bit assignments.

**Figure 16-195: Bit assignment diagram for the pmusela register**



The following table shows the pmusela register bit descriptions.

**Table 16-208: pmusela bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

## 16.13.5 AMNI pmusela register

This register is used to select the event values in the AMNI event crossbar.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

### Width

32-bit

### Address offset

0x10

### Type

RW

### Reset value

0x00000000

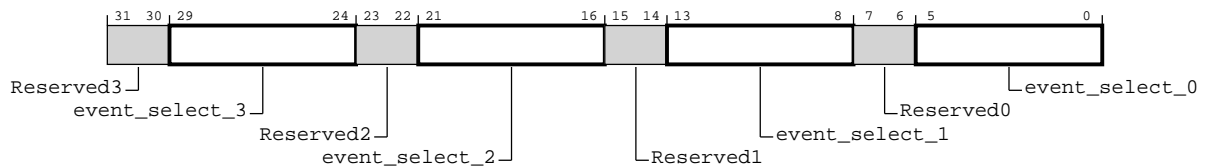
## Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmuselb register bit assignments.

**Figure 16-196: Bit assignment diagram for the pmuselb register**



The following table shows the pmuselb register bit descriptions.

**Table 16-209: pmuselb bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

16.13.6 AMNI interface\_id\_0\_3 register

Contains information about the AMNI interface IDs for interfaces 0-3.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x14

Type

RO

Reset value

See individual bit resets.

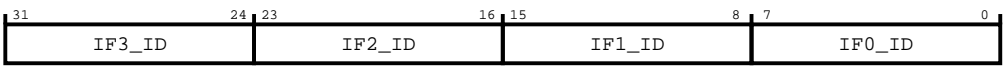
Constraints

None.

Bit descriptions

The following figure shows the interface\_id\_0\_3 register bit assignments.

Figure 16-197: Bit assignment diagram for the interface\_id\_0\_3 register



The following table shows the interface\_id\_0\_3 register bit descriptions.

Table 16-210: interface\_id\_0\_3 bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	IF3_ID	Reserved	RO	Configuration dependent
[23:16]	IF2_ID	Reserved	RO	Configuration dependent
[15:8]	IF1_ID	Reserved	RO	Configuration dependent
[7:0]	IF0_ID	AMNI interface ID 0	RO	Configuration dependent

16.13.7 AMNI num\_sub\_features register

Contains information about the number of sub features AMNI has.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x24

Type

RO

Reset value

See individual bit resets.

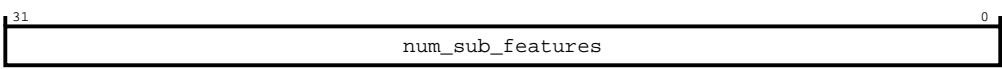
Constraints

None.

Bit descriptions

The following figure shows the num\_sub\_features register bit assignments.

Figure 16-198: Bit assignment diagram for the num\_sub\_features register



The following table shows the num\_sub\_features register bit descriptions.

Table 16-211: num\_sub\_features bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	num_sub_features	Number of sub features	RO	Configuration dependent

16.13.8 AMNI sub\_feature\_0\_type register

Sub feature 0 type.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x28

Type

RO

Reset value

See individual bit resets.

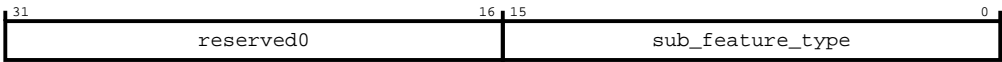
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_type register bit assignments.

Figure 16-199: Bit assignment diagram for the sub\_feature\_0\_type register



The following table shows the sub\_feature\_0\_type register bit descriptions.

Table 16-212: sub\_feature\_0\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[15:0]	sub_feature_type	Sub feature 0 type	RO	Configuration dependent

16.13.9 AMNI sub\_feature\_0\_pointer register

Sub feature 0 pointer.

Configurations

The number of registers of this type that are present depends on the number of subfeatures in the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x2C

Type

RO

Reset value

See individual bit resets.

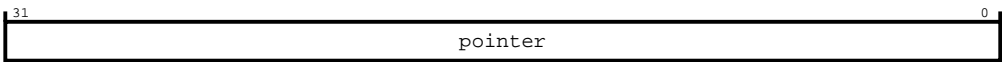
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_pointer register bit assignments.

Figure 16-200: Bit assignment diagram for the sub\_feature\_0\_pointer register



The following table shows the sub\_feature\_0\_pointer register bit descriptions.

Table 16-213: sub\_feature\_0\_pointer bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	pointer	Sub feature 0 pointer	RO	Configuration dependent

16.13.10 AMNI node\_features register

Contains information about the supported features for this AMNI node.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x40

Type

RO

Reset value

0x00000000

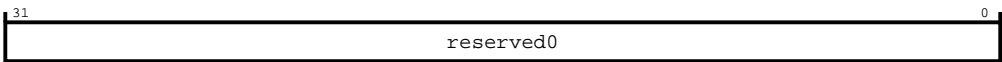
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the node\_features register bit assignments.

Figure 16-201: Bit assignment diagram for the node\_features register



The following table shows the node\_features register bit descriptions.

Table 16-214: node\_features bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0

16.13.11 AMNI silicon\_debug register

This register monitors the status of NoC requester interface channels.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x80

Type

RW

Reset value

0x00000000

Constraints

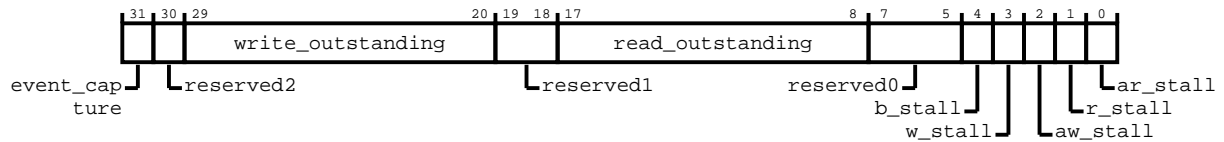
Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.



## Bit descriptions

The following figure shows the silicon\_debug register bit assignments.

**Figure 16-202: Bit assignment diagram for the silicon\_debug register**



The following table shows the silicon\_debug register bit descriptions.

**Table 16-215: silicon\_debug bit descriptions**

Bits	Name	Description	Type	Reset
[31]	event_capture	Enable silicon debug value capture	RW	0
[30]	reserved2	Bits within this register segment are reserved for future product development	RO	0
[29:20]	write_outstanding	Number of outstanding write transactions. From request handshake to response.	RO	0x0
[19:18]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[17:8]	read_outstanding	Number of outstanding read transactions. From request handshake to response.	RO	0x0
[7:5]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[4]	b_stall	When this bit is set to 1, a transfer is stalled on the B channel, where both: BVALID is HIGH. BREADY is LOW.	RO	0
[3]	w_stall	When this bit is set to 1, a transfer is stalled on the W channel, where both: WVALID is HIGH. WREADY is LOW.	RO	0
[2]	aw_stall	When this bit is set to 1, a transfer is stalled on the AW channel, where both: AWVALID is HIGH. AWREADY is LOW	RO	0
[1]	r_stall	When this bit is set to 1, a transfer is stalled on the R channel, where both: RVALID is HIGH. RREADY is LOW.	RO	0
[0]	ar_stall	When this bit is set to 1, a transfer is stalled on the AR channel, where both: ARVALID is HIGH. ARREADY is LOW.	RO	0

## 16.13.12 AMNI qos\_accept\_control register

Contains registers used for configuring QoS acceptance behavior for the AMNI node.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

Address offset

0x84

Type

RW

Reset value

0x00000000

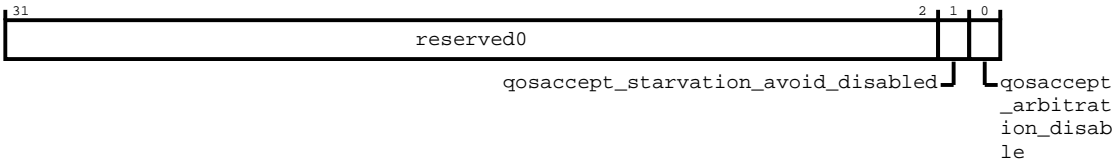
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qos\_accept\_control register bit assignments.

Figure 16-203: Bit assignment diagram for the qos\_accept\_control register



The following table shows the qos\_accept\_control register bit descriptions.

Table 16-216: qos\_accept\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	qosaccept_starvation_avoid_disabled	Disable QoS accept starvation avoidance	RW	0
[0]	qosaccept_arbitration_disable	Disable QoS accept arbitration	RW	0

16.13.13 AMNI cmoovrd register

This register selects between SLVERR or OKAY responses to handle CMOs when downstream completers do not support them.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x88

Type

RW

Reset value

0x00000000

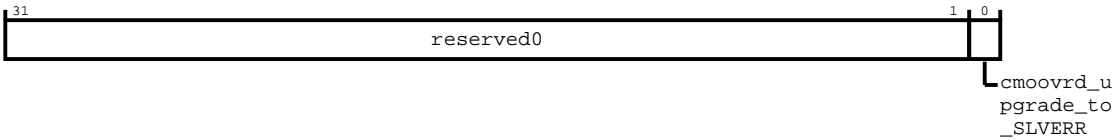
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the cmoovrd register bit assignments.

Figure 16-204: Bit assignment diagram for the cmoovrd register



The following table shows the cmoovrd register bit descriptions.

Table 16-217: cmoovrd bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	cmoovrd_upgrade_to_SLVERR	Upgrade to SLVERR when set, or use the default response OK.	RW	0

16.13.14 AMNI rddata\_agg\_control register

This register contains controls for AMNI read data aggregation. Register value does not take effect if read\_data\_aggregation\_enable in AMNI node\_info is zero.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

**Address offset**

0x8C

**Type**

RW

**Reset value**

0x00000000

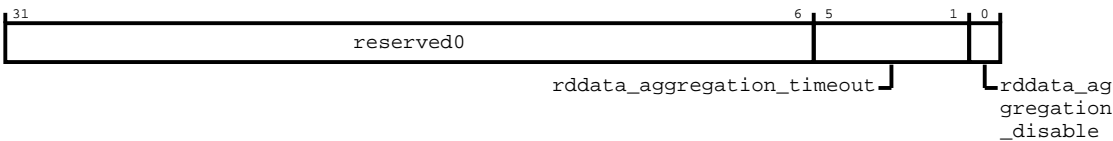
**Constraints**

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the rddata\_agg\_control register bit assignments.

**Figure 16-205: Bit assignment diagram for the rddata\_agg\_control register**



The following table shows the rddata\_agg\_control register bit descriptions.

**Table 16-218: rddata\_agg\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development.	RO	0x0
[5:1]	rddata_aggregation_timeout	The value of this field specifies the maximum cycles the AMNI hold prior read data beat to aggregate with next beat belonging to the same transaction. A value 0 means no timeout, AMNI will wait infinitely when it expects next data beat for the same transaction.	RW	0b00000
[0]	rddata_aggregation_disable	The value of this field specifies whether read data aggregation is disabled on the AMNI.	RW	0

**16.13.15 AMNI interrupt\_status register**

This register indicates the interrupt status of Secure transaction.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0xF0

**Type**  
RW

**Reset value**  
0x00000000

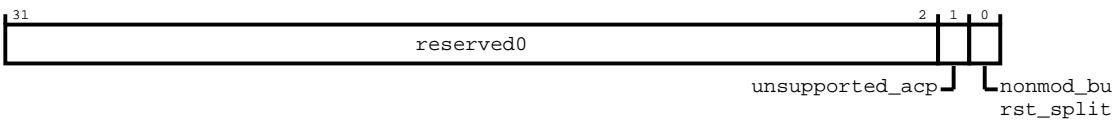
**Constraints**

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the interrupt\_status register bit assignments.

**Figure 16-206: Bit assignment diagram for the interrupt\_status register**



The following table shows the interrupt\_status register bit descriptions.

**Table 16-219: interrupt\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	unsupported_acp	Unsupported ACE5-LiteACP request  Write 1 to clear.	RW	0
[0]	nonmod_burst_split	Non-modifiable Burst Split. Used for non-modifiable transactions which are split  Write 1 to clear.	RW	0

16.13.16 AMNI interrupt\_mask register

This register is the interrupt mask of Secure transactions.

**Configurations**

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF4

Type

RW

Reset value

0x00000000

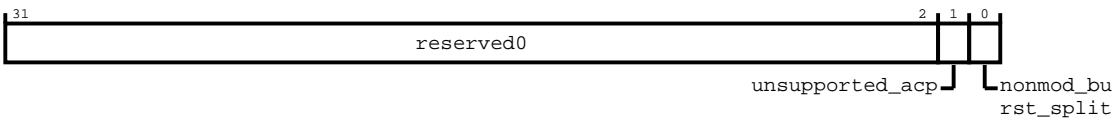
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the interrupt\_mask register bit assignments.

Figure 16-207: Bit assignment diagram for the interrupt\_mask register



The following table shows the interrupt\_mask register bit descriptions.

Table 16-220: interrupt\_mask bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	unsupported_acp	Mask the unsupported ACE5-LiteACP interrupt	RW	0
[0]	nonmod_burst_split	Mask the non-modifiable Burst split interrupt	RW	0

16.13.17 AMNI interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF8

Type

RW

Reset value

0x00000000

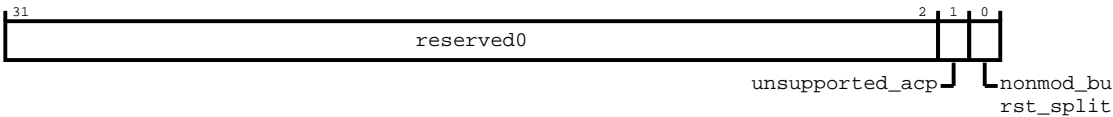
Constraints

None.

Bit descriptions

The following figure shows the interrupt\_status\_ns register bit assignments.

Figure 16-208: Bit assignment diagram for the interrupt\_status\_ns register



The following table shows the interrupt\_status\_ns register bit descriptions.

Table 16-221: interrupt\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	unsupported_acp	Unsupported ACE5-LiteACP request  Write 1 to clear.	RW	0
[0]	nonmod_burst_split	Non-modifiable Burst Split. Used for non-modifiable transactions which are split  Write 1 to clear.	RW	0

16.13.18 AMNI interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

**Width**  
32-bit

**Address offset**  
0xFC

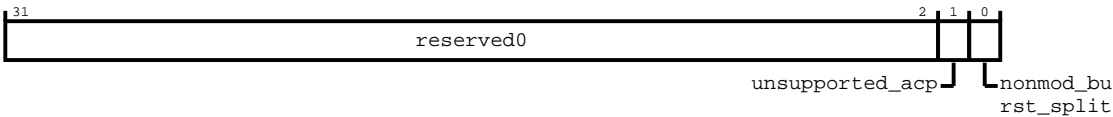
**Type**  
RW

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the interrupt\_mask\_ns register bit assignments.

Figure 16-209: Bit assignment diagram for the interrupt\_mask\_ns register



The following table shows the interrupt\_mask\_ns register bit descriptions.

Table 16-222: interrupt\_mask\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	unsupported_acp	Mask the unsupported ACE5-LiteACP interrupt	RW	0
[0]	nonmod_burst_split	Mask the non-modifiable Burst split interrupt	RW	0

16.13.19 AMNI idm\_device\_id register

This register indicates the statically configured device ID value and is implemented if IDM is enabled.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit



Address offset

0x100

Type

RO

Reset value

See individual bit resets.

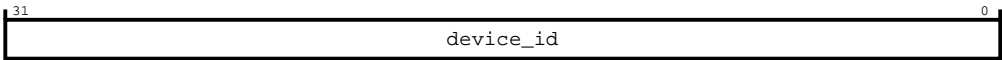
Constraints

None.

Bit descriptions

The following figure shows the `idm_device_id` register bit assignments.

Figure 16-210: Bit assignment diagram for the `idm_device_id` register



The following table shows the `idm_device_id` register bit descriptions.

Table 16-223: `idm_device_id` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	device_id	Returns statically configured ID value	RO	Configuration dependent

16.13.20 AMNI `idm_config` register

This register enables transaction logging, error detection, timeout detection, access control, and reset control.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x104

Type

RW

## Reset value

See individual bit resets.

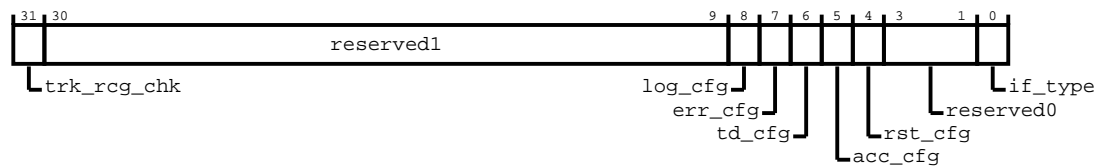
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_config` register bit assignments.

**Figure 16-211: Bit assignment diagram for the `idm_config` register**



The following table shows the `idm_config` register bit descriptions.

**Table 16-224: `idm_config` bit descriptions**

Bits	Name	Description	Type	Reset
[31]	<code>trk_rcg_chk</code>	Tracker Regional Clock Gating (RCG) chicken bit	RW	0
[30:9]	<code>reserved1</code>	Bits within this register segment are reserved for future product development	RO	0x0
[8]	<code>log_cfg</code>	Transaction logging present	RO	1
[7]	<code>err_cfg</code>	Error detection present	RO	1
[6]	<code>td_cfg</code>	Timeout detection present	RO	1
[5]	<code>acc_cfg</code>	Access control present	RO	1
[4]	<code>rst_cfg</code>	Reset control present	RO	1
[3:1]	<code>reserved0</code>	Bits within this register segment are reserved for future product development	RO	0b000
[0]	<code>if_type</code>	Interface type  0 Completer  1 Requester	RO	Configuration dependent

## 16.13.21 AMNI `idm_errctlr` register

This register controls how errors are handled.

## Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x108

Type

RW

Reset value

0x00000000

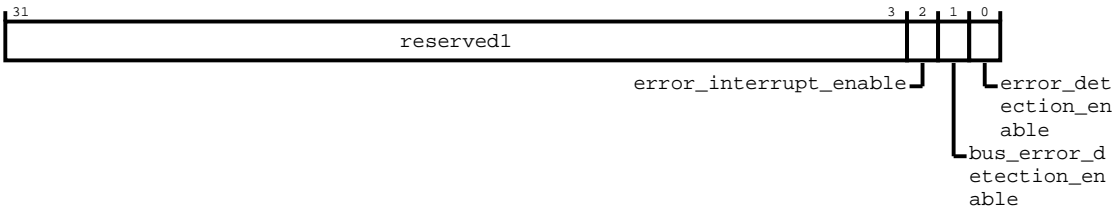
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_errctlr register bit assignments.

Figure 16-212: Bit assignment diagram for the idm\_errctlr register



The following table shows the idm\_errctlr register bit descriptions.

Table 16-225: idm\_errctlr bit descriptions

Bits	Name	Description	Type	Reset
[31:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[2]	error_interrupt_enable	Enable error interrupt for uncorrected error as indicated by IDM_ERRSTATUS.UE fields	RW	0
[1]	bus_error_detection_enable	Enable bus error detection  0 Disabled  1 Enabled when an error is detected and idm_errctlr [ed] is enabled. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set. An error interrupt event is generated (unless masked).	RW	0

Bits	Name	Description	Type	Reset
[0]	error_detection_enable	<p>Error detection global enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when an error is detected. In other words, a timeout error or bus error is detected and its respective detection enable register bit, Timeout_control[TD_EN], or idm_errctrlr[be] is also set. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>An error interrupt event is generated, unless masked.</p>	RW	0

### 16.13.22 AMNI idm\_errstatus register

This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x110

##### Type

RW

##### Reset value

0x00000000

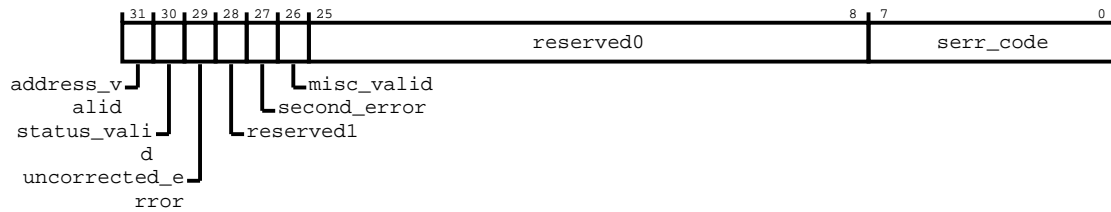
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_errstatus register bit assignments.

**Figure 16-213: Bit assignment diagram for the idm\_errstatus register**



The following table shows the idm\_errstatus register bit descriptions.

**Table 16-226: idm\_errstatus bit descriptions**

Bits	Name	Description	Type	Reset
[31]	address_valid	<p>Address valid. The values are:</p> <p><b>0</b></p> <p>ERRADDR is not valid.</p> <p><b>1</b></p> <p>ERRADDR contains an address that is associated with the highest priority error which this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1 and is not cleared to zero in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[30]	status_valid	<p>Status register is valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRSTATUS not valid</p> <p><b>1</b></p> <p>IDM_ERRSTATUS valid. At least one error has been recorded.</p> <p>This bit ignores writes if any of the following fields is set to 1 and is not being cleared to zero in the same write:</p> <ul style="list-style-type: none"> <li>IDM_ERRSTATUS.UE</li> <li>IDM_ERRSTATUS.AV</li> <li>IDM_ERRSTATUS.OF * IDM_ERRSTATUS.MV</li> </ul> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred</p> <p>This bit ignores writes if IDM_ERRSTATUS.OF is set to 1 and is not being cleared to zero in the same write. This bit is not valid and reads <b>UNKNOWN</b> if IDM_ERRSTATUS.V is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISC0 and IDM_ERRMISC1 not valid</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISC0 and IDM_ERRMISC1 registers contains additional information for an error that this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1, and is not being cleared to 0 in the same write. This bit is a read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	<p>Primary error code. Indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0

16.13.23 AMNI idm\_erraddr\_lsb register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x114

Type

RO

Reset value

0x00000000

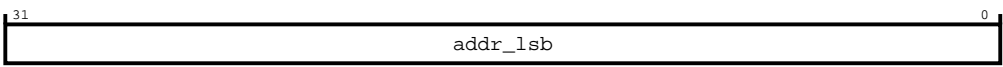
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb register bit assignments.

Figure 16-214: Bit assignment diagram for the idm\_erraddr\_lsb register



The following table shows the idm\_erraddr\_lsb register bit descriptions.

Table 16-227: idm\_erraddr\_lsb bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.13.24 AMNI idm\_erraddr\_msb register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x118

Type

RO

Reset value

0x00000000

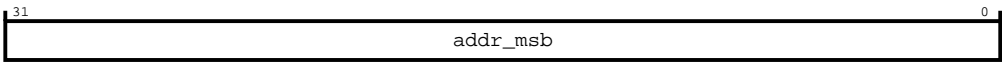
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_msb` register bit assignments.

Figure 16-215: Bit assignment diagram for the `idm_erraddr_msb` register



The following table shows the `idm_erraddr_msb` register bit descriptions.

Table 16-228: `idm_erraddr_msb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.13.25 AMNI `idm_errmisc0` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x128



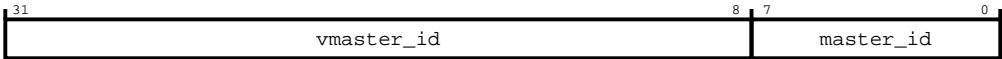
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions.

**Bit descriptions**  
The following figure shows the idm\_errmisc0 register bit assignments.

**Figure 16-216: Bit assignment diagram for the idm\_errmisc0 register**



The following table shows the idm\_errmisc0 register bit descriptions.

**Table 16-229: idm\_errmisc0 bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming AXI AxID into ASNI of the transaction causing an error. The assumption here is there is no manipulation of incoming AXI AxID in ASNI.	RO	0x0
[7:0]	master_id	The AMNI Node ID of the transaction causing an error.	RO	0x0

16.13.26 AMNI idm\_errmisc1 register

This register is the error log of Secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x12C

**Type**  
RO

**Reset value**  
0x00000000

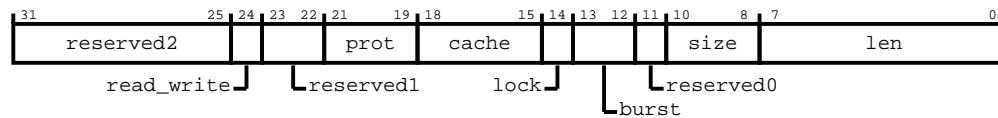
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the idm\_errmisc1 register bit assignments.

**Figure 16-217: Bit assignment diagram for the idm\_errmisc1 register**



The following table shows the idm\_errmisc1 register bit descriptions.

**Table 16-230: idm\_errmisc1 bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	The AXI read or write information of a transaction causing an error  1 Write  0 Read	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	The AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	The AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	The AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	The AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	The AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	The AXI len information of a transaction causing an error.	RO	0x0

## 16.13.27 AMNI idm\_access\_control register

This register controls the state, gated or ungated, of a device.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

Address offset

0x130

Type

RW

Reset value

0x00000000

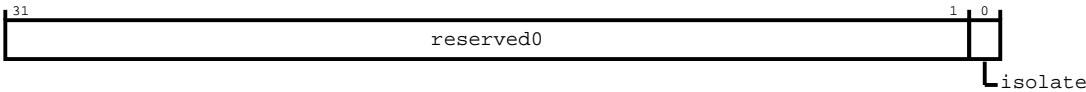
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_access\_control register bit assignments.

Figure 16-218: Bit assignment diagram for the idm\_access\_control register



The following table shows the idm\_access\_control register bit descriptions.

Table 16-231: idm\_access\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	isolate	Perform gating off a device. Reading 1 indicates that the completer device is gated or isolated. Reading 0 indicates that the completer device is ungated or de-isolated. Write 1 to enter gated state. Write 0 to exit gated state. There is some delay to updating this field with the intended write value. Exit from gated state is only successful if there are no outstanding transactions and all error status register bits are cleared. Entry into gated state is only successful if there are no outstanding transactions. While in pending isolation entry state or in active isolation state, a write of 1 to this bit causes reentry to isolation state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 0 is ignored. While in pending isolation exit state, a write of 0 to this bit causes a re-exit to the exit state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS, and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 1 is ignored.	RW	0

16.13.28 AMNI idm\_access\_status register

This register indicates the access status for Secure transactions.

Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

### Width

32-bit

### Address offset

0x134

### Type

RO

### Reset value

0x00000002

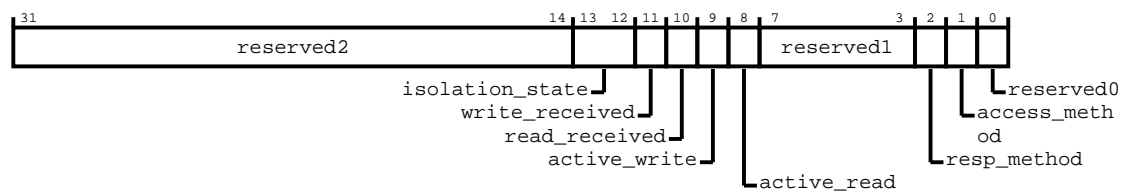
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the idm\_access\_status register bit assignments.

**Figure 16-219: Bit assignment diagram for the idm\_access\_status register**



The following table shows the idm\_access\_status register bit descriptions.

**Table 16-232: idm\_access\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:14]	reserved2	Bits within this register segment are reserved for future product development	RO	0x0
[13:12]	isolation_state	Isolation status:  <b>0b00</b> Isolation exit or entry is successful or not in gated or isolation state  <b>0b01</b> Isolation exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  <b>0b10</b> Isolation entry is unsuccessful or pending because of outstanding transactions  <b>0b11</b> Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00000
[2]	resp_method	Indicates device generates errors in gated access	RO	0
[1]	access_method	Wait for all outstanding to complete, then block input	RO	1
[0]	reserved0	Bits within this register segment are reserved for future product development	RO	0

### 16.13.29 AMNI idm\_access\_readid register

This register is the access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x138

#### Type

RO

#### Reset value

0x00000000

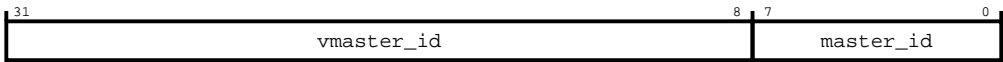
#### Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_access_readid` register bit assignments.

Figure 16-220: Bit assignment diagram for the `idm_access_readid` register



The following table shows the `idm_access_readid` register bit descriptions.

Table 16-233: `idm_access_readid` bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

16.13.30 AMNI `idm_access_writeid` register

This register is the access log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x13C

Type

RO

Reset value

0x00000000

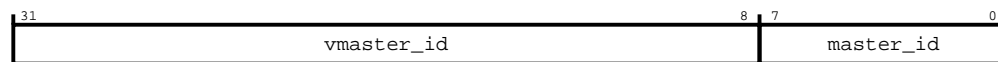
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_access_writeid` register bit assignments.

**Figure 16-221: Bit assignment diagram for the `idm_access_writeid` register**



The following table shows the `idm_access_writeid` register bit descriptions.

**Table 16-234: `idm_access_writeid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AWID signal into the endpoint of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

### 16.13.31 AMNI `idm_reset_control` register

This register controls the reset of a device that is attached to the interconnect.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x140

##### Type

RW

##### Reset value

0x00000002

#### Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the `idm_reset_control` register bit assignments.





Bits	Name	Description	Type	Reset
[0]	reset_control	<p>Performs soft reset of attached device. If the auto bit is set to 1 the network interface gates the external interface, however the soft reset pin is not activated. If the auto bit is 0, the interfaces are not gated until there is a write to bit[0]. In this case, the soft reset pin is activated. Writes have the following effect:</p> <p><b>1</b></p> <p>Request the attached device to enter reset. If the write occurs before soft reset exit has occurred, the write is ignored.</p> <p><b>0</b></p> <p>Request the attached device to exit reset. If the write occurs before soft reset entry has occurred, the write is ignored.</p> <p>Software polls this register to determine if soft reset entry or exit has occurred, using the following values:</p> <p><b>1</b></p> <p>Indicates that the device is in reset.</p> <p><b>0</b></p> <p>Indicates that the device is not in reset.</p> <p>This register value updates to reflect a request for reset entry or reset exit, but the update can only occur after required internal conditions are met. Until these conditions are met, a read to this register returns the old value. For example, outstanding transactions currently being handled must complete before this register value updates. To ensure reset propagation within the device, it is the responsibility of the software to permit enough cycles after soft reset assertion is reflected in the IDM_RESET_CONTROL register before exiting soft reset by triggering a write of 0. If this responsibility is not met, the behavior is <b>UNDEFINED</b> or <b>UNPREDICTABLE</b>. When this register value is 1, the external soft reset pin that connects to the attached AXI requester or completer device is asserted, using the correct polarity of the reset pin. When this register value is 0, the external soft reset pin that connects to the attached AXI requester or completer device is deasserted, using the correct polarity of the reset pin. When in pending soft reset entry state or in active soft reset state, a write of 1 to this bit causes reentry to soft reset state. This write causes the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers to be cleared. A write of 0 is ignored. While in pending soft reset exit state, a write of 0 to this bit causes re-exit to exit state. A write of 0 also clears the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers. A write of 1 is ignored.</p>	RW	0

### 16.13.32 AMNI idm\_reset\_status register

This register indicates mostly the reset status of Secure transactions. However, the rst\_exit\_state field indicates reset exit state of secure or non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

Address offset

0x144

Type

RO

Reset value

0x00000000

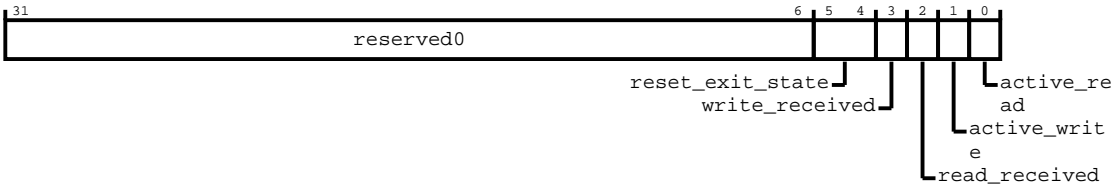
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_reset\_status register bit assignments.

Figure 16-223: Bit assignment diagram for the idm\_reset\_status register



The following table shows the idm\_reset\_status register bit descriptions.

Table 16-236: idm\_reset\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:4]	reset_exit_state	Reset exit state  00 Reset exit or entry is successful or not in reset state  01 Reset exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  10 Reset exit is unsuccessful or pending because of outstanding transactions  11 Reset exit is unsuccessful or pending because of both uncleared error status bits and outstanding transactions	RO	0b00
[3]	write_received	A 1 indicates that an active Secure write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"><li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li><li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li></ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0

### 16.13.33 AMNI idm\_reset\_readid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x148

##### Type

RO

##### Reset value

0x00000000

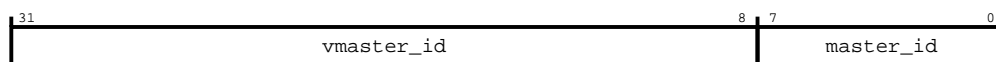
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_readid register bit assignments.

**Figure 16-224: Bit assignment diagram for the idm\_reset\_readid register**



The following table shows the idm\_reset\_readid register bit descriptions.

**Table 16-237: idm\_reset\_readid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.13.34 AMNI idm\_reset\_writeid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x14C

##### Type

RO

##### Reset value

0x00000000

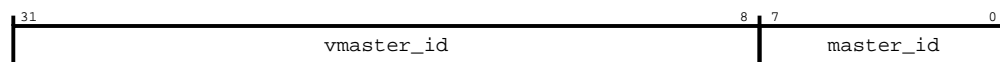
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid register bit assignments.

**Figure 16-225: Bit assignment diagram for the idm\_reset\_writeid register**



The following table shows the idm\_reset\_writeid register bit descriptions.

### Table 16-238: idm\_reset\_writeid bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.13.35 AMNI idm\_timeout\_control register

This register is present when timeout detection is configured.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x150

## Type

RW

## Reset value

0x00000000

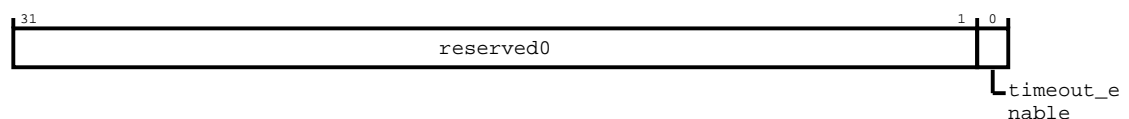
## Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the `idm_timeout_control` register bit assignments.

**Figure 16-226: Bit assignment diagram for the `idm_timeout_control` register**



The following table shows the `idm_timeout_control` register bit descriptions.

**Table 16-239: idm\_timeout\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	timeout_enable	<p>Timeout detection enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when a timeout is detected. The timeout is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>A timeout interrupt event is generated, unless it is masked.</p>	RW	0

### 16.13.36 AMNI idm\_timeout\_value register

This register controls the duration that is used to determine if a transaction has timed out.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x154

##### Type

RW

##### Reset value

0x00000004

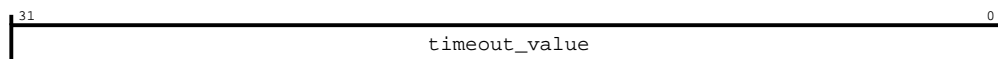
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the idm\_timeout\_value register bit assignments.

**Figure 16-227: Bit assignment diagram for the idm\_timeout\_value register**



The following table shows the `idm_timeout_value` register bit descriptions.

### Table 16-240: idm\_timeout\_value bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	timeout_value	Controls the duration that is used to determine if a transaction has timed out. The actual duration is $2^{\text{timeout\_exponent}}$ cycles. The minimum value is 4. Values of 0, 1, 2, or 3 are treated as 4. The maximum value is 30. Values greater than 30 are treated as 30.	RW	0x4

### 16.13.37 AMNI idm\_interrupt\_status register

This register indicates the interrupt status of Secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x158

## Type

RW

## Reset value

0x00000000

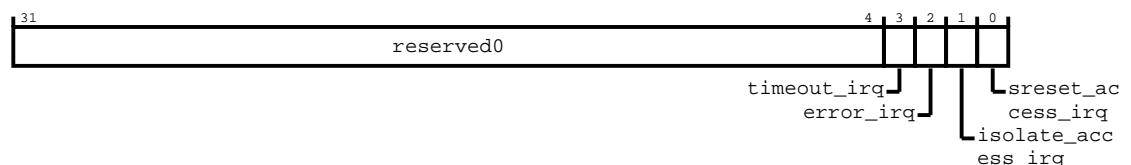
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_interrupt_status` register bit assignments.

**Figure 16-228: Bit assignment diagram for the `idm_interrupt_status` register**



The following table shows the `idm_interrupt_status` register bit descriptions.

**Table 16-241: idm\_interrupt\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.13.38 AMNI idm\_interrupt\_mask register

This register is the interrupt mask of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x15C

##### Type

RW

##### Reset value

0x00000000

#### Constraints

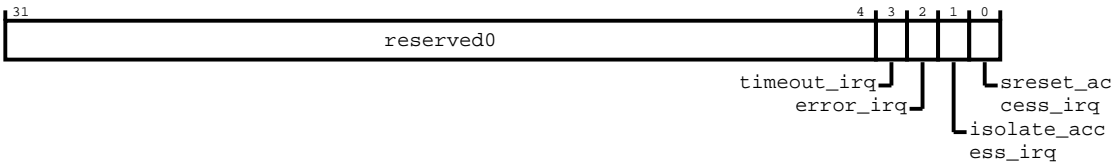
Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask register bit assignments.



Figure 16-229: Bit assignment diagram for the `idm_interrupt_mask` register



The following table shows the `idm_interrupt_mask` register bit descriptions.

Table 16-242: `idm_interrupt_mask` bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

16.13.39 AMNI `idm_errstatus_ns` register

This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x160

Type

RW

Reset value

0x00000000

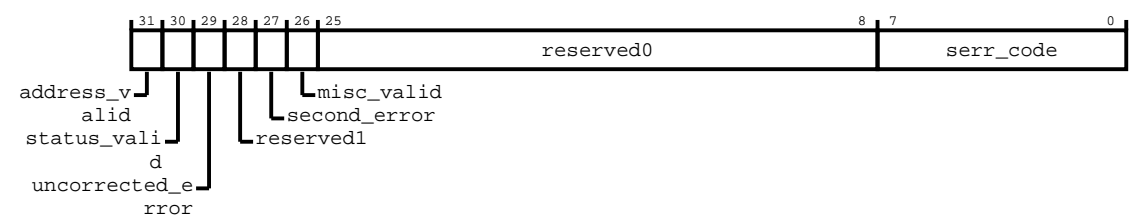
Constraints

None.

Bit descriptions

The following figure shows the idm\_errstatus\_ns register bit assignments.

Figure 16-230: Bit assignment diagram for the idm\_errstatus\_ns register



The following table shows the idm\_errstatus\_ns register bit descriptions.

Table 16-243: idm\_errstatus\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31]	address_valid	Address valid. The values are:  <b>0</b>  ERRADDR is not valid.  <b>1</b>  ERRADDR contains an address that is associated with the highest priority error that this record captures.  This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1 and is not cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0
[30]	status_valid	Status register valid. The values are:  <b>0</b>  IDM_ERRSTATUS_NS is not valid.  <b>1</b>  IDM_ERRSTATUS_NS is valid. At least one error has been recorded.  This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred.</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred.</p> <p>This bit ignores writes if the oe field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is not valid and reads <b>UNKNOWN</b> if the v field of the IDM_ERRSTATUS_NS register is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISCO_NS and IDM_ERRMISC1_NS are not valid.</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISCO_NS and IDM_ERRMISC1_NS registers contains additional information for an error that this record captures.</p> <p>This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1, and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	<p>Primary error code, indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0

16.13.40 AMNI idm\_erraddr\_lsb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x164

Type

RO

Reset value

0x00000000

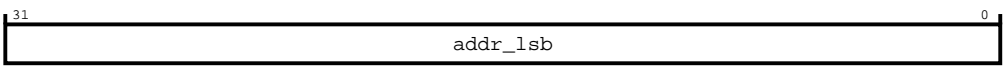
Constraints

None.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb\_ns register bit assignments.

Figure 16-231: Bit assignment diagram for the idm\_erraddr\_lsb\_ns register



The following table shows the idm\_erraddr\_lsb\_ns register bit descriptions.

Table 16-244: idm\_erraddr\_lsb\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.13.41 AMNI idm\_erraddr\_msb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x168

Type

RO

Reset value

0x00000000

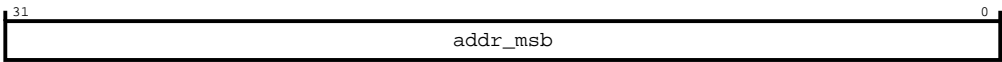
Constraints

None.

Bit descriptions

The following figure shows the `idm_erraddr_msb_ns` register bit assignments.

Figure 16-232: Bit assignment diagram for the `idm_erraddr_msb_ns` register



The following table shows the `idm_erraddr_msb_ns` register bit descriptions.

Table 16-245: `idm_erraddr_msb_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.13.42 AMNI `idm_errmisc0_ns` register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x178

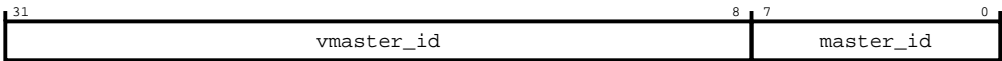
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the `idm_errmisc0_ns` register bit assignments.

**Figure 16-233: Bit assignment diagram for the `idm_errmisc0_ns` register**



The following table shows the `idm_errmisc0_ns` register bit descriptions.

**Table 16-246: `idm_errmisc0_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AxiD into ASNI of the transaction causing an error. The assumption is no manipulation of incoming AXI AxiD in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.13.43 AMNI `idm_errmisc1_ns` register

This register is the error log of Non-secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x17C

**Type**  
RO

**Reset value**  
0x00000000



## Address offset

0x184

## Type

RO

## Reset value

0x00000000

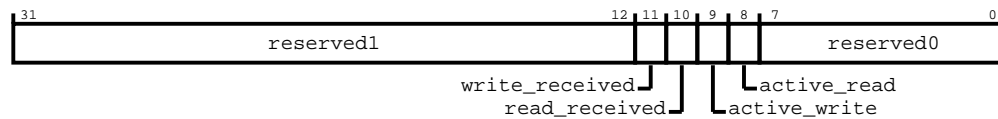
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_access_status_ns` register bit assignments.

**Figure 16-235: Bit assignment diagram for the `idm_access_status_ns` register**



The following table shows the `idm_access_status_ns` register bit descriptions.

**Table 16-248: `idm_access_status_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved1	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:0]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0



### 16.13.45 AMNI idm\_access\_readid\_ns register

This register is the access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x188

##### Type

RO

##### Reset value

0x00000000

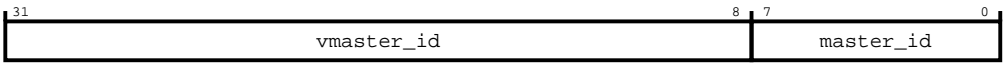
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_access\_readid\_ns register bit assignments.

**Figure 16-236: Bit assignment diagram for the idm\_access\_readid\_ns register**



The following table shows the idm\_access\_readid\_ns register bit descriptions.

**Table 16-249: idm\_access\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

16.13.46 AMNI idm\_access\_writeid\_ns register

This register is the access log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x18C

Type

RO

Reset value

0x00000000

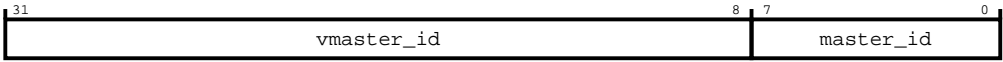
Constraints

None.

Bit descriptions

The following figure shows the idm\_access\_writeid\_ns register bit assignments.

Figure 16-237: Bit assignment diagram for the idm\_access\_writeid\_ns register



The following table shows the idm\_access\_writeid\_ns register bit descriptions.

Table 16-250: idm\_access\_writeid\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the IDM_ACCESS_STATUS_NS register field active_write is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

16.13.47 AMNI idm\_reset\_status\_ns register

This register indicates the reset status of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x194

Type

RO

Reset value

0x00000000

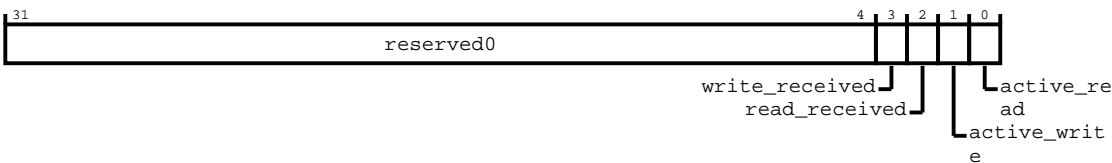
Constraints

None.

Bit descriptions

The following figure shows the idm\_reset\_status\_ns register bit assignments.

Figure 16-238: Bit assignment diagram for the idm\_reset\_status\_ns register



The following table shows the idm\_reset\_status\_ns register bit descriptions.

Table 16-251: idm\_reset\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[3]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"><li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li><li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li></ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to 0 on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates that there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates that there is at least one read transaction currently in progress.	RO	0

### 16.13.48 AMNI idm\_reset\_readid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x198

##### Type

RO

##### Reset value

0x00000000

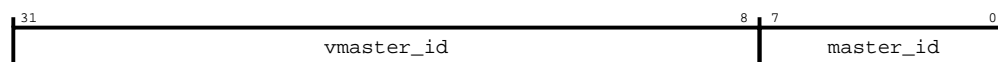
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_readid\_ns register bit assignments.

**Figure 16-239: Bit assignment diagram for the idm\_reset\_readid\_ns register**



The following table shows the idm\_reset\_readid\_ns register bit descriptions.

**Table 16-252: idm\_reset\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

### 16.13.49 AMNI idm\_reset\_writeid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x19C

##### Type

RO

##### Reset value

0x00000000

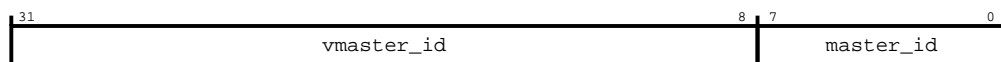
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid\_ns register bit assignments.

**Figure 16-240: Bit assignment diagram for the idm\_reset\_writeid\_ns register**



The following table shows the idm\_reset\_writeid\_ns register bit descriptions.

**Table 16-253: idm\_reset\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when active_write field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

### 16.13.50 AMNI idm\_interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x1A8

##### Type

RW

##### Reset value

0x00000000

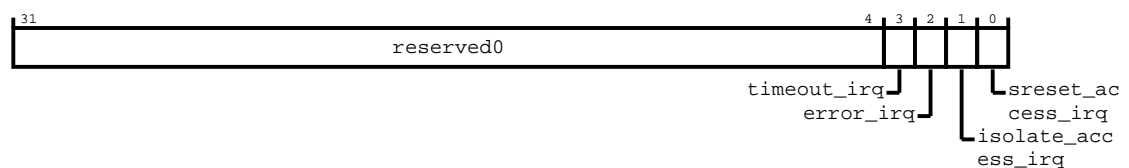
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_interrupt\_status\_ns register bit assignments.

**Figure 16-241: Bit assignment diagram for the idm\_interrupt\_status\_ns register**



The following table shows the idm\_interrupt\_status\_ns register bit descriptions.

**Table 16-254: idm\_interrupt\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.13.51 AMNI idm\_interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x1AC

##### Type

RW

##### Reset value

0x00000000

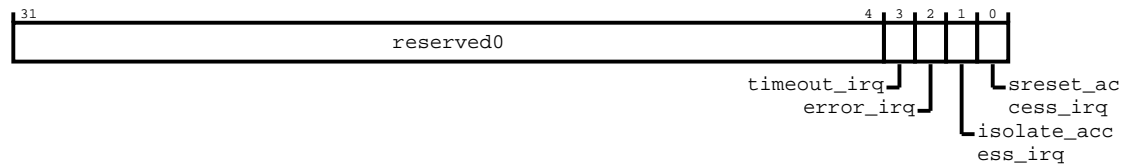
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask\_ns register bit assignments.

**Figure 16-242: Bit assignment diagram for the idm\_interrupt\_mask\_ns register**



The following table shows the idm\_interrupt\_mask\_ns register bit descriptions.

**Table 16-255: idm\_interrupt\_mask\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

## 16.14 HSNi register summary

This section describes the HSNi registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-256: HSNi register summary**

Offset	Name	Type	Reset	Width	Description
0x00	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a node for HSNi registers.
0x04	<a href="#">node_info</a>	RO	See individual bit resets.	32-bit	This register provides node information for HSNi, such as data width.
0x08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register controls Secure access.
0x0C	<a href="#">pmusela</a>	RW	0x00000000	32-bit	This register is used to select the event values in the HSNi event crossbar.
0x10	<a href="#">pmuselb</a>	RW	0x00000000	32-bit	This register is used to select the event values in the HSNi event crossbar.
0x14	<a href="#">interface_id_0_3</a>	RO	See individual bit resets.	32-bit	Contains information about the HSNi interface IDs for interfaces 0-3.
0x24	<a href="#">num_sub_features</a>	RO	See individual bit resets.	32-bit	Number of subfeatures.
0x28	<a href="#">sub_feature_0_type</a>	RO	See individual bit resets.	32-bit	Subfeature 0 type.
0x2C	<a href="#">sub_feature_0_pointer</a>	RO	See individual bit resets.	32-bit	Subfeature 0 pointer.



Offset	Name	Type	Reset	Width	Description
0x44	<a href="#">node_control</a>	RW	See individual bit resets.	32-bit	This register controls how Bursts are split. If the secure_transfers property is also 0, then it controls mapping of the Non-secure bit. It also provides the applied Burst split value.
0x48	<a href="#">address_remap</a>	RW	0x00000000	32-bit	This register is used to program up to eight remap states supported by the address decode logic.
0x4c	<a href="#">hang_detector_ctrl</a>	RW	0x00000000	32-bit	Registers used to configure the hang detector. Fields in this register are only present if the hang detector feature is enabled.
0x80	<a href="#">silicon_debug</a>	RW	0x00000000	32-bit	This register monitors the status of completer interface channels.
0x84	<a href="#">qosctl</a>	RW	0x00000000	32-bit	This register controls the QoS settings for BQV and TSPEC and enables a QoS value on inbound transactions to be overridden.
0x88	<a href="#">wdatthrs</a>	RW	0x00000000	32-bit	This register specifies the number of write data beats to be queued before the write packet is sent.
0x90	<a href="#">qos_values</a>	RW	0x00000000	32-bit	This register stores the value that is applied to GT transactions if the BQV regulator is not present or enabled.
0xA0	<a href="#">qosot</a>	RW	0x00000000	32-bit	Contains registers used for configuring the number of maximum outstanding transaction for the HSNi node.
0xBC	<a href="#">qoscompk</a>	RW	0x00000000	32-bit	This register controls the QoS peak rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.
0xC0	<a href="#">qoscombur</a>	RW	0x00000000	32-bit	This register controls the QoS burstiness allowance for combined read and write hard bandwidth regulation, TSPEC, of a completer interface.
0xC4	<a href="#">qoscomavg</a>	RW	0x00000000	32-bit	This register controls the QoS average rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.
0xD0	<a href="#">qoscombqv</a>	RW	0x00000000	32-bit	This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for both read and write soft bandwidth regulation, BQV, of a completer interface.
0xE0	<a href="#">mpam_control</a>	RW	0x00000000	32-bit	If GT_MPAM_SUPPORT is enabled, the register drives the MPAM values for a specific HSNi.
0xF0	<a href="#">interrupt_status</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0xF4	<a href="#">interrupt_mask</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0xF8	<a href="#">interrupt_status_ns</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0xFC	<a href="#">interrupt_mask_ns</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.
0x100	<a href="#">idm_device_id</a>	RO	See individual bit resets.	32-bit	This register indicates the statically configured device ID value and is implemented if IDM is enabled.
0x104	<a href="#">idm_config</a>	RW	See individual bit resets.	32-bit	This register enables transaction logging, error detection, timeout detection, access control, and reset control.
0x108	<a href="#">idm_errctlr</a>	RW	0x00000000	32-bit	This register controls how errors are handled.
0x110	<a href="#">idm_errstatus</a>	RW	0x00000000	32-bit	This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.
0x114	<a href="#">idm_erraddr_lsb</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x118	<a href="#">idm_erraddr_msb</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x128	<a href="#">idm_errmisc0</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x12C	<a href="#">idm_errmisc1</a>	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x130	<a href="#">idm_access_control</a>	RW	0x00000000	32-bit	This register controls the state, gated or ungated, of a device.
0x134	<a href="#">idm_access_status</a>	RO	0x00000002	32-bit	This register indicates the access status for Secure transactions.
0x138	<a href="#">idm_access_readid</a>	RO	0x00000000	32-bit	This register is the access log of Secure transactions.

Offset	Name	Type	Reset	Width	Description
0x13C	<a href="#">idm_access_writeid</a>	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x140	<a href="#">idm_reset_control</a>	RW	0x00000002	32-bit	This register controls the reset of a device that is attached to the interconnect.
0x144	<a href="#">idm_reset_status</a>	RO	0x00000000	32-bit	This register indicates mostly the reset status of Secure transactions. However, the <code>rst_exit_state</code> field indicates reset exit state of secure or non-secure transactions.
0x148	<a href="#">idm_reset_readid</a>	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x14C	<a href="#">idm_reset_writeid</a>	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x150	<a href="#">idm_timeout_control</a>	RW	0x00000000	32-bit	This register is present when timeout detection is configured.
0x154	<a href="#">idm_timeout_value</a>	RW	0x00000004	32-bit	This register controls the duration that is used to determine if a transaction has timed out.
0x158	<a href="#">idm_interrupt_status</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0x15C	<a href="#">idm_interrupt_mask</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0x160	<a href="#">idm_errstatus_ns</a>	RW	0x00000000	32-bit	This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.
0x164	<a href="#">idm_erraddr_lsb_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x168	<a href="#">idm_erraddr_msb_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x178	<a href="#">idm_errmisc0_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x17C	<a href="#">idm_errmisc1_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x184	<a href="#">idm_access_status_ns</a>	RO	0x00000000	32-bit	This register indicates the access status for Non-secure transactions.
0x188	<a href="#">idm_access_readid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x18C	<a href="#">idm_access_writeid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x194	<a href="#">idm_reset_status_ns</a>	RO	0x00000000	32-bit	This register indicates the reset status of Non-secure transactions.
0x198	<a href="#">idm_reset_readid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x19C	<a href="#">idm_reset_writeid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x1A8	<a href="#">idm_interrupt_status_ns</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0x1AC	<a href="#">idm_interrupt_mask_ns</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.

### 16.14.1 HSNl node\_type register

This register identifies the node type as a node for HSNl registers.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x00

Type

RO

Reset value

See individual bit resets.

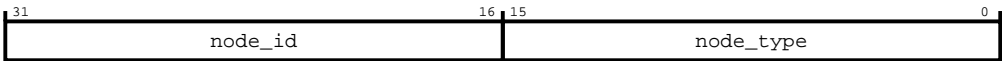
Constraints

None.

Bit descriptions

The following figure shows the node\_type register bit assignments.

Figure 16-243: Bit assignment diagram for the node\_type register



The following table shows the node\_type register bit descriptions.

Table 16-257: node\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	node_id	The HSNI ID that is assigned during network construction.	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x07, and it identifies the associated node type as a node for HSNI registers.	RO	0x7

16.14.2 HSNI node\_info register

This register provides node information for HSNI, such as data width.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RO

Reset value

See individual bit resets.

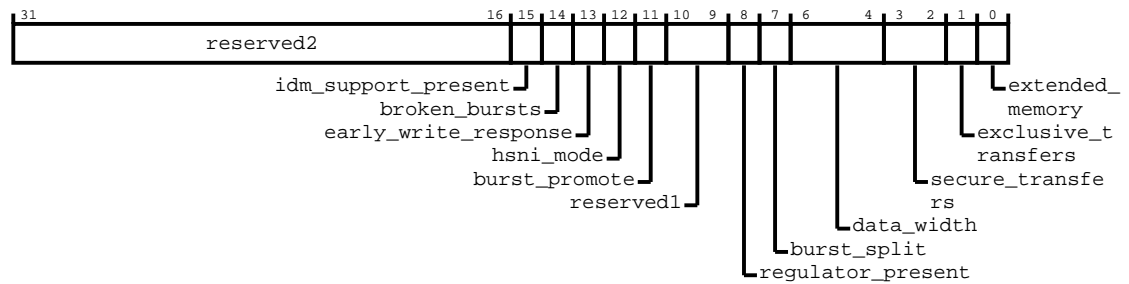
## Constraints

None.

## Bit descriptions

The following figure shows the node\_info register bit assignments.

**Figure 16-244: Bit assignment diagram for the node\_info register**



The following table shows the node\_info register bit descriptions.

**Table 16-258: node\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:16]	reserved2	Bits within this register segment are reserved for future product development	RO	0x0
[15]	idm_support_present	IDM support present: 0 IDM support logic is not present 1 IDM support logic is present	RO	Configuration dependent
[14]	broken_bursts	Broken Bursts: 0 There is no logic to handle broken Bursts. 1 There is logic present to handle broken Bursts.	RO	Configuration dependent
[13]	early_write_response	Early write response: 0 HSNI does not generate early write response. 1 HSNI generates early write response.	RO	Configuration dependent
[12]	hsni_mode	HSNI mode: 0 HSNI is not in mirror mode. 1 HSNI is in mirror mode.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[11]	burst_promote	Burst promote present: <b>0</b> Burst promote logic is not present. <b>1</b> Burst promote logic is present.	RO	Configuration dependent
[10:9]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[8]	regulator_present	Regulator present: <b>0</b> Regulator logic is not present <b>1</b> Regulator logic is present	RO	Configuration dependent
[7]	burst_split	Burst split present: <b>0</b> Burst split logic is not present <b>1</b> Burst split logic is present	RO	Configuration dependent
[6:4]	data_width	Data width, HSIZE encoded: <b>0b000</b> Reserved <b>0b001</b> Reserved <b>0b010</b> 4 bytes <b>0b011</b> 8 bytes <b>0b100</b> 16 bytes <b>0b101</b> 32 bytes <b>0b110</b> 64 bytes <b>0b111</b> 128 bytes	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[3:2]	secure_transfers	<p>Controls the security level of transfers from the HSNI. If secure_transfers = 00, software programs this register to set the security attribute of transfers sent into the network from the HSNI. the encodings of this field are:</p> <p><b>0b00</b> Software programs this register to set the security attribute for requests from this completer</p> <p><b>0b01</b> The HNONSEC pin exists and is used to pass the security attribute</p> <p><b>0b10</b> All requests which originate from this completer interface are marked Secure. Configure at build time.</p> <p><b>0b11</b> All requests which originate from this completer interface are marked Non-secure. Configure at build time.</p>	RO	Configuration dependent
[1]	exclusive_transfers	<p>Indicates if the HSNI node supports exclusive transfers:</p> <p><b>0</b> Does not support exclusive transfers</p> <p><b>1</b> Does support exclusive transfers</p>	RO	Configuration dependent
[0]	extended_memory	<p>Indicates if the HSNI node supports extended memory types:</p> <p><b>0</b> Does not support extended memory types</p> <p><b>1</b> Does support extended memory types</p>	RO	Configuration dependent

### 16.14.3 HSNI secure\_access register

This register controls Secure access.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x08

##### Type

RW

##### Reset value

0x00000000

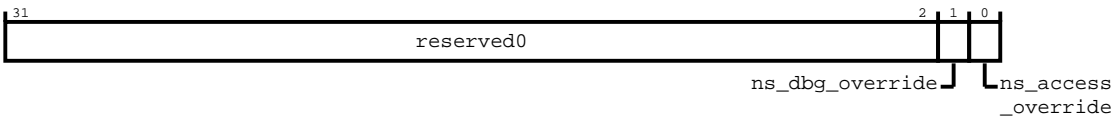
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the secure\_access register bit assignments.

Figure 16-245: Bit assignment diagram for the secure\_access register



The following table shows the secure\_access register bit descriptions.

Table 16-259: secure\_access bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	ns_dbg_override	Enables/Disables non-secure access to AHB completer node PMU and interface registers	RW	0
[0]	ns_access_override	Enables/Disables non-secure access to AHB completer node registers	RW	0

16.14.4 HSNl pmusela register

This register is used to select the event values in the HSNl event crossbar.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x0C

Type

RW

Reset value

0x00000000

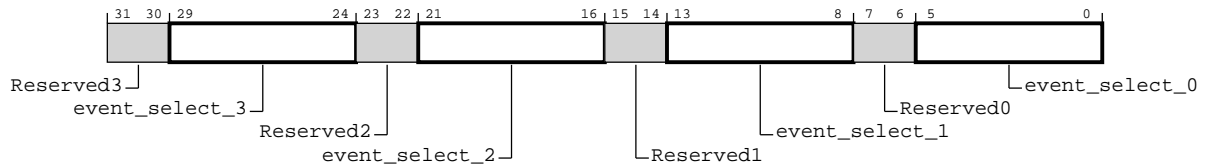
Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmusela register bit assignments.

**Figure 16-246: Bit assignment diagram for the pmusela register**



The following table shows the pmusela register bit descriptions.

**Table 16-260: pmusela bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

### 16.14.5 HSNi pmuselb register

This register is used to select the event values in the HSNi event crossbar.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x10

##### Type

RW

##### Reset value

0x00000000



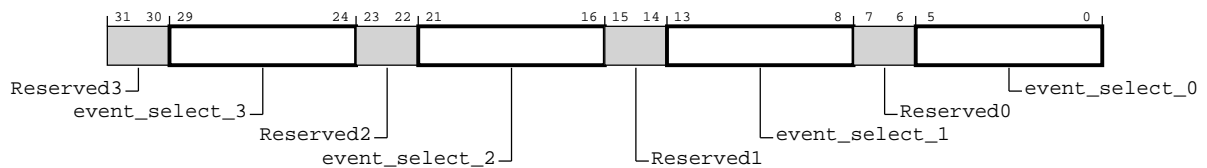
## Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmuselb register bit assignments.

**Figure 16-247: Bit assignment diagram for the pmuselb register**



The following table shows the pmuselb register bit descriptions.

**Table 16-261: pmuselb bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

### 16.14.6 HSNi interface\_id\_0\_3 register

Contains information about the HSNi interface IDs for interfaces 0-3.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

### Width

32-bit

### Address offset

0x14

Type

RO

Reset value

See individual bit resets.

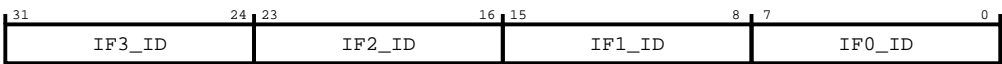
Constraints

None.

Bit descriptions

The following figure shows the interface\_id\_0\_3 register bit assignments.

Figure 16-248: Bit assignment diagram for the interface\_id\_0\_3 register



The following table shows the interface\_id\_0\_3 register bit descriptions.

Table 16-262: interface\_id\_0\_3 bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	IF3_ID	Reserved	RO	Configuration dependent
[23:16]	IF2_ID	Reserved	RO	Configuration dependent
[15:8]	IF1_ID	Reserved	RO	Configuration dependent
[7:0]	IF0_ID	HSNI interface ID 0	RO	Configuration dependent

16.14.7 HSNI num\_sub\_features register

Number of subfeatures.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x24

Type

RO

**Reset value**

See individual bit resets.

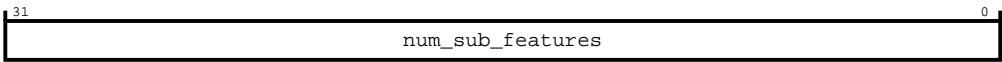
**Constraints**

None.

**Bit descriptions**

The following figure shows the num\_sub\_features register bit assignments.

**Figure 16-249: Bit assignment diagram for the num\_sub\_features register**



The following table shows the num\_sub\_features register bit descriptions.

**Table 16-263: num\_sub\_features bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	num_sub_features	Number of subfeatures	RO	Configuration dependent

**16.14.8 HSNl sub\_feature\_0\_type register**

Subfeature 0 type.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**

32-bit

**Address offset**

0x28

**Type**

RO

**Reset value**

See individual bit resets.

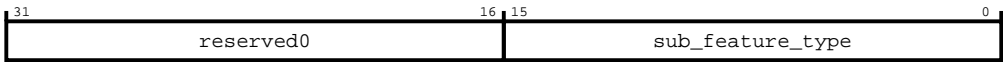
**Constraints**

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_type register bit assignments.

Figure 16-250: Bit assignment diagram for the sub\_feature\_0\_type register



The following table shows the sub\_feature\_0\_type register bit descriptions.

Table 16-264: sub\_feature\_0\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[15:0]	sub_feature_type	Subfeature 0 type	RO	Configuration dependent

16.14.9 HSN1 sub\_feature\_0\_pointer register

Subfeature 0 pointer.

Configurations

The number of registers of this type that are present depends on the number of subfeatures in the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x2C

Type

RO

Reset value

See individual bit resets.

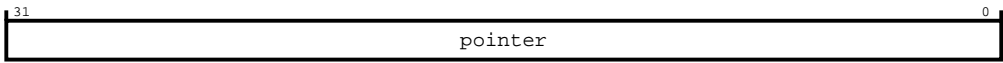
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_pointer register bit assignments.

Figure 16-251: Bit assignment diagram for the sub\_feature\_0\_pointer register



The following table shows the sub\_feature\_0\_pointer register bit descriptions.

Table 16-265: sub\_feature\_0\_pointer bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	pointer	Subfeature 0 pointer	RO	Configuration dependent

16.14.10 HSNi node\_control register

This register controls how Bursts are split. If the secure\_transfers property is also 0, then it controls mapping of the Non-secure bit. It also provides the applied Burst split value.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x44

Type

RW

Reset value

See individual bit resets.

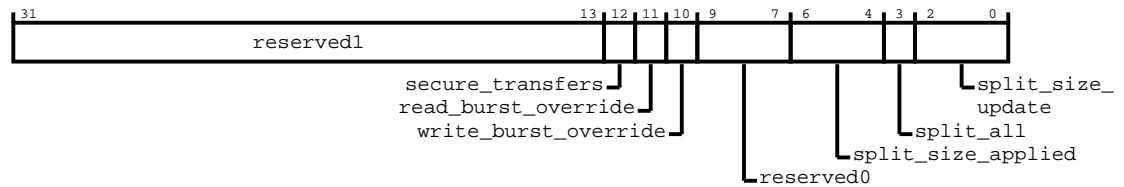
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the node\_control register bit assignments.

**Figure 16-252: Bit assignment diagram for the node\_control register**



The following table shows the node\_control register bit descriptions.

**Table 16-266: node\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:13]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[12]	secure_transfers	<p>If the secure_transfers field from the HSNI_NODE_INFO register = 00 it encodes a software programmable registry. Therefore this field is relevant if the secure_transfers field of HSNI_NODE_INFO = 00. The encoding of this field is then:</p> <p><b>0</b></p> <p>Secure</p> <p><b>1</b></p> <p>Non-secure</p> <p>If secure_transfers = 01, it implies that HNONSEC pin is supported upstream of HSNI. Therefore this register bit is not relevant. If secure_transfers = 00, the HNONSEC pin is unavailable. Therefore this register bit determines the security attribute of all requests from the upstream completer. If secure_transfers = 02 or secure_transfers = 03, the HNONSEC pin is unavailable. However the security attribute of the HSNI is always Secure or Non-secure and is set at build time. This register bit becomes read-only and:</p> <ul style="list-style-type: none"> <li>Reset value is 1 if secure_transfers = 03</li> <li>Reset value is 0 if secure_transfers = 02</li> </ul>	RW	Configuration dependent
[11]	read_burst_override	If set, all AHB read Bursts are converted into singles if the Burst splitter is enabled - the parameter BURST_CONVERT [0] = 1	RW	0
[10]	write_burst_override	If set, all AHB write Bursts are converted into singles if the Burst splitter is enabled - the parameter BURST_CONVERT [0] = 1	RW	0
[9:7]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[6:4]	split_size_applied	<p>The value of Burst split size that is applied. The value is based on the size of the address stripe. This field indicates the applied Burst size. The values are the lower of:</p> <ul style="list-style-type: none"> <li>The configured minimum address stripe size, entered through the address map * Bits 2:0 of this register</li> </ul> <p>This field is read only.</p>	RO	Configuration dependent
[3]	split_all	Burst split all. If set, modifiable Bursts to non-striped regions are also split. This field is read/write.	RW	0

Bits	Name	Description	Type	Reset
[2:0]	split_size_update	<p>The value of Burst split size to apply. The supported encodings are:</p> <p><b>0b000</b> Reserved</p> <p><b>0b001</b> 64 bytes</p> <p><b>0b010</b> 128 bytes</p> <p><b>0b011</b> 256 bytes</p> <p><b>0b100</b> 512 bytes</p> <p><b>0b101</b> 1024 bytes, no Burst split</p> <p><b>0b110</b> Reserved, no Burst split</p> <p><b>0b111</b> Reserved, no Burst split</p> <p>This field is read/write.</p>	RW	0b111

### 16.14.11 HSN1 address\_remap register

This register is used to program up to eight remap states supported by the address decode logic.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x48

##### Type

RW

##### Reset value

0x00000000

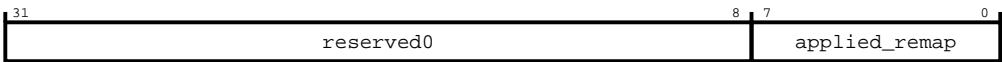
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the address\_remap register bit assignments.

Figure 16-253: Bit assignment diagram for the address\_remap register



The following table shows the address\_remap register bit descriptions.

Table 16-267: address\_remap bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	applied_remap	If multiple bits are set, the bit for each remap with the lowest bit set is taken.	RW	0x0

16.14.12 HSNl hang\_detector\_ctrl register

Registers used to configure the hang detector. Fields in this register are only present if the hang detector feature is enabled.

Configurations

This register is only present if timeout detection is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x4c

Type

RW

Reset value

0x00000000



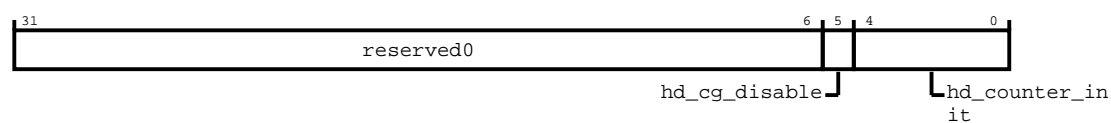
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the hang\_detector\_ctrl register bit assignments.

Figure 16-254: Bit assignment diagram for the hang\_detector\_ctrl register



The following table shows the hang\_detector\_ctrl register bit descriptions.

Table 16-268: hang\_detector\_ctrl bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5]	hd_cg_disable	Hang detector clock gate disable:  0 clock gate in the hang detector is enabled  1 clock gate in the hang detector is disabled, which disables the hang detector	RW	0

Bits	Name	Description	Type	Reset
[4:0]	hd_counter_init	<p>Timeout setting for the hang detector. Each 5'dx encoding encodes a timeout range, shown in both clock cycles and duration for a 1GHz clock.</p> <p><b>5'd0</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{30}</math> to <math>4 \times 2^{30}</math>. Timeout range in duration at 1GHz = 3.0s to 4s.</p> <p><b>5'd1</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{29}</math> to <math>4 \times 2^{29}</math>. Timeout range in duration at 1GHz = 1.6s to 2s.</p> <p><b>5'd2</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{28}</math> to <math>4 \times 2^{28}</math>. Timeout range in duration at 1GHz = 805ms to 1s.</p> <p><b>5'd3</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{27}</math> to <math>4 \times 2^{27}</math>. Timeout range in duration at 1GHz = 403ms to 537ms.</p> <p><b>5'd4</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{26}</math> to <math>4 \times 2^{26}</math>. Timeout range in duration at 1GHz = 201ms to 268ms.</p> <p><b>5'd5</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{25}</math> to <math>4 \times 2^{25}</math>. Timeout range in duration at 1GHz = 101ms to 134ms.</p> <p><b>5'd6</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{24}</math> to <math>4 \times 2^{24}</math>. Timeout range in duration at 1GHz = 50ms to 67ms.</p> <p><b>5'd7</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{23}</math> to <math>4 \times 2^{23}</math>. Timeout range in duration at 1GHz = 25ms to 34ms.</p> <p><b>5'd8</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{22}</math> to <math>4 \times 2^{22}</math>. Timeout range in duration at 1GHz = 12.6ms to 17ms.</p> <p><b>5'd9</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{21}</math> to <math>4 \times 2^{21}</math>. Timeout range in duration at 1GHz = 6.3ms to 8.4ms.</p> <p><b>5'd10</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{20}</math> to <math>4 \times 2^{20}</math>. Timeout range in duration at 1GHz = 3.15ms to 4.2ms.</p> <p><b>5'd11</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{19}</math> to <math>4 \times 2^{19}</math>. Timeout range in duration at 1GHz = 1.6ms to 2.1ms.</p> <p><b>5'd12</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{18}</math> to <math>4 \times 2^{18}</math>. Timeout range in duration at 1GHz = 786us to 1.0ms.</p> <p><b>5'd13</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{17}</math> to <math>4 \times 2^{17}</math>. Timeout range in duration at 1GHz = 393us to 524us.</p> <p><b>5'd14</b></p> <p>Timeout range in clock cycles = <math>3 \times 2^{16}</math> to <math>4 \times 2^{16}</math>. Timeout range in duration at 1GHz = 196us to 262us.</p>	RW	0b00000

Bits	Name	Description	Type	Reset
[4:0]	hd_counter_init	<p><b>5'd15</b> Timeout range in clock cycles = <math>3 \times 2^{15}</math> to <math>4 \times 2^{15}</math>. Timeout range in duration at 1GHz = 98us to 131us.</p> <p><b>5'd16</b> Timeout range in clock cycles = <math>3 \times 2^{14}</math> to <math>4 \times 2^{14}</math>. Timeout range in duration at 1GHz = 49us to 65us.</p> <p><b>5'd17</b> Timeout range in clock cycles = <math>3 \times 2^{13}</math> to <math>4 \times 2^{13}</math>. Timeout range in duration at 1GHz = 25us to 33us.</p> <p><b>5'd18</b> Timeout range in clock cycles = <math>3 \times 2^{12}</math> to <math>4 \times 2^{12}</math>. Timeout range in duration at 1GHz = 12us to 16us.</p> <p><b>5'd19</b> Timeout range in clock cycles = <math>3 \times 2^{11}</math> to <math>4 \times 2^{11}</math>. Timeout range in duration at 1GHz = 6us to 8.2us.</p> <p><b>5'd20</b> Timeout range in clock cycles = <math>3 \times 2^{10}</math> to <math>4 \times 2^{10}</math>. Timeout range in duration at 1GHz = 3us to 4.1us.</p> <p><b>5'd21</b> Timeout range in clock cycles = <math>3 \times 2^9</math> to <math>4 \times 2^9</math>. Timeout range in duration at 1GHz = 1.5us to 2.0us.</p> <p><b>5'd22</b> Timeout range in clock cycles = <math>3 \times 2^8</math> to <math>4 \times 2^8</math>. Timeout range in duration at 1GHz = 768ns to 1.0us.</p> <p><b>5'd23</b> Timeout range in clock cycles = <math>3 \times 2^7</math> to <math>4 \times 2^7</math>. Timeout range in duration at 1GHz = 384ns to 512ns.</p> <p><b>5'd24</b> Timeout range in clock cycles = <math>3 \times 2^6</math> to <math>4 \times 2^6</math>. Timeout range in duration at 1GHz = 192ns to 256ns.</p> <p><b>5'd25</b> Timeout range in clock cycles = <math>3 \times 2^5</math> to <math>4 \times 2^5</math>. Timeout range in duration at 1GHz = 96ns to 128ns.</p> <p><b>5'd26</b> Timeout range in clock cycles = <math>3 \times 2^4</math> to <math>4 \times 2^4</math>. Timeout range in duration at 1GHz = 48ns to 64ns.</p> <p><b>5'd27</b> Timeout range in clock cycles = <math>3 \times 2^3</math> to <math>4 \times 2^3</math>. Timeout range in duration at 1GHz = 24ns to 32ns.</p> <p><b>5'd28</b> Illegal</p> <p><b>5'd29</b> Illegal</p> <p><b>5'd30</b> Illegal</p>	RW	0b00000



**Table 16-269: silicon\_debug bit descriptions**

Bits	Name	Description	Type	Reset
[31]	event_capture	Enable event capture	RW	0
[30:24]	reserved1	Bits within this register segment are reserved for future product development	RO	0b0000000
[23:16]	write_outstanding	Indicates that the interface has outstanding write requests.	RO	0x0
[15:5]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[4]	write_response_stall	Indicates HSNI write response channel stall event	RO	0
[3]	write_stall	Prior write address phase, HREADY LOW	RO	0
[2]	write_address_stall	Not implemented in the HSNI, tied to 0	RO	0
[1]	read_stall	Prior read address phase, HREADY LOW	RO	0
[0]	read_address_stall	Not implemented in the HSNI, tied to 0	RO	0

### 16.14.14 HSNI qosctl register

This register controls the QoS settings for BQV and TSPEC and enables a QoS value on inbound transactions to be overridden.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x84

##### Type

RW

##### Reset value

0x00000000

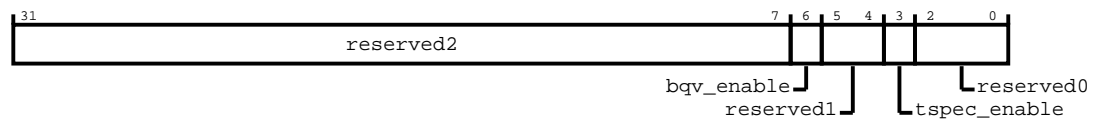
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the qosctl register bit assignments.

**Figure 16-256: Bit assignment diagram for the qosctl register**



The following table shows the qosctl register bit descriptions.

**Table 16-270: qosctl bit descriptions**

Bits	Name	Description	Type	Reset
[31:7]	reserved2	Bits within this register segment are reserved for future product development	RO	0x0
[6]	bqv_enable	Enables BQV. For BQV, both of the following conditions (in *soft BW Regulator Target Bandwidth register) are true: <ul style="list-style-type: none"> <li>BW_ALLOC &gt; 0</li> <li>QVMAX &gt; QVMIN</li> </ul>	RW	0
[5:4]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[3]	tspec_enable	Enables TSPEC. For TSPEC, the *Hard Bandwidth Regulator Average Rate must be > 0. Also, one of the following conditions must be true: <ul style="list-style-type: none"> <li>Peak regulation must be disabled, that is, *Hard Bandwidth Regulator Peak Rate = 0</li> <li>If peak regulation is enabled, both Hard Bandwidth Regulator Burstiness Allowance must be &gt; 0, and Hard Bandwidth Regulator Peak Rate must be &gt; *Hard Bandwidth Regulator Average Rate.</li> </ul>	RW	0
[2:0]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000

### 16.14.15 HSNi wdatthrs register

This register specifies the number of write data beats to be queued before the write packet is sent.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x88

##### Type

RW

##### Reset value

0x00000000

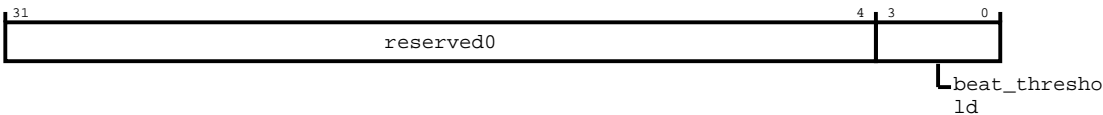
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the wdatthrs register bit assignments.

Figure 16-257: Bit assignment diagram for the wdatthrs register



The following table shows the wdatthrs register bit descriptions.

Table 16-271: wdatthrs bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3:0]	beat_threshold	Write data threshold decimal value. Specifies the number of write data beats to be buffered before the write data packet is sent.	RW	0b0000

16.14.16 HSNl qos\_values register

This register stores the value that is applied to GT transactions if the BQV regulator is not present or enabled.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x90

Type

RW

Reset value

0x00000000

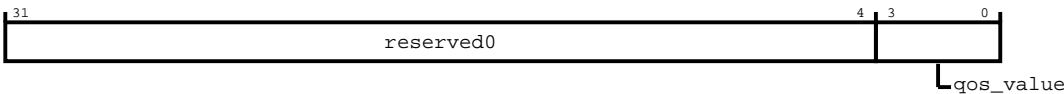
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qos\_values register bit assignments.

Figure 16-258: Bit assignment diagram for the qos\_values register



The following table shows the qos\_values register bit descriptions.

Table 16-272: qos\_values bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3:0]	qos_value	QOS value override for the completer interface	RW	0b0000

16.14.17 HSNl qosot register

Contains registers used for configuring the number of maximum outstanding transaction for the HSNl node.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xA0

Type

RW

Reset value

0x00000000



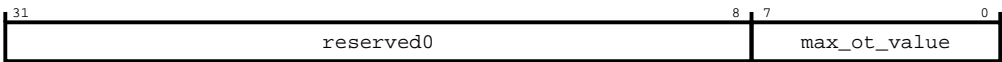
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qosot register bit assignments.

Figure 16-259: Bit assignment diagram for the qosot register



The following table shows the qosot register bit descriptions.

Table 16-273: qosot bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	max_ot_value	Configuration register to set the maximum outstanding transaction	RW	0x0

16.14.18 HSNl qoscompk register

This register controls the QoS peak rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xBC

Type

RW

Reset value

0x00000000

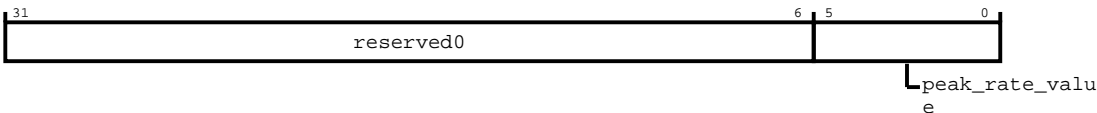
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscompk register bit assignments.

Figure 16-260: Bit assignment diagram for the qoscompk register



The following table shows the qoscompk register bit descriptions.

Table 16-274: qoscompk bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	peak_rate_value	The peak rate value of both read and write channels. The value is a binary fraction of the peak number of both read and write transfers for each cycle.	RW	0b000000

16.14.19 HSNl qoscombur register

This register controls the QoS burstiness allowance for combined read and write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC0

Type

RW

Reset value

0x00000000

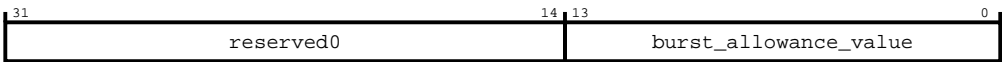
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscombur register bit assignments.

Figure 16-261: Bit assignment diagram for the qoscombur register



The following table shows the qoscombur register bit descriptions.

Table 16-275: qoscombur bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[13:0]	burst_allowance_value	Specifies the combined read and write TSPEC burstiness allowance	RW	0x0

16.14.20 HSNl qoscomavg register

This register controls the QoS average rate for both read and write hard bandwidth regulation, TSPEC, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xC4

Type

RW

Reset value

0x00000000

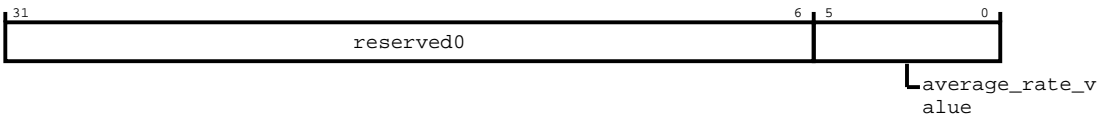
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscomavg register bit assignments.

Figure 16-262: Bit assignment diagram for the qoscomavg register



The following table shows the qoscomavg register bit descriptions.

Table 16-276: qoscomavg bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:0]	average_rate_value	The QoS average rate value of both read and write channels. The value is a binary fraction of the average number of both read and write transfers for each cycle.	RW	0b000000

16.14.21 HSNl qoscombqv register

This register controls the maximum and minimum QoS values, bandwidth allocation, burstiness, and overspend for both read and write soft bandwidth regulation, BQV, of a completer interface.

Configurations

This register is only present if QoS bandwidth regulation is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xD0

Type

RW

Reset value

0x00000000

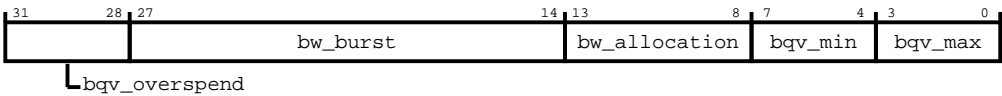
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the qoscombqv register bit assignments.

Figure 16-263: Bit assignment diagram for the qoscombqv register



The following table shows the qoscombqv register bit descriptions.

Table 16-277: qoscombqv bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	bqv_overspend	BQV overspend. The excess number of full data bus transfers permitted	RW	0b0000
[27:14]	bw_burst	Bandwidth burstiness. The excess number of full data bus transfers to permit as burstiness allowance.	RW	0x0
[13:8]	bw_allocation	Bandwidth allocation. The proportion of data bus width for bandwidth allocation.	RW	0b000000
[7:4]	bqv_min	BQV minimum QoS value. The minimum value of AxQoS.	RW	0b0000
[3:0]	bqv_max	BQV maximum QoS value. The maximum value of AxQoS.	RW	0b0000

16.14.22 HSNi mpam\_control register

If GT\_MPAM\_SUPPORT is enabled, the register drives the MPAM values for a specific HSNi.

Configurations

This register is only present if support for MPAM is enabled on the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xE0

Type

RW

Reset value

0x00000000

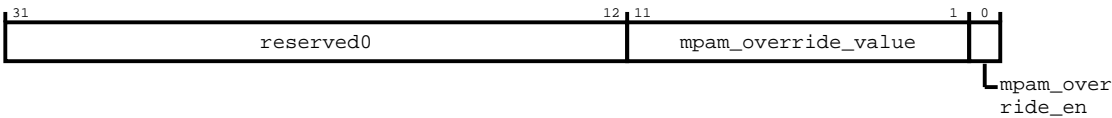
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the mpam\_control register bit assignments.

Figure 16-264: Bit assignment diagram for the mpam\_control register



The following table shows the mpam\_control register bit descriptions.

Table 16-278: mpam\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:12]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[11:1]	mpam_override_value	MPAM override value	RW	0x0
[0]	mpam_override_en	For AHB interfaces, the MPAM override value is always used if GT_MPAM_SUPPORT is enabled irrespective of this bit value.	RW	0

16.14.23 HSNi interrupt\_status register

This register indicates the interrupt status of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF0

Type

RW

Reset value

0x00000000

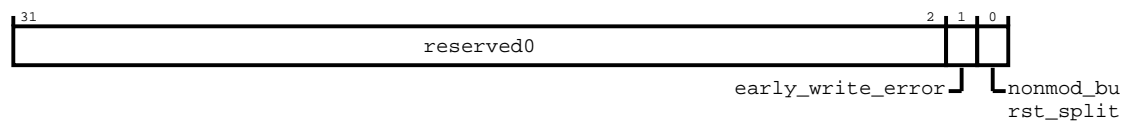
## Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the interrupt\_status register bit assignments.

**Figure 16-265: Bit assignment diagram for the interrupt\_status register**



The following table shows the interrupt\_status register bit descriptions.

### Table 16-279: interrupt\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	early_write_error	HSNI implements an interrupt mechanism to signal imprecise errors that are detected on actual write responses received for requests for which early write responses were already provided.  Write 1 to clear.	RW	0
[0]	nonmod_burst_split	If there is a burst split, an interrupt is generated if a non-modifiable transaction is split.  Write 1 to clear.	RW	0

#### 16.14.24 HSN1 interrupt\_mask register

This register is the interrupt mask of Secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0xF4

## Type

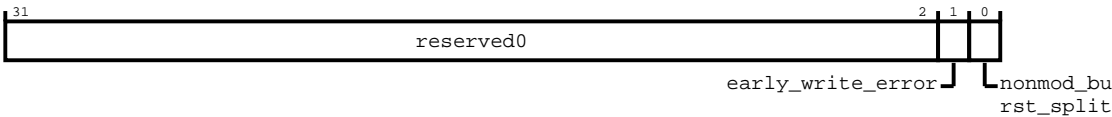
RW

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**  
The following figure shows the interrupt\_mask register bit assignments.

**Figure 16-266: Bit assignment diagram for the interrupt\_mask register**



The following table shows the interrupt\_mask register bit descriptions.

**Table 16-280: interrupt\_mask bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	early_write_error	Mask the early write response with imprecise error interrupt.	RW	0
[0]	nonmod_burst_split	Mask the non-modifiable burst split interrupt.	RW	0

16.14.25 HSNi interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0xF8

**Type**  
RW

**Reset value**  
0x00000000



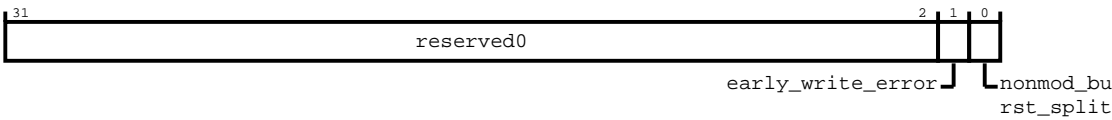
Constraints

None.

Bit descriptions

The following figure shows the interrupt\_status\_ns register bit assignments.

Figure 16-267: Bit assignment diagram for the interrupt\_status\_ns register



The following table shows the interrupt\_status\_ns register bit descriptions.

Table 16-281: interrupt\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	early_write_error	HSNI implements an interrupt mechanism to signal imprecise errors that are detected on actual write responses received for requests for which early write responses were already provided.  Write 1 to clear.	RW	0
[0]	nonmod_burst_split	If there is a burst split, an interrupt is generated if a non-modifiable transaction is split.  Write 1 to clear.	RW	0

16.14.26 HSNI interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFC

Type

RW

Reset value

0x00000000

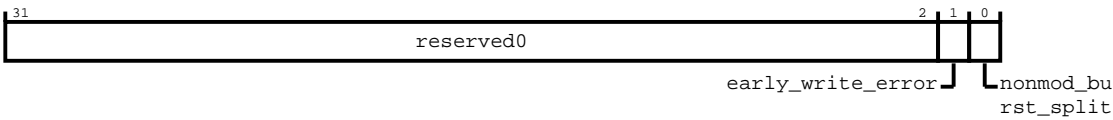
Constraints

None.

Bit descriptions

The following figure shows the interrupt\_mask\_ns register bit assignments.

Figure 16-268: Bit assignment diagram for the interrupt\_mask\_ns register



The following table shows the interrupt\_mask\_ns register bit descriptions.

Table 16-282: interrupt\_mask\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	early_write_error	Mask the early write response with imprecise error interrupt	RW	0
[0]	nonmod_burst_split	Mask the non-modifiable burst split interrupt.	RW	0

16.14.27 HSNi idm\_device\_id register

This register indicates the statically configured device ID value and is implemented if IDM is enabled.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x100

Type

RO

Reset value

See individual bit resets.

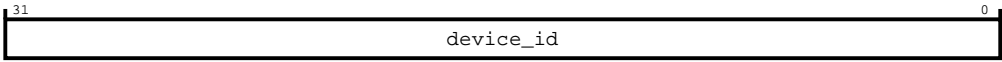
Constraints

None.

Bit descriptions

The following figure shows the `idm_device_id` register bit assignments.

Figure 16-269: Bit assignment diagram for the `idm_device_id` register



The following table shows the `idm_device_id` register bit descriptions.

Table 16-283: `idm_device_id` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	device_id	Returns statically configured ID value	RO	Configuration dependent

16.14.28 HSNl `idm_config` register

This register enables transaction logging, error detection, timeout detection, access control, and reset control.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x104

Type

RW

Reset value

See individual bit resets.

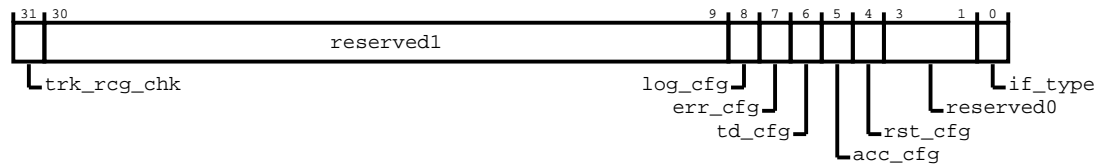
Constraints

None.

Bit descriptions

The following figure shows the `idm_config` register bit assignments.

**Figure 16-270: Bit assignment diagram for the idm\_config register**



The following table shows the idm\_config register bit descriptions.

**Table 16-284: idm\_config bit descriptions**

Bits	Name	Description	Type	Reset
[31]	trk_rcg_chk	Tracker Regional Clock Gating (RCG) chicken bit	RW	0
[30:9]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[8]	log_cfg	Transaction logging present	RO	1
[7]	err_cfg	Error detection present	RO	1
[6]	td_cfg	Timeout detection present	RO	1
[5]	acc_cfg	Access control present	RO	1
[4]	rst_cfg	Reset control present	RO	1
[3:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[0]	if_type	Interface type <b>0</b> Completer <b>1</b> Requester	RO	Configuration dependent

### 16.14.29 HSNi idm\_errctlr register

This register controls how errors are handled.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x108

##### Type

RW

## Reset value

0x00000000

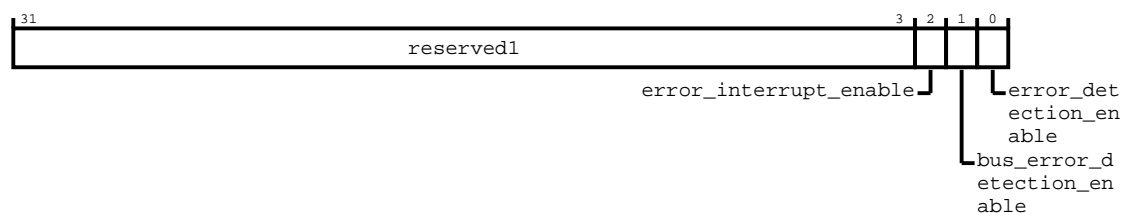
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the idm\_errctlr register bit assignments.

**Figure 16-271: Bit assignment diagram for the idm\_errctlr register**



The following table shows the idm\_errctlr register bit descriptions.

**Table 16-285: idm\_errctlr bit descriptions**

Bits	Name	Description	Type	Reset
[31:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[2]	error_interrupt_enable	Enable error interrupt for uncorrected error as indicated by IDM_ERRSTATUS.UE fields	RW	0
[1]	bus_error_detection_enable	<p>Enable bus error detection</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when an error is detected and idm_errctlr [ed] is enabled. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set. An error interrupt event is generated (unless masked).</p>	RW	0
[0]	error_detection_enable	<p>Error detection global enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when an error is detected. In other words, a timeout error or bus error is detected and its respective detection enable register bit, Timeout_control[TD_EN], or idm_errctlr[be] is also set. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>An error interrupt event is generated, unless masked.</p>	RW	0

16.14.30 HSNi idm\_errstatus register

This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x110

Type

RW

Reset value

0x00000000

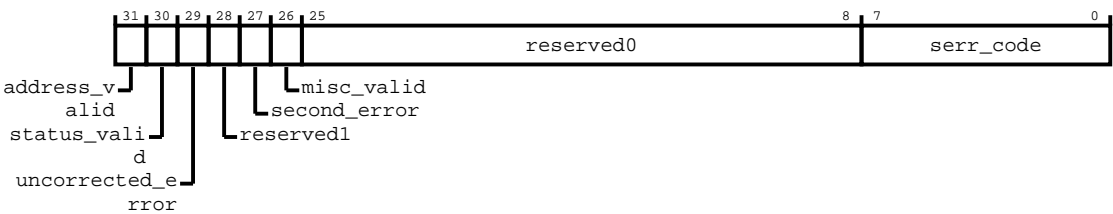
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_errstatus register bit assignments.

Figure 16-272: Bit assignment diagram for the idm\_errstatus register



The following table shows the idm\_errstatus register bit descriptions.

**Table 16-286: idm\_errstatus bit descriptions**

Bits	Name	Description	Type	Reset
[31]	address_valid	<p>Address valid. The values are:</p> <p><b>0</b></p> <p>ERRADDR is not valid.</p> <p><b>1</b></p> <p>ERRADDR contains an address that is associated with the highest priority error which this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1 and is not cleared to zero in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[30]	status_valid	<p>Status register is valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRSTATUS not valid</p> <p><b>1</b></p> <p>IDM_ERRSTATUS valid. At least one error has been recorded.</p> <p>This bit ignores writes if any of the following fields is set to 1 and is not being cleared to zero in the same write:</p> <ul style="list-style-type: none"> <li>IDM_ERRSTATUS.UE</li> <li>IDM_ERRSTATUS.AV</li> <li>IDM_ERRSTATUS.OF * IDM_ERRSTATUS.MV</li> </ul> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred</p> <p>This bit ignores writes if IDM_ERRSTATUS.OF is set to 1 and is not being cleared to zero in the same write. This bit is not valid and reads <b>UNKNOWN</b> if IDM_ERRSTATUS.V is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0

Bits	Name	Description	Type	Reset
[27]	second_error	Returns whether a second error has been received while handling a first error. The values are:  <b>1</b> Second error received  <b>0</b> No other error received  This bit is read, or write 1 to clear  Write 1 to clear.	RW	0
[26]	misc_valid	Miscellaneous registers valid. The values are:  <b>0</b> IDM_ERRMISC0 and IDM_ERRMISC1 not valid  <b>1</b> The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISC0 and IDM_ERRMISC1 registers contains additional information for an error that this record records.  This bit ignores writes if IDM_ERRSTATUS.UE is set to 1, and is not being cleared to 0 in the same write. This bit is a read, or write 1 to clear.  Write 1 to clear.	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	Primary error code. Indicates the type of error. The values are:  <b>00</b> No error  <b>13</b> Illegal address - decode error  <b>18</b> Error response from completer  <b>20</b> Internal timeout	RO	0x0

### 16.14.31 HSNl idm\_erraddr\_lsb register

This register is the error log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit



Address offset

0x114

Type

RO

Reset value

0x00000000

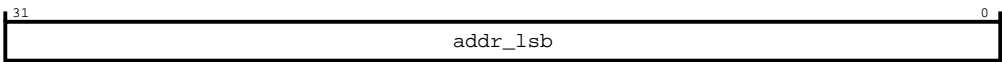
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_lsb` register bit assignments.

Figure 16-273: Bit assignment diagram for the `idm_erraddr_lsb` register



The following table shows the `idm_erraddr_lsb` register bit descriptions.

Table 16-287: `idm_erraddr_lsb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.14.32 HSNl `idm_erraddr_msb` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x118

Type

RO

Reset value

0x00000000

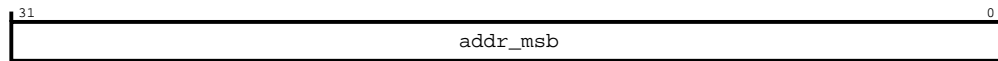
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_erraddr_msb` register bit assignments.

**Figure 16-274: Bit assignment diagram for the `idm_erraddr_msb` register**



The following table shows the `idm_erraddr_msb` register bit descriptions.

### Table 16-288: idm\_erraddr\_msb bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

### 16.14.33 HSN1 idm\_errmisc0 register

This register is the error log of Secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x128

## Type

RO

## Reset value

0x00000000

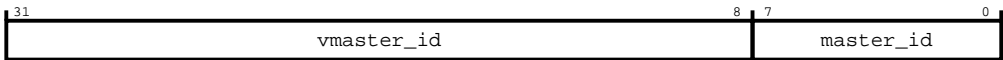
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_errmisc0` register bit assignments.

Figure 16-275: Bit assignment diagram for the `idm_errmisc0` register



The following table shows the `idm_errmisc0` register bit descriptions.

Table 16-289: `idm_errmisc0` bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AxiD into ASNI of the transaction causing an error. The assumption here is there is no manipulation of incoming AXI AxiD in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.14.34 HSNi `idm_errmisc1` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x12C

Type

RO

Reset value

0x00000000

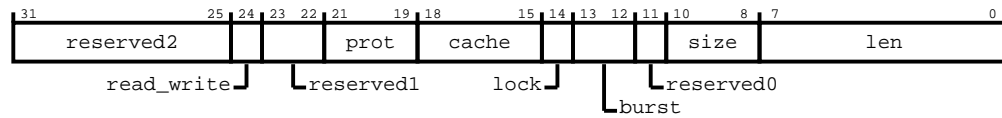
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_errmisc1` register bit assignments.

**Figure 16-276: Bit assignment diagram for the idm\_errmisc1 register**



The following table shows the idm\_errmisc1 register bit descriptions.

**Table 16-290: idm\_errmisc1 bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	The AXI read or write information of a transaction causing an error  1 Write 0 Read	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	The AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	The AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	The AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	The AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	The AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	The AXI len information of a transaction causing an error.	RO	0x0

### 16.14.35 HSNi idm\_access\_control register

This register controls the state, gated or ungated, of a device.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x130

##### Type

RW

**Reset value**

0x00000000

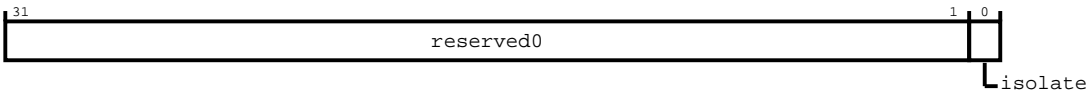
**Constraints**

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**

The following figure shows the idm\_access\_control register bit assignments.

**Figure 16-277: Bit assignment diagram for the idm\_access\_control register**



The following table shows the idm\_access\_control register bit descriptions.

**Table 16-291: idm\_access\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	isolate	Perform gating off a device. Reading 1 indicates that the completer device is gated or isolated. Reading 0 indicates that the completer device is ungated or de-isolated. Write 1 to enter gated state. Write 0 to exit gated state. There is some delay to updating this field with the intended write value. Exit from gated state is only successful if there are no outstanding transactions and all error status register bits are cleared. Entry into gated state is only successful if there are no outstanding transactions. While in pending isolation entry state or in active isolation state, a write of 1 to this bit causes reentry to isolation state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 0 is ignored. While in pending isolation exit state, a write of 0 to this bit causes a re-exit to the exit state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS, and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 1 is ignored.	RW	0

**16.14.36 HSNi idm\_access\_status register**

This register indicates the access status for Secure transactions.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**

32-bit



Bits	Name	Description	Type	Reset
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00000
[2]	resp_method	Indicates device generates errors in gated access	RO	0
[1]	access_method	Wait for all outstanding to complete, then block input	RO	1
[0]	reserved0	Bits within this register segment are reserved for future product development	RO	0

### 16.14.37 HSNi idm\_access\_readid register

This register is the access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x138

##### Type

RO

##### Reset value

0x00000000

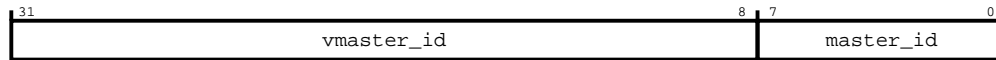
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_access\_readid register bit assignments.

**Figure 16-279: Bit assignment diagram for the `idm_access_readid` register**



The following table shows the `idm_access_readid` register bit descriptions.

**Table 16-293: `idm_access_readid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming signal into the endpoint of the first transaction to arrive after isolation when the <code>active_read</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the ARID of the transaction on ASNI and contains the HMASTER on HSNi. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNi of the first transaction to arrive after isolation when the <code>active_read</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

### 16.14.38 HSNi `idm_access_writeid` register

This register is the access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x13C

##### Type

RO

##### Reset value

0x00000000

#### Constraints

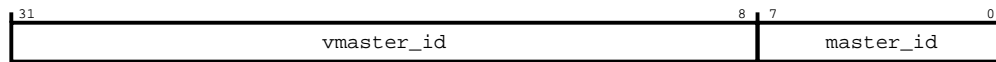
Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the `idm_access_writeid` register bit assignments.



**Figure 16-280: Bit assignment diagram for the `idm_access_writeid` register**



The following table shows the `idm_access_writeid` register bit descriptions.

**Table 16-294: `idm_access_writeid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AWID signal into the endpoint of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

### 16.14.39 HSNI `idm_reset_control` register

This register controls the reset of a device that is attached to the interconnect.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x140

##### Type

RW

##### Reset value

0x00000002

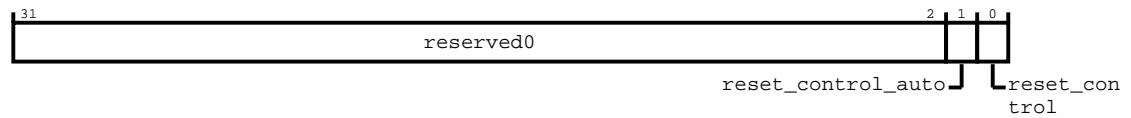
#### Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the `idm_reset_control` register bit assignments.

**Figure 16-281: Bit assignment diagram for the idm\_reset\_control register**



The following table shows the idm\_reset\_control register bit descriptions.

**Table 16-295: idm\_reset\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	reset_control_auto	<p>Configures the device for auto or internal reset mode. For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. There are several constraints on this field:</p> <ul style="list-style-type: none"> <li>You can only change this field during initialization or when the interface is fully quiesced. * Arm does not support changing this field while the interface is active. If you change this field during runtime, behavior is <b>UNPREDICTABLE</b>.</li> </ul> <p>Reads have the following effect:</p> <p><b>1</b></p> <p>A read of 1 indicates that the device is in auto or internal reset mode.</p> <p><b>0</b></p> <p>A read of 0 indicates that the device is not in auto or internal reset mode.</p> <p>Writes have the following effect:</p> <p><b>1</b></p> <p>A write of 1 configures the device for auto or internal reset mode.</p> <p><b>0</b></p> <p>A write of 0 disables auto or internal reset mode.</p> <p>For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. Bit[1] of the IDM_RESET_CONTROL register is 1 out of reset. This bit enables internal recovery mode out of reset. When not in auto reset mode and a timeout is detected, a write of 1 to the IDM_RESET_CONTROL.reset field initiates internal recovery mode. Changing this bit while the interface is not in idle mode results in <b>UNPREDICTABLE</b> behavior.</p>	RO	1

Bits	Name	Description	Type	Reset
[0]	reset_control	<p>Performs soft reset of attached device. If the auto bit is set to 1 the network interface gates the external interface, however the soft reset pin is not activated. If the auto bit is 0, the interfaces are not gated until there is a write to bit[0]. In this case, the soft reset pin is activated. Writes have the following effect:</p> <p><b>1</b></p> <p>Request the attached device to enter reset. If the write occurs before soft reset exit has occurred, the write is ignored.</p> <p><b>0</b></p> <p>Request the attached device to exit reset. If the write occurs before soft reset entry has occurred, the write is ignored.</p> <p>Software polls this register to determine if soft reset entry or exit has occurred, using the following values:</p> <p><b>1</b></p> <p>Indicates that the device is in reset.</p> <p><b>0</b></p> <p>Indicates that the device is not in reset.</p> <p>This register value updates to reflect a request for reset entry or reset exit, but the update can only occur after required internal conditions are met. Until these conditions are met, a read to this register returns the old value. For example, outstanding transactions currently being handled must complete before this register value updates. To ensure reset propagation within the device, it is the responsibility of the software to permit enough cycles after soft reset assertion is reflected in the IDM_RESET_CONTROL register before exiting soft reset by triggering a write of 0. If this responsibility is not met, the behavior is <b>UNDEFINED</b> or <b>UNPREDICTABLE</b>. When this register value is 1, the external soft reset pin that connects to the attached AXI requester or completer device is asserted, using the correct polarity of the reset pin. When this register value is 0, the external soft reset pin that connects to the attached AXI requester or completer device is deasserted, using the correct polarity of the reset pin. When in pending soft reset entry state or in active soft reset state, a write of 1 to this bit causes reentry to soft reset state. This write causes the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers to be cleared. A write of 0 is ignored. While in pending soft reset exit state, a write of 0 to this bit causes re-exit to exit state. A write of 0 also clears the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers. A write of 1 is ignored.</p>	RW	0

#### 16.14.40 HSNI idm\_reset\_status register

This register indicates mostly the reset status of Secure transactions. However, the rst\_exit\_state field indicates reset exit state of secure or non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

## Address offset

0x144

## Type

RO

## Reset value

0x00000000

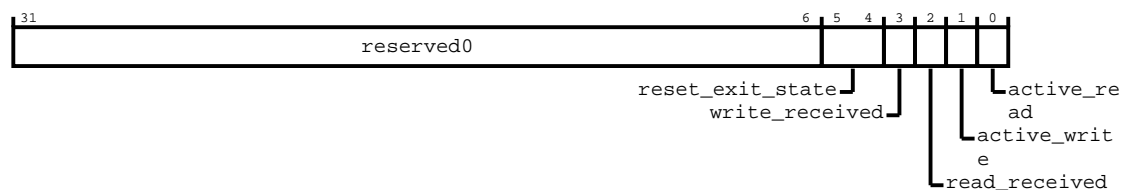
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the idm\_reset\_status register bit assignments.

**Figure 16-282: Bit assignment diagram for the idm\_reset\_status register**



The following table shows the idm\_reset\_status register bit descriptions.

**Table 16-296: idm\_reset\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:4]	reset_exit_state	Reset exit state  <b>00</b> Reset exit or entry is successful or not in reset state  <b>01</b> Reset exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  <b>10</b> Reset exit is unsuccessful or pending because of outstanding transactions  <b>11</b> Reset exit is unsuccessful or pending because of both uncleared error status bits and outstanding transactions	RO	0b00
[3]	write_received	A 1 indicates that an active Secure write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0

### 16.14.41 HSNl idm\_reset\_readid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x148

##### Type

RO

##### Reset value

0x00000000

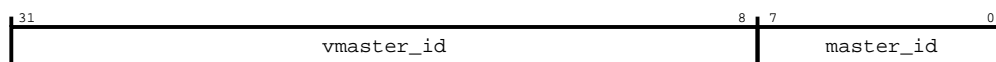
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_readid register bit assignments.

**Figure 16-283: Bit assignment diagram for the idm\_reset\_readid register**



The following table shows the idm\_reset\_readid register bit descriptions.

**Table 16-297: idm\_reset\_readid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

## 16.14.42 HSNI idm\_reset\_writeid register

This register is the reset access log of Secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x14C

#### Type

RO

#### Reset value

0x00000000

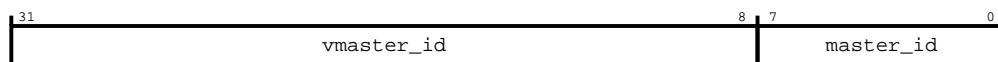
### Constraints

Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the idm\_reset\_writeid register bit assignments.

**Figure 16-284: Bit assignment diagram for the idm\_reset\_writeid register**



The following table shows the idm\_reset\_writeid register bit descriptions.

**Table 16-298: idm\_reset\_writeid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.14.43 HSNI idm\_timeout\_control register

This register is present when timeout detection is configured.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x150

##### Type

RW

##### Reset value

0x00000000

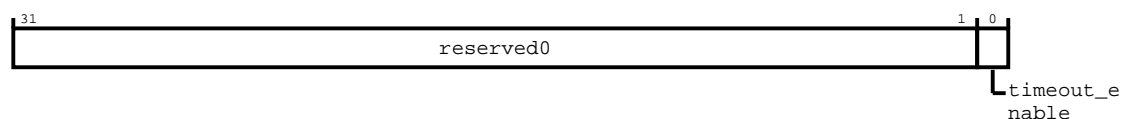
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the idm\_timeout\_control register bit assignments.

**Figure 16-285: Bit assignment diagram for the idm\_timeout\_control register**



The following table shows the idm\_timeout\_control register bit descriptions.

**Table 16-299: idm\_timeout\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	timeout_enable	<p>Timeout detection enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when a timeout is detected. The timeout is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>A timeout interrupt event is generated, unless it is masked.</p>	RW	0

#### 16.14.44 HSNi idm\_timeout\_value register

This register controls the duration that is used to determine if a transaction has timed out.

##### Configurations

This register is available in all configurations.

##### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x154

##### Type

RW

##### Reset value

0x00000004

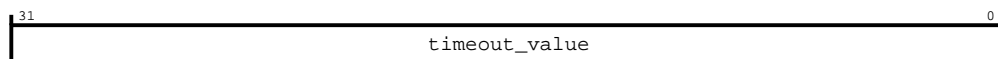
##### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

##### Bit descriptions

The following figure shows the idm\_timeout\_value register bit assignments.

**Figure 16-286: Bit assignment diagram for the idm\_timeout\_value register**





The following table shows the `idm_timeout_value` register bit descriptions.

**Table 16-300: `idm_timeout_value` bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	<code>timeout_value</code>	Controls the duration that is used to determine if a transaction has timed out. The actual duration is $2^{\text{timeout\_exponent}}$ cycles. The minimum value is 4. Values of 0, 1, 2, or 3 are treated as 4. The maximum value is 30. Values greater than 30 are treated as 30.	RW	0x4

16.14.45 HSNi `idm_interrupt_status` register

This register indicates the interrupt status of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x158

Type

RW

Reset value

0x00000000

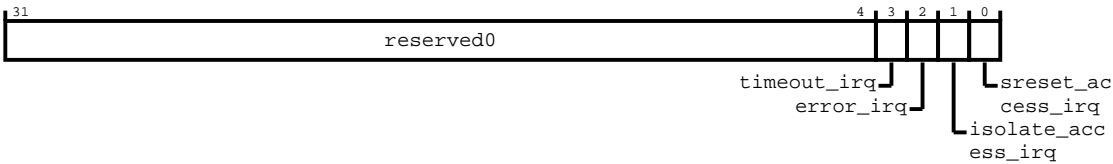
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_interrupt_status` register bit assignments.

**Figure 16-287: Bit assignment diagram for the `idm_interrupt_status` register**



The following table shows the `idm_interrupt_status` register bit descriptions.

**Table 16-301: idm\_interrupt\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.14.46 HSNi idm\_interrupt\_mask register

This register is the interrupt mask of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x15C

##### Type

RW

##### Reset value

0x00000000

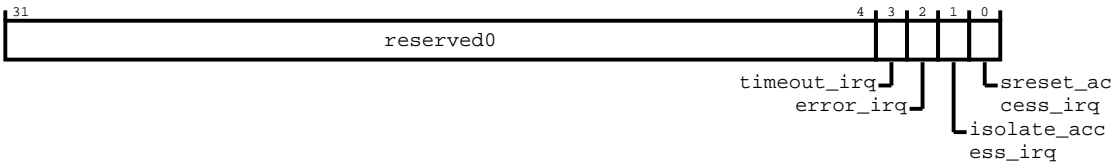
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask register bit assignments.

**Figure 16-288: Bit assignment diagram for the idm\_interrupt\_mask register**



The following table shows the idm\_interrupt\_mask register bit descriptions.

**Table 16-302: idm\_interrupt\_mask bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

16.14.47 HSNi idm\_errstatus\_ns register

This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x160

Type

RW

Reset value

0x00000000

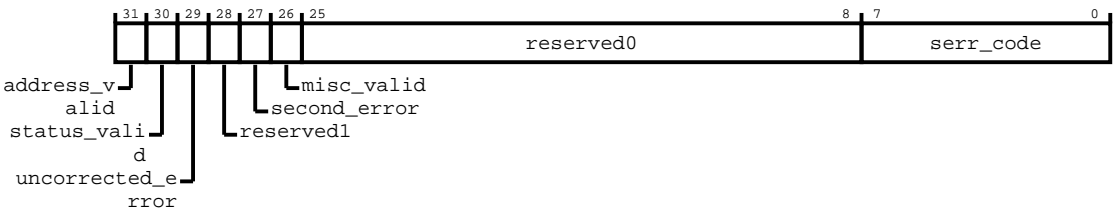
Constraints

None.

Bit descriptions

The following figure shows the idm\_errstatus\_ns register bit assignments.

Figure 16-289: Bit assignment diagram for the idm\_errstatus\_ns register



The following table shows the idm\_errstatus\_ns register bit descriptions.

Table 16-303: idm\_errstatus\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31]	address_valid	Address valid. The values are:  <b>0</b>  ERRADDR is not valid.  <b>1</b>  ERRADDR contains an address that is associated with the highest priority error that this record captures.  This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1 and is not cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0
[30]	status_valid	Status register valid. The values are:  <b>0</b>  IDM_ERRSTATUS_NS is not valid.  <b>1</b>  IDM_ERRSTATUS_NS is valid. At least one error has been recorded.  This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred.</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred.</p> <p>This bit ignores writes if the oe field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is not valid and reads <b>UNKNOWN</b> if the v field of the IDM_ERRSTATUS_NS register is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISCO_NS and IDM_ERRMISC1_NS are not valid.</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISCO_NS and IDM_ERRMISC1_NS registers contains additional information for an error that this record captures.</p> <p>This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1, and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	<p>Primary error code, indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0

16.14.48 HSNi idm\_erraddr\_lsb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x164

Type

RO

Reset value

0x00000000

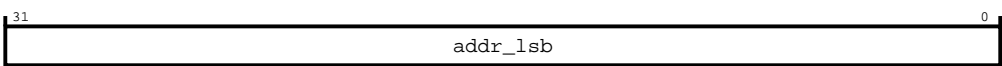
Constraints

None.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb\_ns register bit assignments.

Figure 16-290: Bit assignment diagram for the idm\_erraddr\_lsb\_ns register



The following table shows the idm\_erraddr\_lsb\_ns register bit descriptions.

Table 16-304: idm\_erraddr\_lsb\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.14.49 HSNi idm\_erraddr\_msb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x168

Type

RO

Reset value

0x00000000

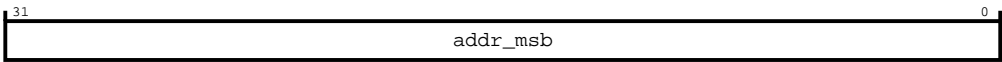
Constraints

None.

Bit descriptions

The following figure shows the `idm_erraddr_msb_ns` register bit assignments.

Figure 16-291: Bit assignment diagram for the `idm_erraddr_msb_ns` register



The following table shows the `idm_erraddr_msb_ns` register bit descriptions.

Table 16-305: `idm_erraddr_msb_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.14.50 HSNi `idm_errmisc0_ns` register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x178

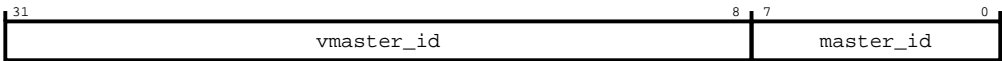
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the idm\_errmisc0\_ns register bit assignments.

**Figure 16-292: Bit assignment diagram for the idm\_errmisc0\_ns register**



The following table shows the idm\_errmisc0\_ns register bit descriptions.

**Table 16-306: idm\_errmisc0\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming AXI AxID into ASNI of the transaction causing an error. The assumption is no manipulation of incoming AXI AxID in ASNI.	RO	0x0
[7:0]	master_id	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.14.51 HSNi idm\_errmisc1\_ns register

This register is the error log of Non-secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x17C

**Type**  
RO

**Reset value**  
0x00000000



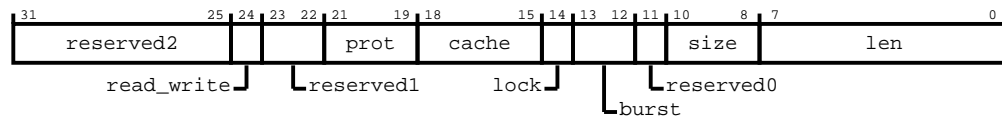
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_errmisc1_ns` register bit assignments.

**Figure 16-293: Bit assignment diagram for the `idm_errmisc1_ns` register**



The following table shows the `idm_errmisc1_ns` register bit descriptions.

Table 16-307: idm\_errmisc1\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	Returns the AXI read or write information of a transaction causing an error:  <div> <div>1</div> <div>Write</div> </div> <div> <div>0</div> <div>Read</div> </div>	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	Returns the AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	Returns the AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	Returns the AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	Returns the AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	Returns the AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	Returns the AXI len information of a transaction causing an error.	RO	0x0

#### 16.14.52 HSN1 idm\_access\_status\_ns register

This register indicates the access status for Non-secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x184

## Type

RO

## Reset value

0x00000000

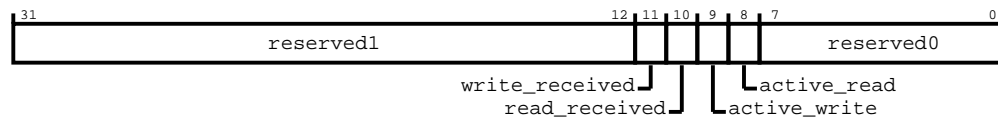
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_access_status_ns` register bit assignments.

**Figure 16-294: Bit assignment diagram for the `idm_access_status_ns` register**



The following table shows the `idm_access_status_ns` register bit descriptions.

**Table 16-308: `idm_access_status_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved1	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:0]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0

16.14.53 HSNi idm\_access\_readid\_ns register

This register is the access log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x188

Type

RO

Reset value

0x00000000

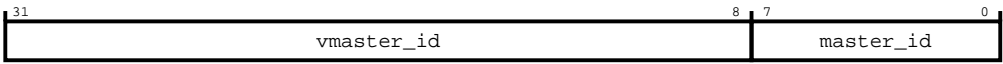
Constraints

None.

Bit descriptions

The following figure shows the idm\_access\_readid\_ns register bit assignments.

Figure 16-295: Bit assignment diagram for the idm\_access\_readid\_ns register



The following table shows the idm\_access\_readid\_ns register bit descriptions.

Table 16-309: idm\_access\_readid\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNi. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNi of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

## 16.14.54 HSNi idm\_access\_writeid\_ns register

This register is the access log of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x18C

#### Type

RO

#### Reset value

0x00000000

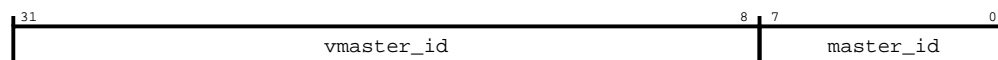
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_access\_writeid\_ns register bit assignments.

**Figure 16-296: Bit assignment diagram for the idm\_access\_writeid\_ns register**



The following table shows the idm\_access\_writeid\_ns register bit descriptions.

**Table 16-310: idm\_access\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the IDM_ACCESS_STATUS_NS register field active_write is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNi. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNi of the first transaction to arrive after isolation when the active_write field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

16.14.55 HSNl idm\_reset\_status\_ns register

This register indicates the reset status of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x194

Type

RO

Reset value

0x00000000

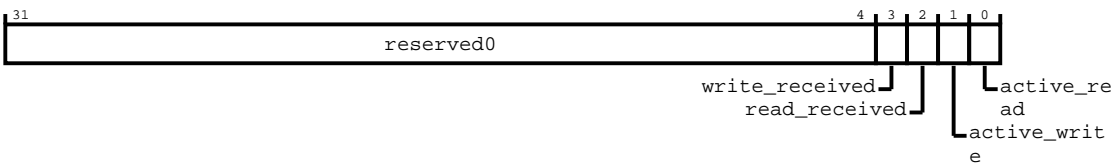
Constraints

None.

Bit descriptions

The following figure shows the idm\_reset\_status\_ns register bit assignments.

Figure 16-297: Bit assignment diagram for the idm\_reset\_status\_ns register



The following table shows the idm\_reset\_status\_ns register bit descriptions.

Table 16-311: idm\_reset\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[3]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"><li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li><li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li></ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to 0 on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates that there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates that there is at least one read transaction currently in progress.	RO	0

### 16.14.56 HSNi idm\_reset\_readid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x198

##### Type

RO

##### Reset value

0x00000000

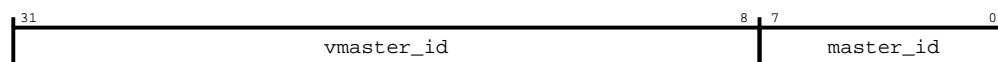
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_readid\_ns register bit assignments.

**Figure 16-298: Bit assignment diagram for the idm\_reset\_readid\_ns register**



The following table shows the idm\_reset\_readid\_ns register bit descriptions.

**Table 16-312: idm\_reset\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

### 16.14.57 HSNI idm\_reset\_writeid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x19C

##### Type

RO

##### Reset value

0x00000000

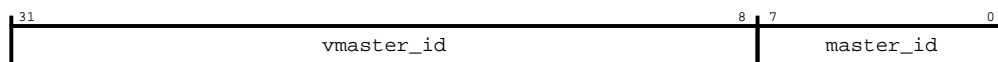
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid\_ns register bit assignments.

**Figure 16-299: Bit assignment diagram for the idm\_reset\_writeid\_ns register**



The following table shows the idm\_reset\_writeid\_ns register bit descriptions.

**Table 16-313: idm\_reset\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when active_write field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

## 16.14.58 HSNI idm\_interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1A8

#### Type

RW

#### Reset value

0x00000000

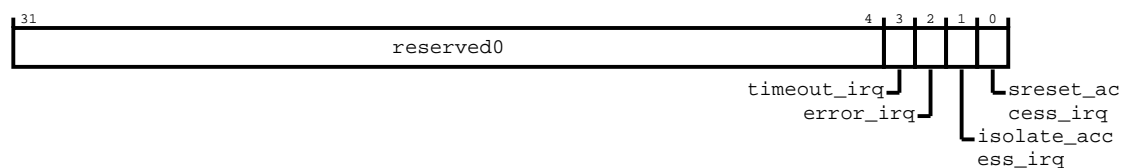
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_status\_ns register bit assignments.

**Figure 16-300: Bit assignment diagram for the idm\_interrupt\_status\_ns register**



The following table shows the idm\_interrupt\_status\_ns register bit descriptions.



**Table 16-314: idm\_interrupt\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.14.59 HSNi idm\_interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x1AC

##### Type

RW

##### Reset value

0x00000000

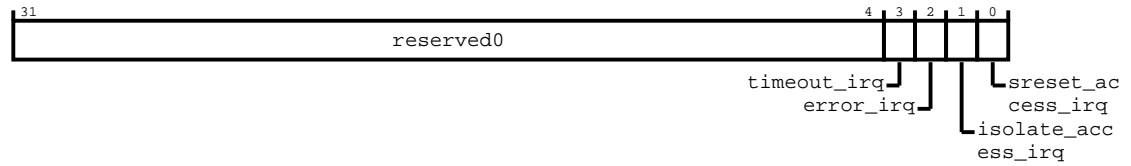
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask\_ns register bit assignments.

**Figure 16-301: Bit assignment diagram for the `idm_interrupt_mask_ns` register**



The following table shows the `idm_interrupt_mask_ns` register bit descriptions.

**Table 16-315: `idm_interrupt_mask_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

## 16.15 HMNI register summary

This section describes the HMNI registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-316: HMNI register summary**

Offset	Name	Type	Reset	Width	Description
0x00	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a node for HMNI registers.
0x04	<a href="#">node_info</a>	RO	See individual bit resets.	32-bit	This register provides information about the HMNI node features and configuration.
0x40	<a href="#">node_features</a>	RO	0x00000000	32-bit	This register configures the node features.
0x08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	Contains register bits that are used for configuring the Secure access behavior.
0x44	<a href="#">node_control</a>	RW	See individual bit resets.	32-bit	Contains registers used for configuring the behavior of the HMNI node.
0x0C	<a href="#">pmusela</a>	RW	0x00000000	32-bit	This register is used to select the event values in the HMNI event crossbar.
0x10	<a href="#">pmuselb</a>	RW	0x00000000	32-bit	This register is used to select the event values in the HMNI event crossbar.
0x14	<a href="#">interface_id_0_3</a>	RO	See individual bit resets.	32-bit	Contains information about the HMNI interface IDs for interfaces 0-3.
0x24	<a href="#">num_sub_features</a>	RO	See individual bit resets.	32-bit	Number of subfeatures.

Offset	Name	Type	Reset	Width	Description
0x28	sub_feature_0_type	RO	See individual bit resets.	32-bit	Subfeature 0 type.
0x2C	sub_feature_0_pointer	RO	See individual bit resets.	32-bit	Subfeature 0 pointer.
0x80	silicon_debug	RW	0x00000000	32-bit	This register monitors the status of requester interface channels.
0xF0	interrupt_status	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0xF4	interrupt_mask	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0xF8	interrupt_status_ns	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0xFC	interrupt_mask_ns	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.
0x100	idm_device_id	RO	See individual bit resets.	32-bit	This register indicates the statically configured device ID value and is implemented if IDM is enabled.
0x104	idm_config	RW	See individual bit resets.	32-bit	This register enables transaction logging, error detection, timeout detection, access control, and reset control.
0x108	idm_errctlr	RW	0x00000000	32-bit	This register controls how errors are handled.
0x110	idm_errstatus	RW	0x00000000	32-bit	This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.
0x114	idm_erraddr_lsb	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x118	idm_erraddr_msb	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x128	idm_errmisc0	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x12C	idm_errmisc1	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x130	idm_access_control	RW	0x00000000	32-bit	This register controls the state, gated or ungated, of a device.
0x134	idm_access_status	RO	0x00000002	32-bit	This register indicates the access status for Secure transactions.
0x138	idm_access_readid	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x13C	idm_access_writeid	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x140	idm_reset_control	RW	0x00000002	32-bit	This register controls the reset of a device that is attached to the interconnect.
0x144	idm_reset_status	RO	0x00000000	32-bit	This register indicates mostly the reset status of Secure transactions. However, the rst_exit_state field indicates reset exit state of secure or non-secure transactions.
0x148	idm_reset_readid	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x14C	idm_reset_writeid	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x150	idm_timeout_control	RW	0x00000000	32-bit	This register is present when timeout detection is configured.
0x154	idm_timeout_value	RW	0x00000004	32-bit	This register controls the duration that is used to determine if a transaction has timed out.
0x158	idm_interrupt_status	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0x15C	idm_interrupt_mask	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0x160	idm_errstatus_ns	RW	0x00000000	32-bit	This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.
0x164	idm_erraddr_lsb_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x168	idm_erraddr_msb_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x178	idm_errmisc0_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x17C	idm_errmisc1_ns	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.

Offset	Name	Type	Reset	Width	Description
0x184	<a href="#">idm_access_status_ns</a>	RO	0x00000000	32-bit	This register indicates the access status for Non-secure transactions.
0x188	<a href="#">idm_access_readid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x18C	<a href="#">idm_access_writeid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x194	<a href="#">idm_reset_status_ns</a>	RO	0x00000000	32-bit	This register indicates the reset status of Non-secure transactions.
0x198	<a href="#">idm_reset_readid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x19C	<a href="#">idm_reset_writeid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x1A8	<a href="#">idm_interrupt_status_ns</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0x1AC	<a href="#">idm_interrupt_mask_ns</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.

### 16.15.1 HMNI node\_type register

This register identifies the node type as a node for HMNI registers.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x00

##### Type

RO

##### Reset value

See individual bit resets.

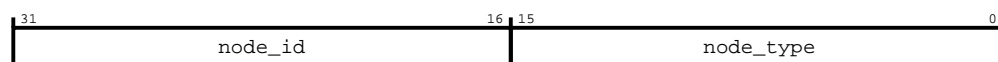
#### Constraints

None.

#### Bit descriptions

The following figure shows the node\_type register bit assignments.

**Figure 16-302: Bit assignment diagram for the node\_type register**



The following table shows the node\_type register bit descriptions.

**Table 16-317: node\_type bit descriptions**

Bits	Name	Description	Type	Reset
[31:16]	node_id	The HMNI ID that is assigned during network construction	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x0008, and it identifies the associated node type as a node for HMNI registers	RO	0x8

## 16.15.2 HMNI node\_info register

This register provides information about the HMNI node features and configuration.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x04

#### Type

RO

#### Reset value

See individual bit resets.

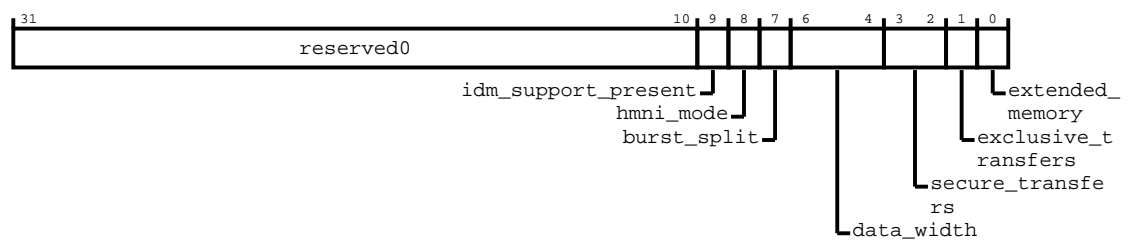
### Constraints

None.

### Bit descriptions

The following figure shows the node\_info register bit assignments.

**Figure 16-303: Bit assignment diagram for the node\_info register**



The following table shows the node\_info register bit descriptions.

**Table 16-318: node\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:10]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[9]	idm_support_present	IDM support:  <b>0</b> IDM support logic is not present.  <b>1</b> IDM support logic is present.	RO	Configuration dependent
[8]	hmni_mode	HMNI mode:  <b>0</b> HMNI is not in mirror mode.  <b>1</b> HMNI is in mirror mode.	RO	Configuration dependent
[7]	burst_split	Burst split:  <b>0</b> Burst split logic is not present.  <b>1</b> Burst split logic is present.	RO	Configuration dependent
[6:4]	data_width	Data width, HSIZE encoded:  <b>0b000</b> This value is reserved.  <b>0b001</b> This value is reserved.  <b>0b010</b> 4 bytes.  <b>0b011</b> 8 bytes.  <b>0b100</b> 16 bytes.  <b>0b101</b> 32 bytes.  <b>0b110</b> 64 bytes.  <b>0b111</b> 128 bytes.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[3:2]	secure_transfers	Specifies the behavior for Secure transfers:  <b>0b00</b> The software programs this register to set the security attribute of the downstream completer of this requester interface.  <b>0b10</b> The transfers are always set to Secure. The downstream AHB completer interface assets of the requester are Secure. Therefore, only Secure requests can travel downstream.  <b>0b11</b> The transfers are always Non-secure. The downstream AHB completer interface assets of the requester are Non-secure. Both Secure and Non-secure requests can travel downstream.	RO	Configuration dependent
[1]	exclusive_transfers	Indicates whether the HMNI node supports exclusive transfers.	RO	Configuration dependent
[0]	extended_memory	Indicates whether the HMNI node supports extended memory types.	RO	Configuration dependent

### 16.15.3 HMNI node\_features register

This register configures the node features.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x40

##### Type

RO

##### Reset value

0x00000000

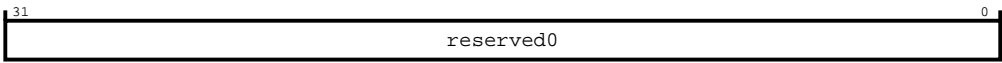
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the node\_features register bit assignments.

Figure 16-304: Bit assignment diagram for the node\_features register



The following table shows the node\_features register bit descriptions.

Table 16-319: node\_features bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0

16.15.4 HMNI secure\_access register

Contains register bits that are used for configuring the Secure access behavior.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x08

Type

RW

Reset value

0x00000000

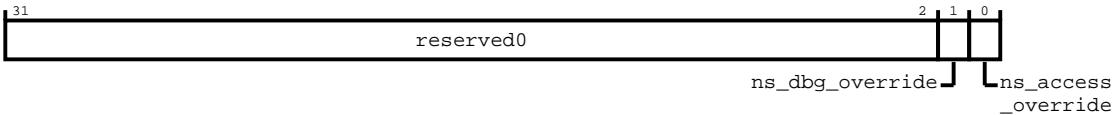
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the secure\_access register bit assignments.

Figure 16-305: Bit assignment diagram for the secure\_access register





The following table shows the secure\_access register bit descriptions.

**Table 16-320: secure\_access bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	ns_dbg_override	Enables/Disables non-secure access to AHB requester node PMU and interface registers	RW	0
[0]	ns_access_override	Enables/Disables non-secure access to AHB requester node registers	RW	0

### 16.15.5 HMNI node\_control register

Contains registers used for configuring the behavior of the HMNI node.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x44

##### Type

RW

##### Reset value

See individual bit resets.

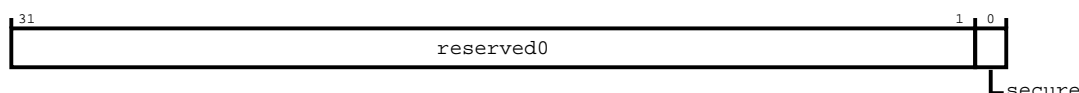
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the node\_control register bit assignments.

**Figure 16-306: Bit assignment diagram for the node\_control register**



The following table shows the node\_control register bit descriptions.

**Table 16-321: node\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	secure	<p>If the secure_transfers field of the HMNI NODE_INFO register is 0b00, it encodes a software programmable register. Therefore, the secure_ctrl field marks downstream completers as Secure or Non-secure based on its configuration setting:</p> <p><b>0</b></p> <p>Secure. Only Secure transactions can travel downstream.</p> <p><b>1</b></p> <p>Non-secure. Both Secure and Non-secure transactions can travel downstream.</p> <p>If the incoming request is Non-secure and the downstream completer is configured as Secure, then the transaction is not sent downstream. A Non-secure read transaction returns zero data. The data corresponding to a Non-secure write transaction is dropped, but a protocol-compliant write response is returned. The read or write response does not contain an error indication. If secure_transfers is 0b10 or 0b11, then the HNONSEC pin is unavailable. However, the interface security attribute is set at build time to either Always Secure or Always Non-secure. Therefore, the register bit becomes read-only. If secure_transfers is 0b11, the reset value is 1. If secure_transfers is 0b10, the reset value is 0b00.</p>	RW	Configuration dependent

### 16.15.6 HMNI pmusela register

This register is used to select the event values in the HMNI event crossbar.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x0C

##### Type

RW

##### Reset value

0x00000000

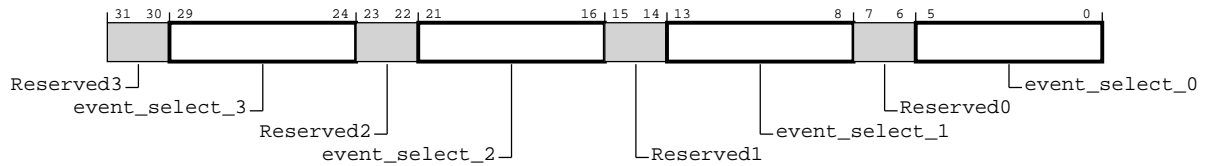
#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the pmusela register bit assignments.

**Figure 16-307: Bit assignment diagram for the pmusela register**



The following table shows the pmusela register bit descriptions.

**Table 16-322: pmusela bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

### 16.15.7 HMNI pmuselb register

This register is used to select the event values in the HMNI event crossbar.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x10

##### Type

RW

##### Reset value

0x00000000

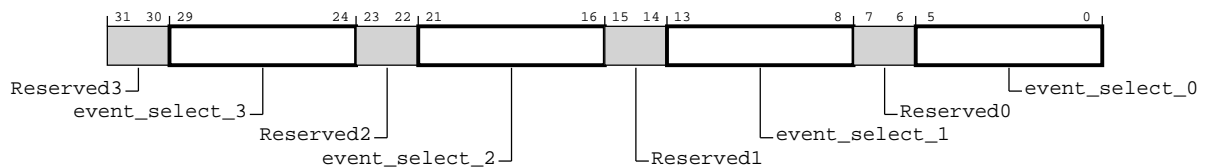
## Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the pmuselib register bit assignments.

**Figure 16-308: Bit assignment diagram for the pmuselib register**



The following table shows the pmuselib register bit descriptions.

**Table 16-323: pmuselib bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	PMU event 3 select	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	PMU event 2 select	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	PMU event 1 select	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	PMU event 0 select	RW	0b000000

## 16.15.8 HMNI interface\_id\_0\_3 register

Contains information about the HMNI interface IDs for interfaces 0-3.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x14

Type

RO

Reset value

See individual bit resets.

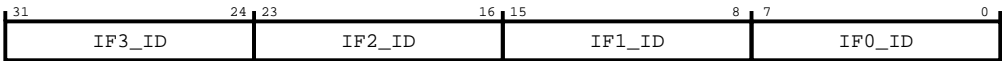
Constraints

None.

Bit descriptions

The following figure shows the interface\_id\_0\_3 register bit assignments.

Figure 16-309: Bit assignment diagram for the interface\_id\_0\_3 register



The following table shows the interface\_id\_0\_3 register bit descriptions.

Table 16-324: interface\_id\_0\_3 bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	IF3_ID	Reserved	RO	Configuration dependent
[23:16]	IF2_ID	Reserved	RO	Configuration dependent
[15:8]	IF1_ID	Reserved	RO	Configuration dependent
[7:0]	IF0_ID	HMNI interface ID 0	RO	Configuration dependent

16.15.9 HMNI num\_sub\_features register

Number of subfeatures.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x24

Type

RO

**Reset value**

See individual bit resets.

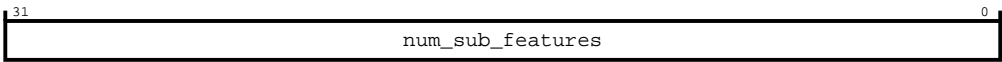
**Constraints**

None.

**Bit descriptions**

The following figure shows the num\_sub\_features register bit assignments.

**Figure 16-310: Bit assignment diagram for the num\_sub\_features register**



The following table shows the num\_sub\_features register bit descriptions.

**Table 16-325: num\_sub\_features bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	num_sub_features	Number of subfeatures	RO	Configuration dependent

**16.15.10 HMNI sub\_feature\_0\_type register**

Subfeature 0 type.

**Configurations**

This register is available in all configurations.

**Attributes**

Its characteristics are:

**Width**

32-bit

**Address offset**

0x28

**Type**

RO

**Reset value**

See individual bit resets.

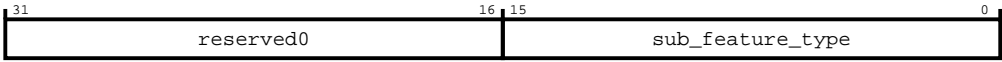
**Constraints**

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_type register bit assignments.

Figure 16-311: Bit assignment diagram for the sub\_feature\_0\_type register



The following table shows the sub\_feature\_0\_type register bit descriptions.

Table 16-326: sub\_feature\_0\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[15:0]	sub_feature_type	Subfeature 0 type	RO	Configuration dependent

16.15.11 HMNI sub\_feature\_0\_pointer register

Subfeature 0 pointer.

Configurations

The number of registers of this type that are present depends on the number of subfeatures in the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x2C

Type

RO

Reset value

See individual bit resets.

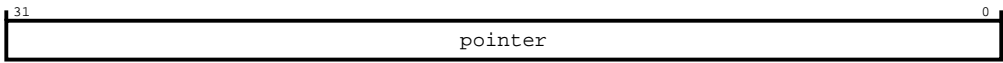
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_pointer register bit assignments.

Figure 16-312: Bit assignment diagram for the sub\_feature\_0\_pointer register



The following table shows the sub\_feature\_0\_pointer register bit descriptions.

Table 16-327: sub\_feature\_0\_pointer bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	pointer	Subfeature 0 pointer	RO	Configuration dependent

16.15.12 HMNI silicon\_debug register

This register monitors the status of requester interface channels.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x80

Type

RW

Reset value

0x00000000

Constraints

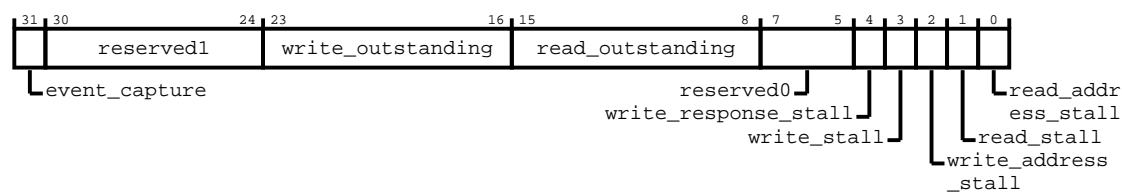
Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the silicon\_debug register bit assignments.



Figure 16-313: Bit assignment diagram for the silicon\_debug register



The following table shows the silicon\_debug register bit descriptions.

Table 16-328: silicon\_debug bit descriptions

Bits	Name	Description	Type	Reset
[31]	event_capture	Enable event capture.	RW	0
[30:24]	reserved1	Bits within this register segment are reserved for future product development	RO	0b0000000
[23:16]	write_outstanding	Indicates that the interface has outstanding write requests. Maximum value is 1.	RO	0x0
[15:8]	read_outstanding	Indicates that the interface has outstanding read requests. Maximum value is 1.	RO	0x0
[7:5]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[4]	write_response_stall	Indicates HMNI write response channel stall event	RO	0
[3]	write_stall	Prior write address phase, HREADY LOW.	RO	0
[2]	write_address_stall	HTRANS[1] HIGH, HWRITE HIGH, HREADY LOW.	RO	0
[1]	read_stall	Prior read address phase, HREADY LOW.	RO	0
[0]	read_address_stall	HTRANS[1] HIGH, HWRITE LOW, HREADY LOW.	RO	0

16.15.13 HMNI interrupt\_status register

This register indicates the interrupt status of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF0

Type

RW

Reset value

0x00000000

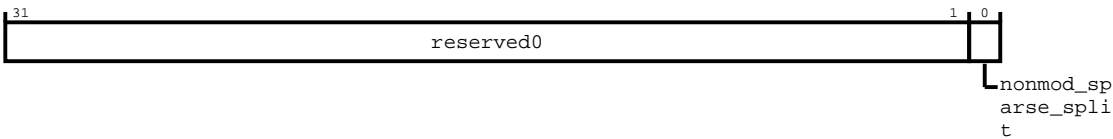
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the interrupt\_status register bit assignments.

Figure 16-314: Bit assignment diagram for the interrupt\_status register



The following table shows the interrupt\_status register bit descriptions.

Table 16-329: interrupt\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	nonmod_sparse_split	Indicates that a non-modifiable sparse transaction burst split has been observed by HMNI node.  Write 1 to clear.	RW	0

16.15.14 HMNI interrupt\_mask register

This register is the interrupt mask of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF4

Type

RW

Reset value

0x00000000

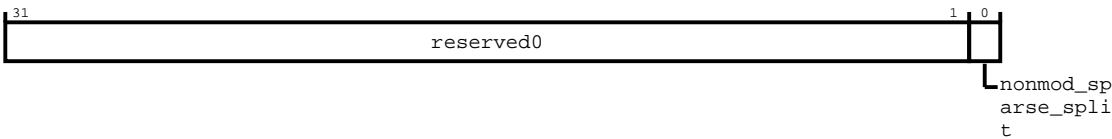
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the interrupt\_mask register bit assignments.

Figure 16-315: Bit assignment diagram for the interrupt\_mask register



The following table shows the interrupt\_mask register bit descriptions.

Table 16-330: interrupt\_mask bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	nonmod_sparse_split	Mask non-modifiable sparse transaction burst split interrupts observed by HMNI node.	RW	0

16.15.15 HMNI interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xF8

Type

RW

Reset value

0x00000000

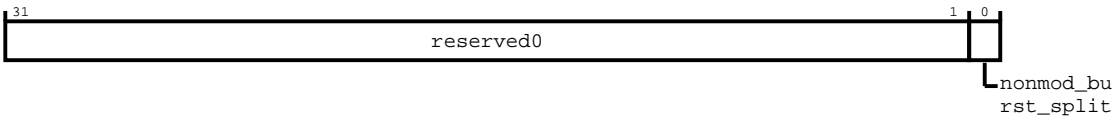
Constraints

None.

Bit descriptions

The following figure shows the interrupt\_status\_ns register bit assignments.

Figure 16-316: Bit assignment diagram for the interrupt\_status\_ns register



The following table shows the interrupt\_status\_ns register bit descriptions.

Table 16-331: interrupt\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	nonmod_burst_split	Indicates that a non-modifiable burst split has been observed by HMNI node.  Write 1 to clear.	RW	0

16.15.16 HMNI interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0xFC

Type

RW

Reset value

0x00000000

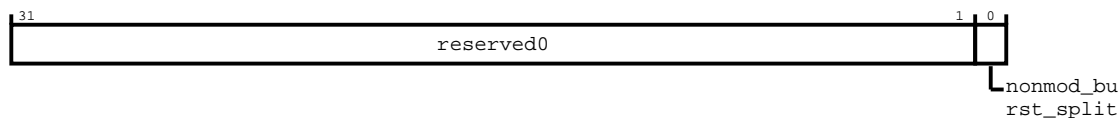
Constraints

None.

Bit descriptions

The following figure shows the interrupt\_mask\_ns register bit assignments.

**Figure 16-317: Bit assignment diagram for the interrupt\_mask\_ns register**



The following table shows the interrupt\_mask\_ns register bit descriptions.

**Table 16-332: interrupt\_mask\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	nonmod_burst_split	Mask non-modifiable burst split interrupts that are observed by HMNI node.	RW	0

### 16.15.17 HMNI idm\_device\_id register

This register indicates the statically configured device ID value and is implemented if IDM is enabled.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x100

#### Type

RO

#### Reset value

See individual bit resets.

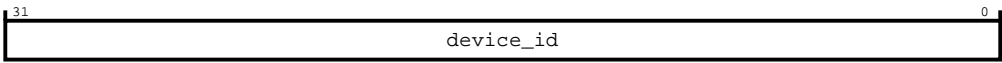
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_device\_id register bit assignments.

Figure 16-318: Bit assignment diagram for the `idm_device_id` register



The following table shows the `idm_device_id` register bit descriptions.

Table 16-333: `idm_device_id` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	device_id	Returns statically configured ID value	RO	Configuration dependent

16.15.18 HMNI `idm_config` register

This register enables transaction logging, error detection, timeout detection, access control, and reset control.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x104

Type

RW

Reset value

See individual bit resets.

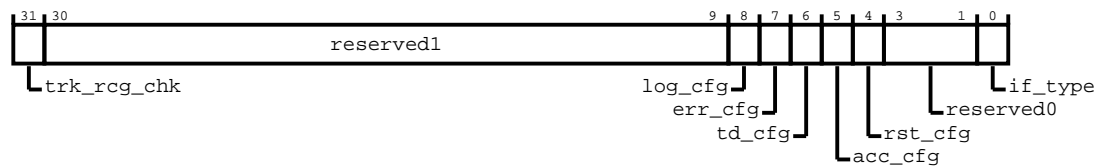
Constraints

None.

Bit descriptions

The following figure shows the `idm_config` register bit assignments.

**Figure 16-319: Bit assignment diagram for the idm\_config register**



The following table shows the idm\_config register bit descriptions.

**Table 16-334: idm\_config bit descriptions**

Bits	Name	Description	Type	Reset
[31]	trk_rcg_chk	Tracker Regional Clock Gating (RCG) chicken bit	RW	0
[30:9]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[8]	log_cfg	Transaction logging present	RO	1
[7]	err_cfg	Error detection present	RO	1
[6]	td_cfg	Timeout detection present	RO	1
[5]	acc_cfg	Access control present	RO	1
[4]	rst_cfg	Reset control present	RO	1
[3:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[0]	if_type	Interface type <b>0</b> Completer <b>1</b> Requester	RO	Configuration dependent

### 16.15.19 HMNI idm\_errctlr register

This register controls how errors are handled.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x108

##### Type

RW





16.15.20 HMNI idm\_errstatus register

This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured then OF is never set and SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x110

Type

RW

Reset value

0x00000000

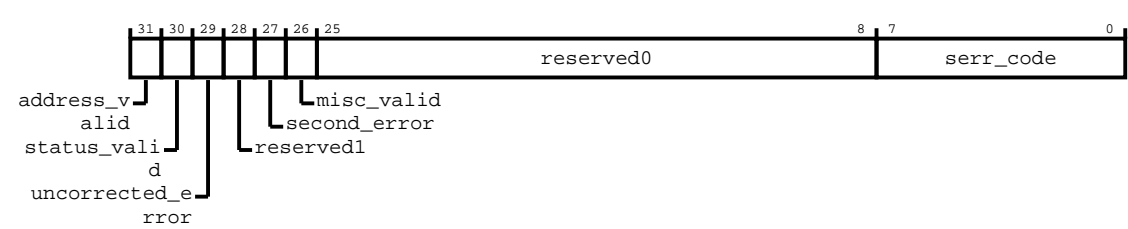
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_errstatus register bit assignments.

Figure 16-321: Bit assignment diagram for the idm\_errstatus register



The following table shows the idm\_errstatus register bit descriptions.

**Table 16-336: idm\_errstatus bit descriptions**

Bits	Name	Description	Type	Reset
[31]	address_valid	<p>Address valid. The values are:</p> <p><b>0</b></p> <p>ERRADDR is not valid.</p> <p><b>1</b></p> <p>ERRADDR contains an address that is associated with the highest priority error which this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1 and is not cleared to zero in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[30]	status_valid	<p>Status register is valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRSTATUS not valid</p> <p><b>1</b></p> <p>IDM_ERRSTATUS valid. At least one error has been recorded.</p> <p>This bit ignores writes if any of the following fields is set to 1 and is not being cleared to zero in the same write:</p> <ul style="list-style-type: none"> <li>IDM_ERRSTATUS.UE</li> <li>IDM_ERRSTATUS.AV</li> <li>IDM_ERRSTATUS.OF * IDM_ERRSTATUS.MV</li> </ul> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred</p> <p>This bit ignores writes if IDM_ERRSTATUS.OF is set to 1 and is not being cleared to zero in the same write. This bit is not valid and reads <b>UNKNOWN</b> if IDM_ERRSTATUS.V is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0

Bits	Name	Description	Type	Reset
[27]	second_error	Returns whether a second error has been received while handling a first error. The values are:  <b>1</b> Second error received  <b>0</b> No other error received  This bit is read, or write 1 to clear  Write 1 to clear.	RW	0
[26]	misc_valid	Miscellaneous registers valid. The values are:  <b>0</b> IDM_ERRMISC0 and IDM_ERRMISC1 not valid  <b>1</b> The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISC0 and IDM_ERRMISC1 registers contains additional information for an error that this record records.  This bit ignores writes if IDM_ERRSTATUS.UE is set to 1, and is not being cleared to 0 in the same write. This bit is a read, or write 1 to clear.  Write 1 to clear.	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	Primary error code. Indicates the type of error. The values are:  <b>00</b> No error  <b>13</b> Illegal address - decode error  <b>18</b> Error response from completer  <b>20</b> Internal timeout	RO	0x0

### 16.15.21 HMNI idm\_erraddr\_lsb register

This register is the error log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

Address offset

0x114

Type

RO

Reset value

0x00000000

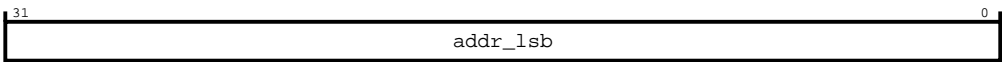
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_lsb` register bit assignments.

Figure 16-322: Bit assignment diagram for the `idm_erraddr_lsb` register



The following table shows the `idm_erraddr_lsb` register bit descriptions.

Table 16-337: `idm_erraddr_lsb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.15.22 HMNI `idm_erraddr_msb` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x118

Type

RO

Reset value

0x00000000

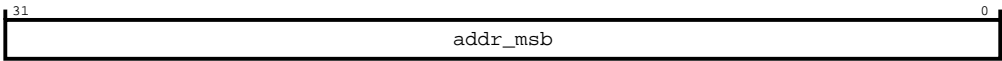
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_msb` register bit assignments.

Figure 16-323: Bit assignment diagram for the `idm_erraddr_msb` register



The following table shows the `idm_erraddr_msb` register bit descriptions.

Table 16-338: `idm_erraddr_msb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.15.23 HMNI `idm_errmisc0` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x128

Type

RO

Reset value

0x00000000

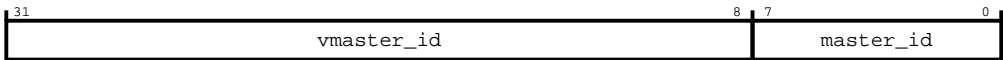
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_errmisc0` register bit assignments.

Figure 16-324: Bit assignment diagram for the `idm_errmisc0` register



The following table shows the `idm_errmisc0` register bit descriptions.

Table 16-339: `idm_errmisc0` bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AxiD into ASNI of the transaction causing an error. The assumption here is there is no manipulation of incoming AXI AxiD in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.15.24 HMNI `idm_errmisc1` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x12C

Type

RO

Reset value

0x00000000

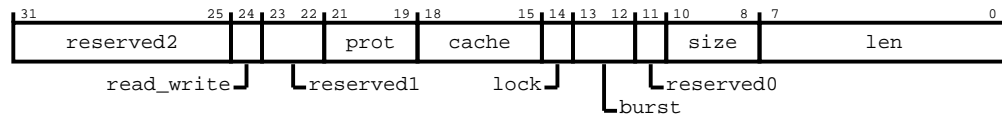
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_errmisc1` register bit assignments.

**Figure 16-325: Bit assignment diagram for the idm\_errmisc1 register**



The following table shows the idm\_errmisc1 register bit descriptions.

**Table 16-340: idm\_errmisc1 bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	The AXI read or write information of a transaction causing an error  1 Write 0 Read	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	The AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	The AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	The AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	The AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	The AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	The AXI len information of a transaction causing an error.	RO	0x0

### 16.15.25 HMNI idm\_access\_control register

This register controls the state, gated or ungated, of a device.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x130

##### Type

RW

Reset value

0x00000000

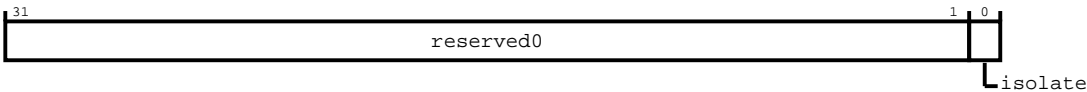
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the idm\_access\_control register bit assignments.

Figure 16-326: Bit assignment diagram for the idm\_access\_control register



The following table shows the idm\_access\_control register bit descriptions.

Table 16-341: idm\_access\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	isolate	Perform gating off a device. Reading 1 indicates that the completer device is gated or isolated. Reading 0 indicates that the completer device is ungated or de-isolated. Write 1 to enter gated state. Write 0 to exit gated state. There is some delay to updating this field with the intended write value. Exit from gated state is only successful if there are no outstanding transactions and all error status register bits are cleared. Entry into gated state is only successful if there are no outstanding transactions. While in pending isolation entry state or in active isolation state, a write of 1 to this bit causes reentry to isolation state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 0 is ignored. While in pending isolation exit state, a write of 0 to this bit causes a re-exit to the exit state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS, and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 1 is ignored.	RW	0

16.15.26 HMNI idm\_access\_status register

This register indicates the access status for Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit



## Address offset

0x134

## Type

RO

## Reset value

0x00000002

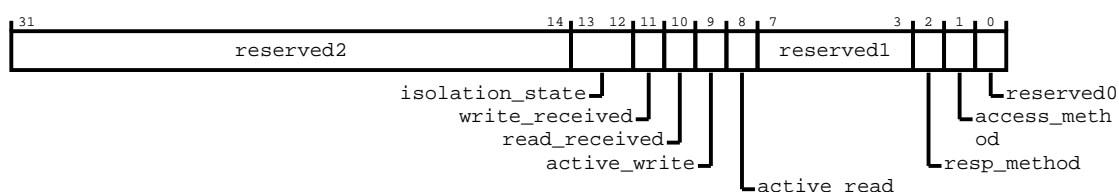
## Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the `idm_access_status` register bit assignments.

**Figure 16-327: Bit assignment diagram for the `idm_access_status` register**



The following table shows the `idm_access_status` register bit descriptions.

### Table 16-342: idm\_access\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved2	Bits within this register segment are reserved for future product development	RO	0x0
[13:12]	isolation_state	<p>Isolation status:</p> <p><b>00</b></p> <p>Isolation exit or entry is successful or not in gated or isolation state</p> <p><b>01</b></p> <p>Isolation exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus</p> <p><b>10</b></p> <p>Isolation entry is unsuccessful or pending because of outstanding transactions</p> <p><b>11</b></p> <p>Reserved</p>	RO	0b00
[11]	write_received	<p>A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on:</p> <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0

Bits	Name	Description	Type	Reset
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:3]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00000
[2]	resp_method	Indicates device generates errors in gated access	RO	0
[1]	access_method	Wait for all outstanding to complete, then block input	RO	1
[0]	reserved0	Bits within this register segment are reserved for future product development	RO	0

### 16.15.27 HMNI idm\_access\_readid register

This register is the access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x138

##### Type

RO

##### Reset value

0x00000000

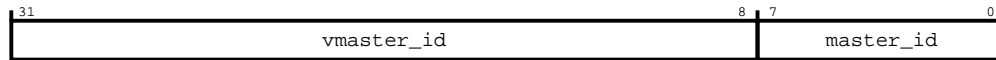
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_access\_readid register bit assignments.

**Figure 16-328: Bit assignment diagram for the `idm_access_readid` register**



The following table shows the `idm_access_readid` register bit descriptions.

**Table 16-343: `idm_access_readid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming signal into the endpoint of the first transaction to arrive after isolation when the <code>active_read</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the ARID of the transaction on ASNI and contains the HMASTER on HSNi. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNi of the first transaction to arrive after isolation when the <code>active_read</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

## 16.15.28 HMNI `idm_access_writeid` register

This register is the access log of Secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x13C

#### Type

RO

#### Reset value

0x00000000

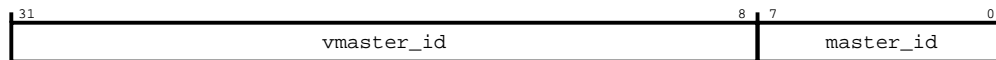
### Constraints

Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the `idm_access_writeid` register bit assignments.

**Figure 16-329: Bit assignment diagram for the `idm_access_writeid` register**



The following table shows the `idm_access_writeid` register bit descriptions.

**Table 16-344: `idm_access_writeid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AWID signal into the endpoint of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

### 16.15.29 HMNI `idm_reset_control` register

This register controls the reset of a device that is attached to the interconnect.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x140

##### Type

RW

##### Reset value

0x00000002

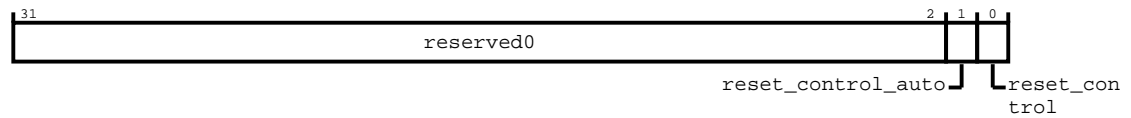
#### Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the `idm_reset_control` register bit assignments.

**Figure 16-330: Bit assignment diagram for the idm\_reset\_control register**



The following table shows the idm\_reset\_control register bit descriptions.

**Table 16-345: idm\_reset\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[1]	reset_control_auto	<p>Configures the device for auto or internal reset mode. For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. There are several constraints on this field:</p> <ul style="list-style-type: none"> <li>You can only change this field during initialization or when the interface is fully quiesced. * Arm does not support changing this field while the interface is active. If you change this field during runtime, behavior is <b>UNPREDICTABLE</b>.</li> </ul> <p>Reads have the following effect:</p> <p><b>1</b></p> <p>A read of 1 indicates that the device is in auto or internal reset mode.</p> <p><b>0</b></p> <p>A read of 0 indicates that the device is not in auto or internal reset mode.</p> <p>Writes have the following effect:</p> <p><b>1</b></p> <p>A write of 1 configures the device for auto or internal reset mode.</p> <p><b>0</b></p> <p>A write of 0 disables auto or internal reset mode.</p> <p>For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. Bit[1] of the IDM_RESET_CONTROL register is 1 out of reset. This bit enables internal recovery mode out of reset. When not in auto reset mode and a timeout is detected, a write of 1 to the IDM_RESET_CONTROL.reset field initiates internal recovery mode. Changing this bit while the interface is not in idle mode results in <b>UNPREDICTABLE</b> behavior.</p>	RO	1

Bits	Name	Description	Type	Reset
[0]	reset_control	<p>Performs soft reset of attached device. If the auto bit is set to 1 the network interface gates the external interface, however the soft reset pin is not activated. If the auto bit is 0, the interfaces are not gated until there is a write to bit[0]. In this case, the soft reset pin is activated. Writes have the following effect:</p> <p><b>1</b></p> <p>Request the attached device to enter reset. If the write occurs before soft reset exit has occurred, the write is ignored.</p> <p><b>0</b></p> <p>Request the attached device to exit reset. If the write occurs before soft reset entry has occurred, the write is ignored.</p> <p>Software polls this register to determine if soft reset entry or exit has occurred, using the following values:</p> <p><b>1</b></p> <p>Indicates that the device is in reset.</p> <p><b>0</b></p> <p>Indicates that the device is not in reset.</p> <p>This register value updates to reflect a request for reset entry or reset exit, but the update can only occur after required internal conditions are met. Until these conditions are met, a read to this register returns the old value. For example, outstanding transactions currently being handled must complete before this register value updates. To ensure reset propagation within the device, it is the responsibility of the software to permit enough cycles after soft reset assertion is reflected in the IDM_RESET_CONTROL register before exiting soft reset by triggering a write of 0. If this responsibility is not met, the behavior is <b>UNDEFINED</b> or <b>UNPREDICTABLE</b>. When this register value is 1, the external soft reset pin that connects to the attached AXI requester or completer device is asserted, using the correct polarity of the reset pin. When this register value is 0, the external soft reset pin that connects to the attached AXI requester or completer device is deasserted, using the correct polarity of the reset pin. When in pending soft reset entry state or in active soft reset state, a write of 1 to this bit causes reentry to soft reset state. This write causes the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers to be cleared. A write of 0 is ignored. While in pending soft reset exit state, a write of 0 to this bit causes re-exit to exit state. A write of 0 also clears the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers. A write of 1 is ignored.</p>	RW	0

### 16.15.30 HMNI idm\_reset\_status register

This register indicates mostly the reset status of Secure transactions. However, the rst\_exit\_state field indicates reset exit state of secure or non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

## Address offset

0x144

## Type

RO

## Reset value

0x00000000

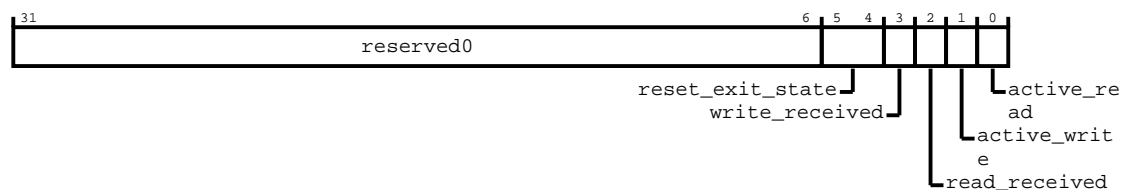
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the idm\_reset\_status register bit assignments.

**Figure 16-331: Bit assignment diagram for the idm\_reset\_status register**



The following table shows the idm\_reset\_status register bit descriptions.

**Table 16-346: idm\_reset\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[5:4]	reset_exit_state	Reset exit state  <b>00</b> Reset exit or entry is successful or not in reset state  <b>01</b> Reset exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  <b>10</b> Reset exit is unsuccessful or pending because of outstanding transactions  <b>11</b> Reset exit is unsuccessful or pending because of both uncleared error status bits and outstanding transactions	RO	0b00
[3]	write_received	A 1 indicates that an active Secure write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0

### 16.15.31 HMNI idm\_reset\_readid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x148

##### Type

RO

##### Reset value

0x00000000

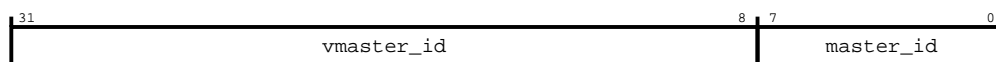
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_readid register bit assignments.

**Figure 16-332: Bit assignment diagram for the idm\_reset\_readid register**



The following table shows the idm\_reset\_readid register bit descriptions.



**Table 16-347: idm\_reset\_readid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.15.32 HMNI idm\_reset\_writeid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x14C

##### Type

RO

##### Reset value

0x00000000

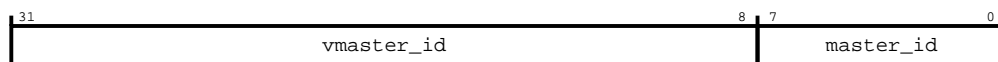
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid register bit assignments.

**Figure 16-333: Bit assignment diagram for the idm\_reset\_writeid register**



The following table shows the idm\_reset\_writeid register bit descriptions.

### Table 16-348: idm\_reset\_writeid bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNi. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNi of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.15.33 HMNI idm\_timeout\_control register

This register is present when timeout detection is configured.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x150

## Type

RW

## Reset value

```
0x00000000
```

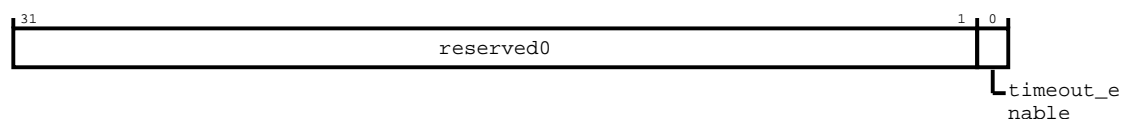
## Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the `idm_timeout_control` register bit assignments.

**Figure 16-334: Bit assignment diagram for the `idm_timeout_control` register**



The following table shows the `idm_timeout_control` register bit descriptions.

**Table 16-349: idm\_timeout\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[0]	timeout_enable	<p>Timeout detection enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when a timeout is detected. The timeout is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>A timeout interrupt event is generated, unless it is masked.</p>	RW	0

### 16.15.34 HMNI idm\_timeout\_value register

This register controls the duration that is used to determine if a transaction has timed out.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x154

##### Type

RW

##### Reset value

0x00000004

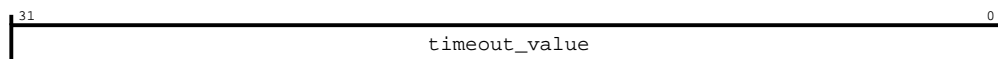
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the idm\_timeout\_value register bit assignments.

**Figure 16-335: Bit assignment diagram for the idm\_timeout\_value register**



The following table shows the `idm_timeout_value` register bit descriptions.

### Table 16-350: idm\_timeout\_value bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	timeout_value	Controls the duration that is used to determine if a transaction has timed out. The actual duration is $2^{\text{timeout\_exponent}}$ cycles. The minimum value is 4. Values of 0, 1, 2, or 3 are treated as 4. The maximum value is 30. Values greater than 30 are treated as 30.	RW	0x4

### 16.15.35 HMNI idm\_interrupt\_status register

This register indicates the interrupt status of Secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x158

## Type

RW

## Reset value

0x00000000

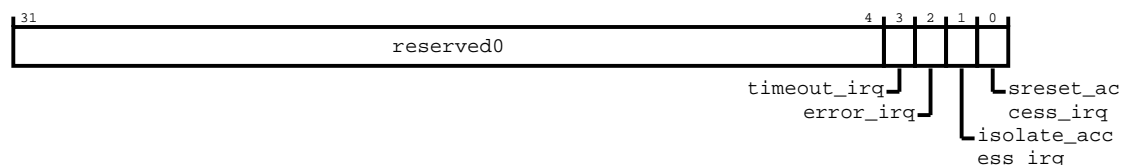
## Constraints

Only accessible using Secure transactions.

## Bit descriptions

The following figure shows the `idm_interrupt_status` register bit assignments.

**Figure 16-336: Bit assignment diagram for the `idm_interrupt_status` register**



The following table shows the `idm_interrupt_status` register bit descriptions.

**Table 16-351: idm\_interrupt\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.15.36 HMNI idm\_interrupt\_mask register

This register is the interrupt mask of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x15C

##### Type

RW

##### Reset value

0x00000000

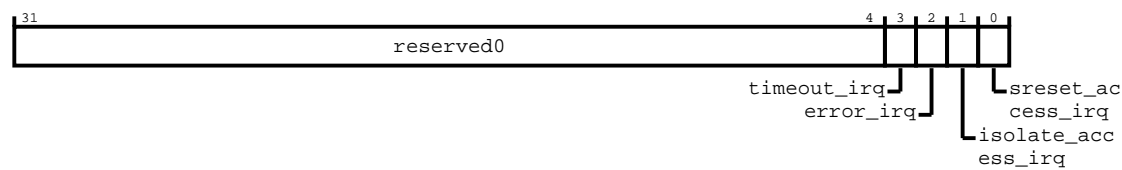
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask register bit assignments.

Figure 16-337: Bit assignment diagram for the `idm_interrupt_mask` register



The following table shows the `idm_interrupt_mask` register bit descriptions.

Table 16-352: `idm_interrupt_mask` bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

16.15.37 HMNI `idm_errstatus_ns` register

This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x160

Type

RW

Reset value

0x00000000

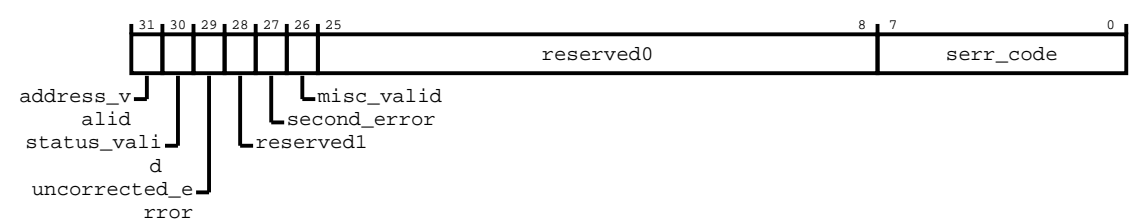
Constraints

None.

Bit descriptions

The following figure shows the `idm_errstatus_ns` register bit assignments.

Figure 16-338: Bit assignment diagram for the `idm_errstatus_ns` register



The following table shows the `idm_errstatus_ns` register bit descriptions.

Table 16-353: `idm_errstatus_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31]	<code>address_valid</code>	Address valid. The values are: <b>0</b> ERRADDR is not valid. <b>1</b> ERRADDR contains an address that is associated with the highest priority error that this record captures.  This bit ignores writes if the <code>ue</code> field of the <code>IDM_ERRSTATUS_NS</code> register is set to 1 and is not cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0
[30]	<code>status_valid</code>	Status register valid. The values are: <b>0</b> IDM_ERRSTATUS_NS is not valid. <b>1</b> IDM_ERRSTATUS_NS is valid. At least one error has been recorded.  This bit ignores writes if the <code>ue</code> field of the <code>IDM_ERRSTATUS_NS</code> register is set to 1 and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred.</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred.</p> <p>This bit ignores writes if the oe field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is not valid and reads <b>UNKNOWN</b> if the v field of the IDM_ERRSTATUS_NS register is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Bits within this register segment are reserved for future product development	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISCO_NS and IDM_ERRMISC1_NS are not valid.</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISCO_NS and IDM_ERRMISC1_NS registers contains additional information for an error that this record captures.</p> <p>This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1, and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[7:0]	serr_code	<p>Primary error code, indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0



16.15.38 HMNI idm\_erraddr\_lsb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x164

Type

RO

Reset value

0x00000000

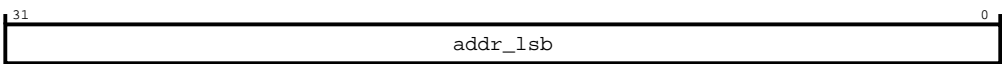
Constraints

None.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb\_ns register bit assignments.

Figure 16-339: Bit assignment diagram for the idm\_erraddr\_lsb\_ns register



The following table shows the idm\_erraddr\_lsb\_ns register bit descriptions.

Table 16-354: idm\_erraddr\_lsb\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.15.39 HMNI idm\_erraddr\_msb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x168

Type

RO

Reset value

0x00000000

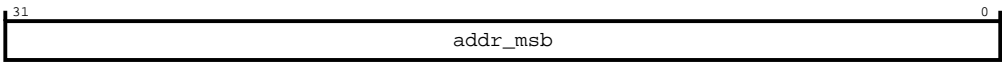
Constraints

None.

Bit descriptions

The following figure shows the `idm_erraddr_msb_ns` register bit assignments.

Figure 16-340: Bit assignment diagram for the `idm_erraddr_msb_ns` register



The following table shows the `idm_erraddr_msb_ns` register bit descriptions.

Table 16-355: `idm_erraddr_msb_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.15.40 HMNI `idm_errmisc0_ns` register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x178

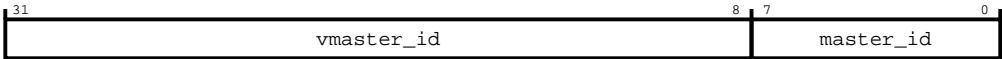
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the idm\_errmisc0\_ns register bit assignments.

**Figure 16-341: Bit assignment diagram for the idm\_errmisc0\_ns register**



The following table shows the idm\_errmisc0\_ns register bit descriptions.

**Table 16-356: idm\_errmisc0\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming AXI AxID into ASNI of the transaction causing an error. The assumption is no manipulation of incoming AXI AxID in ASNI.	RO	0x0
[7:0]	master_id	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.15.41 HMNI idm\_errmisc1\_ns register

This register is the error log of Non-secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x17C

**Type**  
RO

**Reset value**  
0x00000000

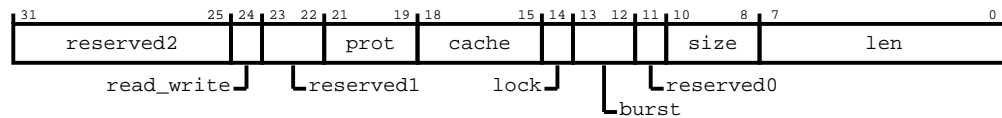
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_errmisc1_ns` register bit assignments.

**Figure 16-342: Bit assignment diagram for the `idm_errmisc1_ns` register**



The following table shows the `idm_errmisc1_ns` register bit descriptions.

### Table 16-357: idm\_errmisc1\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Bits within this register segment are reserved for future product development	RO	0b0000000
[24]	read_write	Returns the AXI read or write information of a transaction causing an error:  <div> <div>1</div> <div>Write</div> </div> <div> <div>0</div> <div>Read</div> </div>	RO	0
[23:22]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[21:19]	prot	Returns the AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	Returns the AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	Returns the AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	Returns the AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Bits within this register segment are reserved for future product development	RO	0
[10:8]	size	Returns the AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	Returns the AXI len information of a transaction causing an error.	RO	0x0

#### 16.15.42 HMNI idm\_access\_status\_ns register

This register indicates the access status for Non-secure transactions.

## Configurations

This register is available in all configurations.

## Attributes

Its characteristics are:

## Width

32-bit

## Address offset

0x184

## Type

RO

## Reset value

0x00000000

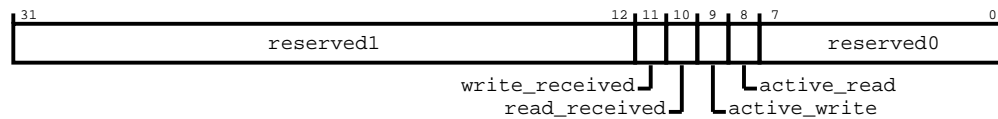
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_access_status_ns` register bit assignments.

**Figure 16-343: Bit assignment diagram for the `idm_access_status_ns` register**



The following table shows the `idm_access_status_ns` register bit descriptions.

**Table 16-358: `idm_access_status_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved1	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:0]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0

### 16.15.43 HMNI idm\_access\_readid\_ns register

This register is the access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x188

##### Type

RO

##### Reset value

0x00000000

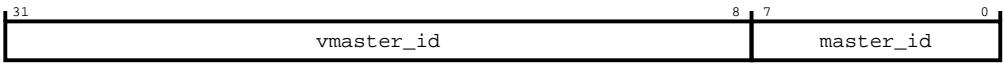
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_access\_readid\_ns register bit assignments.

**Figure 16-344: Bit assignment diagram for the idm\_access\_readid\_ns register**



The following table shows the idm\_access\_readid\_ns register bit descriptions.

**Table 16-359: idm\_access\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

16.15.44 HMNI idm\_access\_writeid\_ns register

This register is the access log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x18C

Type

RO

Reset value

0x00000000

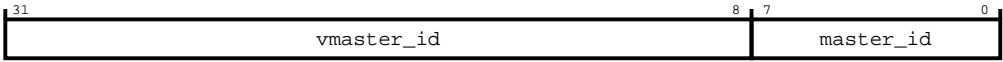
Constraints

None.

Bit descriptions

The following figure shows the idm\_access\_writeid\_ns register bit assignments.

Figure 16-345: Bit assignment diagram for the idm\_access\_writeid\_ns register



The following table shows the idm\_access\_writeid\_ns register bit descriptions.

Table 16-360: idm\_access\_writeid\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the IDM_ACCESS_STATUS_NS register field active_write is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

16.15.45 HMNI idm\_reset\_status\_ns register

This register indicates the reset status of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x194

Type

RO

Reset value

0x00000000

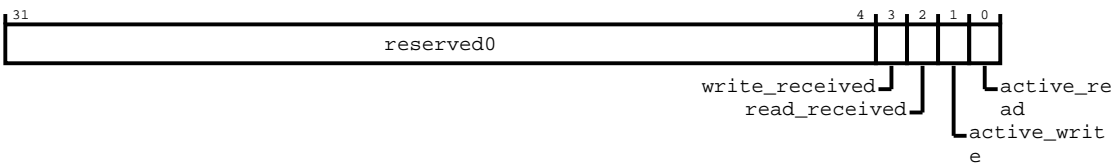
Constraints

None.

Bit descriptions

The following figure shows the idm\_reset\_status\_ns register bit assignments.

Figure 16-346: Bit assignment diagram for the idm\_reset\_status\_ns register



The following table shows the idm\_reset\_status\_ns register bit descriptions.

Table 16-361: idm\_reset\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[3]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"><li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li><li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li></ul>	RO	0



Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to 0 on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates that there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates that there is at least one read transaction currently in progress.	RO	0

### 16.15.46 HMNI idm\_reset\_readid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x198

##### Type

RO

##### Reset value

0x00000000

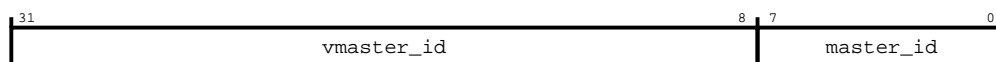
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_readid\_ns register bit assignments.

**Figure 16-347: Bit assignment diagram for the idm\_reset\_readid\_ns register**



The following table shows the idm\_reset\_readid\_ns register bit descriptions.

**Table 16-362: idm\_reset\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

### 16.15.47 HMNI idm\_reset\_writeid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x19C

##### Type

RO

##### Reset value

0x00000000

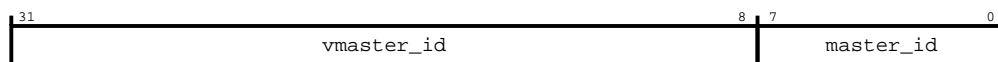
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid\_ns register bit assignments.

**Figure 16-348: Bit assignment diagram for the idm\_reset\_writeid\_ns register**



The following table shows the idm\_reset\_writeid\_ns register bit descriptions.

**Table 16-363: idm\_reset\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when active_write field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

## 16.15.48 HMNI idm\_interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1A8

#### Type

RW

#### Reset value

0x00000000

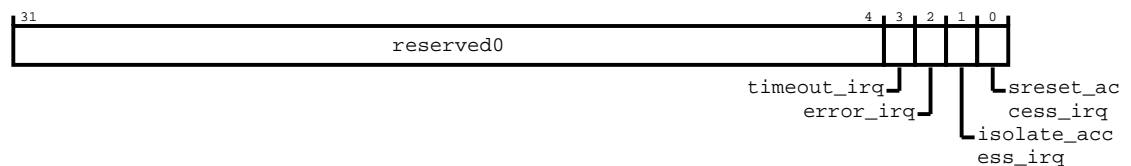
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_status\_ns register bit assignments.

**Figure 16-349: Bit assignment diagram for the idm\_interrupt\_status\_ns register**



The following table shows the idm\_interrupt\_status\_ns register bit descriptions.

**Table 16-364: idm\_interrupt\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

## 16.15.49 HMNI idm\_interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1AC

#### Type

RW

#### Reset value

0x00000000

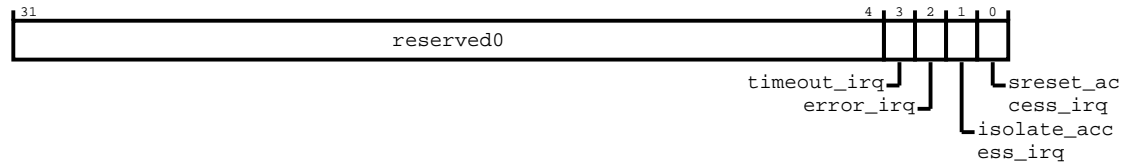
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_mask\_ns register bit assignments.

**Figure 16-350: Bit assignment diagram for the `idm_interrupt_mask_ns` register**



The following table shows the `idm_interrupt_mask_ns` register bit descriptions.

**Table 16-365: `idm_interrupt_mask_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

## 16.16 PMNI register summary

This section describes the PMNI registers. It contains a summary of the registers, in order of address offset, and a description of the bitfields for each register.

### Summary table

**Table 16-366: PMNI register summary**

Offset	Name	Type	Reset	Width	Description
0x00	<a href="#">node_type</a>	RO	See individual bit resets.	32-bit	This register identifies the node type as a node for PMNI registers.
0x04	<a href="#">node_info</a>	RO	See individual bit resets.	32-bit	This register provides node information for PMNI, such as data width.
0x08	<a href="#">secure_access</a>	RW	0x00000000	32-bit	This register contains information to configure the secure access behavior of the PMNI node.
0x0C	<a href="#">pmusela</a>	RW	0x00000000	32-bit	This register configures the PMU A crossbar to select particular events to monitor.
0x10	<a href="#">pmuselb</a>	RW	0x00000000	32-bit	This register configures the PMU B crossbar to select particular events to monitor.
0x14	<a href="#">interface_id_0_3</a>	RO	See individual bit resets.	32-bit	To configure APB interface IDs 0-3, use offset 0x014 in the <code>PMNI_INTERFACEID</code> register.
0x18	<a href="#">interface_id_4_7</a>	RO	See individual bit resets.	32-bit	To configure APB interface IDs 4-7, use offset 0x018 in the <code>PMNI_INTERFACEID</code> register.
0x1C	<a href="#">interface_id_8_11</a>	RO	See individual bit resets.	32-bit	To configure APB interface IDs 8-11, use offset 0x01C in the <code>PMNI_INTERFACEID</code> register.
0x20	<a href="#">interface_id_12_15</a>	RO	See individual bit resets.	32-bit	To configure APB interface IDs 12-15, use offset 0x020 in the <code>PMNI_INTERFACEID</code> register.

Offset	Name	Type	Reset	Width	Description
0x24	num_sub_features	RO	See individual bit resets.	32-bit	The number of subfeatures.
0x28	sub_feature_0_type	RO	See individual bit resets.	32-bit	Subfeature 0 type.
0x2C	sub_feature_0_pointer	RO	See individual bit resets.	32-bit	Subfeature 0 pointer.
0x30	secure_info	RO	See individual bit resets.	32-bit	Shows the security attribute for each of the APB interfaces downstream of the PMNI.
0x40	node_features	RO	See individual bit resets.	32-bit	This register configures the node features. You can configure up to 16 APB interfaces for a PMNI. Use two bits to identify the APB protocol for a specific interface.
0x44	node_control	RW	See individual bit resets.	32-bit	This register indicates the security status, Secure or Non-secure, of APB interfaces that are attached to a PMNI.
0x80	silicon_debug	RW	0x00000000	32-bit	This register monitors the status of requester interface channels.
0x100	idm_device_id	RO	See individual bit resets.	32-bit	This register indicates the statically configured device ID value and is implemented if IDM is enabled.
0x104	idm_config	RW	See individual bit resets.	32-bit	This register enables transaction logging, error detection, timeout detection, access control, and reset control.
0x108	idm_errctrl	RW	0x00000000	32-bit	This register controls how errors are handled.
0x110	idm_errstatus	RW	0x00000000	32-bit	This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured, then OF is never set and SERR only reads as no error or timeout error.
0x114	idm_erraddr_lsb	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x118	idm_erraddr_msb	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x128	idm_errmisc0	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x12C	idm_errmisc1	RO	0x00000000	32-bit	This register is the error log of Secure transactions.
0x130	idm_access_control	RW	0x00000000	32-bit	This register controls the state, gated or ungated, of a device.
0x134	idm_access_status	RO	0x00000002	32-bit	This register indicates the access status for Secure transactions.
0x138	idm_access_readid	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x13C	idm_access_writeid	RO	0x00000000	32-bit	This register is the access log of Secure transactions.
0x140	idm_reset_control	RW	0x00000002	32-bit	This register controls the reset of a device that is attached to the interconnect.
0x144	idm_reset_status	RO	0x00000000	32-bit	This register indicates mostly the reset status of Secure transactions. However, the rst_exit_state field indicates reset exit state of Secure or Non-secure transactions.
0x148	idm_reset_readid	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x14C	idm_reset_writeid	RO	0x00000000	32-bit	This register is the reset access log of Secure transactions.
0x150	idm_timeout_control	RW	0x00000000	32-bit	This register is present when timeout detection is configured.
0x154	idm_timeout_value	RW	0x00000004	32-bit	This register controls the duration that is used to determine if a transaction has timed out.
0x158	idm_interrupt_status	RW	0x00000000	32-bit	This register indicates the interrupt status of Secure transactions.
0x15C	idm_interrupt_mask	RW	0x00000000	32-bit	This register is the interrupt mask of Secure transactions.
0x160	idm_errstatus_ns	RW	0x00000000	32-bit	This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Offset	Name	Type	Reset	Width	Description
0x164	<a href="#">idm_erraddr_lsb_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x168	<a href="#">idm_erraddr_msb_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x178	<a href="#">idm_errmisc0_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x17C	<a href="#">idm_errmisc1_ns</a>	RO	0x00000000	32-bit	This register is the error log of Non-secure transactions.
0x184	<a href="#">idm_access_status_ns</a>	RO	0x00000000	32-bit	This register indicates the access status for Non-secure transactions.
0x188	<a href="#">idm_access_readid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x18C	<a href="#">idm_access_writeid_ns</a>	RO	0x00000000	32-bit	This register is the access log of Non-secure transactions.
0x194	<a href="#">idm_reset_status_ns</a>	RO	0x00000000	32-bit	This register indicates the reset status of Non-secure transactions.
0x198	<a href="#">idm_reset_readid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x19C	<a href="#">idm_reset_writeid_ns</a>	RO	0x00000000	32-bit	This register is the reset access log of Non-secure transactions.
0x1A8	<a href="#">idm_interrupt_status_ns</a>	RW	0x00000000	32-bit	This register indicates the interrupt status of Non-secure transactions.
0x1AC	<a href="#">idm_interrupt_mask_ns</a>	RW	0x00000000	32-bit	This register is the interrupt mask of Non-secure transactions.

### 16.16.1 PMNI node\_type register

This register identifies the node type as a node for PMNI registers.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x00

##### Type

RO

##### Reset value

See individual bit resets.

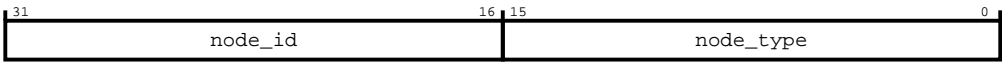
#### Constraints

None.

#### Bit descriptions

The following figure shows the node\_type register bit assignments.

Figure 16-351: Bit assignment diagram for the node\_type register



The following table shows the node\_type register bit descriptions.

Table 16-367: node\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	node_id	The PMNI ID that is assigned during network construction.	RO	Configuration dependent
[15:0]	node_type	The value of this field is 0x0009, and it identifies the associated node type as a node for PMNI registers.	RO	0x9

16.16.2 PMNI node\_info register

This register provides node information for PMNI, such as data width.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x04

Type

RO

Reset value

See individual bit resets.

Constraints

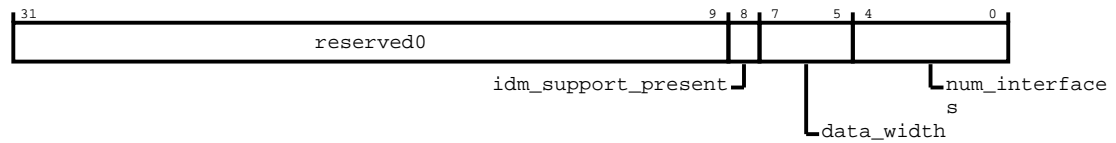
None.

Bit descriptions

The following figure shows the node\_info register bit assignments.



**Figure 16-352: Bit assignment diagram for the node\_info register**



The following table shows the node\_info register bit descriptions.

**Table 16-368: node\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:9]	reserved0	Bits within this register segment are reserved for future product development.	RO	0x0
[8]	idm_support_present	IDM support: <b>0</b> IDM support logic is not present. <b>1</b> IDM support logic is present.	RO	Configuration dependent
[7:5]	data_width	Data width, HSIZE encoded: <b>0b000</b> This value is reserved. <b>0b001</b> This value is reserved. <b>0b010</b> 4 bytes. <b>0b011</b> This value is reserved. <b>0b100</b> This value is reserved. <b>0b101</b> This value is reserved. <b>0b110</b> This value is reserved. <b>0b111</b> This value is reserved.	RO	Configuration dependent
[4:0]	num_interfaces	The number of enabled APB interfaces at a specific PMNI. Permitted values are between 1 and 16.	RO	Configuration dependent

### 16.16.3 PMNI secure\_access register

This register contains information to configure the secure access behavior of the PMNI node.

#### Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x08

Type

RW

Reset value

0x00000000

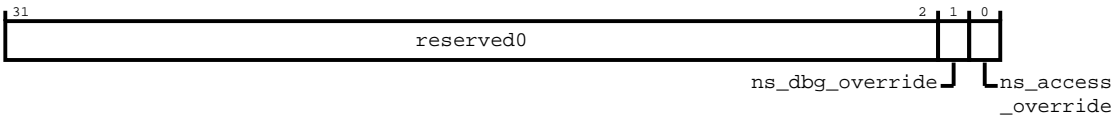
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the secure\_access register bit assignments.

Figure 16-353: Bit assignment diagram for the secure\_access register



The following table shows the secure\_access register bit descriptions.

Table 16-369: secure\_access bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Bits within this register segment are reserved for future product development.	RO	0x0
[1]	ns_dbg_override	Enables and disables Non-secure access to the APB completer node PMU and interface registers. The values are:  0 Disables Non-secure access to the APB completer node PMU and interface registers.  1 Enables Non-secure access to the APB completer node PMU and interface registers.	RW	0
[0]	ns_access_override	Enables and disables Non-secure access to the APB completer node registers. The values are:  0 Disables Non-secure access to the APB completer node registers  1 Enables Non-secure access to the APB completer node registers	RW	0

## 16.16.4 PMNI pmusela register

This register configures the PMU A crossbar to select particular events to monitor.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x0C

#### Type

RW

#### Reset value

0x00000000

### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the pmusela register bit assignments.

**Figure 16-354: Bit assignment diagram for the pmusela register**



The following table shows the pmusela register bit descriptions.

**Table 16-370: pmusela bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	Indicates ASNI event element 3 to monitor	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	Indicates ASNI event element 2 to monitor	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00

Bits	Name	Description	Type	Reset
[13:8]	event_select_1	Indicates ASNI event element 1 to monitor	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	Indicates ASNI event element 0 to monitor	RW	0b000000

### 16.16.5 PMNI pmuselb register

This register configures the PMU B crossbar to select particular events to monitor.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x10

##### Type

RW

##### Reset value

0x00000000

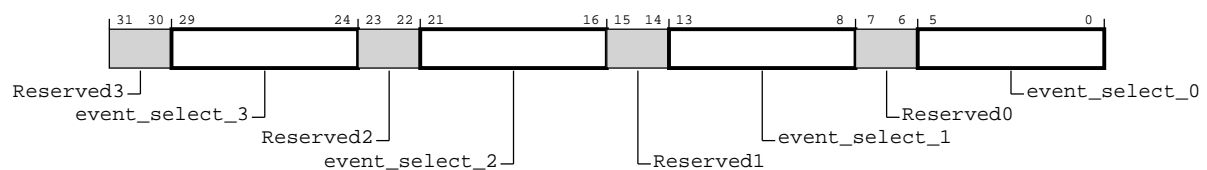
#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the pmuselb register bit assignments.

**Figure 16-355: Bit assignment diagram for the pmuselb register**



The following table shows the pmuselb register bit descriptions.

**Table 16-371: pmuselb bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	Reserved3	Bits within this register segment are reserved for future product development	RO	0b00
[29:24]	event_select_3	Indicates ASNI event element 3 to monitor	RW	0b000000
[23:22]	Reserved2	Bits within this register segment are reserved for future product development	RO	0b00
[21:16]	event_select_2	Indicates ASNI event element 2 to monitor	RW	0b000000
[15:14]	Reserved1	Bits within this register segment are reserved for future product development	RO	0b00
[13:8]	event_select_1	Indicates ASNI event element 1 to monitor	RW	0b000000
[7:6]	Reserved0	Bits within this register segment are reserved for future product development	RO	0b00
[5:0]	event_select_0	Indicates ASNI event element 0 to monitor	RW	0b000000

### 16.16.6 PMNI interface\_id\_0\_3 register

To configure APB interface IDs 0-3, use offset 0x014 in the PMNI\_INTERFACEID register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x14

##### Type

RO

##### Reset value

See individual bit resets.

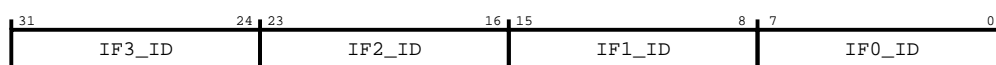
#### Constraints

None.

#### Bit descriptions

The following figure shows the interface\_id\_0\_3 register bit assignments.

**Figure 16-356: Bit assignment diagram for the interface\_id\_0\_3 register**



The following table shows the interface\_id\_0\_3 register bit descriptions.

**Table 16-372: interface\_id\_0\_3 bit descriptions**

Bits	Name	Description	Type	Reset
[31:24]	IF3_ID	APB interface ID 3	RO	Configuration dependent
[23:16]	IF2_ID	APB interface ID 2	RO	Configuration dependent
[15:8]	IF1_ID	APB interface ID 1	RO	Configuration dependent
[7:0]	IF0_ID	APB interface ID 0	RO	Configuration dependent

### 16.16.7 PMNI interface\_id\_4\_7 register

To configure APB interface IDs 4-7, use offset 0x018 in the PMNI\_INTERFACEID register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x18

##### Type

RO

##### Reset value

See individual bit resets.

#### Constraints

None.

#### Bit descriptions

The following figure shows the interface\_id\_4\_7 register bit assignments.

**Figure 16-357: Bit assignment diagram for the interface\_id\_4\_7 register**



The following table shows the interface\_id\_4\_7 register bit descriptions.

**Table 16-373: interface\_id\_4\_7 bit descriptions**

Bits	Name	Description	Type	Reset
[31:24]	IF7_ID	APB interface ID 7	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[23:16]	IF6_ID	APB interface ID 6	RO	Configuration dependent
[15:8]	IF5_ID	APB interface ID 5	RO	Configuration dependent
[7:0]	IF4_ID	APB interface ID 4	RO	Configuration dependent

### 16.16.8 PMNI interface\_id\_8\_11 register

To configure APB interface IDs 8-11, use offset 0x01C in the PMNI\_INTERFACEID register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x1C

##### Type

RO

##### Reset value

See individual bit resets.

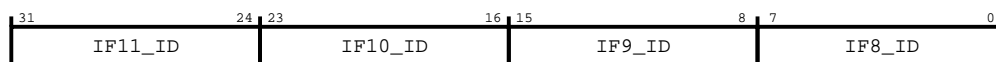
#### Constraints

None.

#### Bit descriptions

The following figure shows the interface\_id\_8\_11 register bit assignments.

**Figure 16-358: Bit assignment diagram for the interface\_id\_8\_11 register**



The following table shows the interface\_id\_8\_11 register bit descriptions.

**Table 16-374: interface\_id\_8\_11 bit descriptions**

Bits	Name	Description	Type	Reset
[31:24]	IF11_ID	APB interface ID 11	RO	Configuration dependent
[23:16]	IF10_ID	APB interface ID 10	RO	Configuration dependent
[15:8]	IF9_ID	APB interface ID 9	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[7:0]	IF8_ID	APB interface ID 8	RO	Configuration dependent

### 16.16.9 PMNI interface\_id\_12\_15 register

To configure APB interface IDs 12-15, use offset 0x020 in the PMNI\_INTERFACEID register.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x20

##### Type

RO

##### Reset value

See individual bit resets.

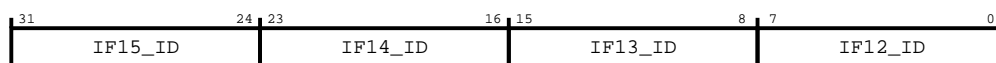
#### Constraints

None.

#### Bit descriptions

The following figure shows the interface\_id\_12\_15 register bit assignments.

**Figure 16-359: Bit assignment diagram for the interface\_id\_12\_15 register**



The following table shows the interface\_id\_12\_15 register bit descriptions.

**Table 16-375: interface\_id\_12\_15 bit descriptions**

Bits	Name	Description	Type	Reset
[31:24]	IF15_ID	APB interface ID 15	RO	Configuration dependent
[23:16]	IF14_ID	APB interface ID 14	RO	Configuration dependent
[15:8]	IF13_ID	APB interface ID 13	RO	Configuration dependent
[7:0]	IF12_ID	APB interface ID 12	RO	Configuration dependent



16.16.10 PMNI num\_sub\_features register

The number of subfeatures.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x24

Type

RO

Reset value

See individual bit resets.

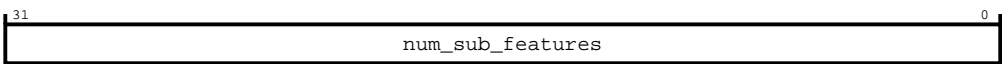
Constraints

None.

Bit descriptions

The following figure shows the num\_sub\_features register bit assignments.

Figure 16-360: Bit assignment diagram for the num\_sub\_features register



The following table shows the num\_sub\_features register bit descriptions.

Table 16-376: num\_sub\_features bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	num_sub_features	Number of subfeatures	RO	Configuration dependent

16.16.11 PMNI sub\_feature\_0\_type register

Subfeature 0 type.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x28

Type

RO

Reset value

See individual bit resets.

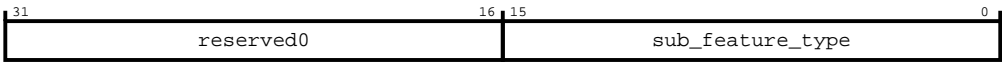
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_type register bit assignments.

Figure 16-361: Bit assignment diagram for the sub\_feature\_0\_type register



The following table shows the sub\_feature\_0\_type register bit descriptions.

Table 16-377: sub\_feature\_0\_type bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	reserved0	Bits within this register segment are reserved for future product development	RO	0x0
[15:0]	sub_feature_type	Subfeature 0 type	RO	Configuration dependent

16.16.12 PMNI sub\_feature\_0\_pointer register

Subfeature 0 pointer.

Configurations

The number of registers of this type that are present depends on the number of subfeatures in the interface.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x2C

Type

RO

Reset value

See individual bit resets.

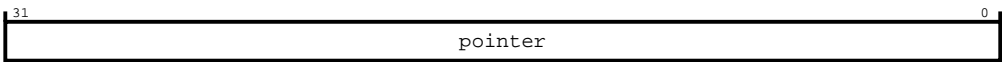
Constraints

None.

Bit descriptions

The following figure shows the sub\_feature\_0\_pointer register bit assignments.

Figure 16-362: Bit assignment diagram for the sub\_feature\_0\_pointer register



The following table shows the sub\_feature\_0\_pointer register bit descriptions.

Table 16-378: sub\_feature\_0\_pointer bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	pointer	Subfeature 0 pointer	RO	Configuration dependent

16.16.13 PMNI secure\_info register

Shows the security attribute for each of the APB interfaces downstream of the PMNI.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x30

Type

RO

Reset value

See individual bit resets.

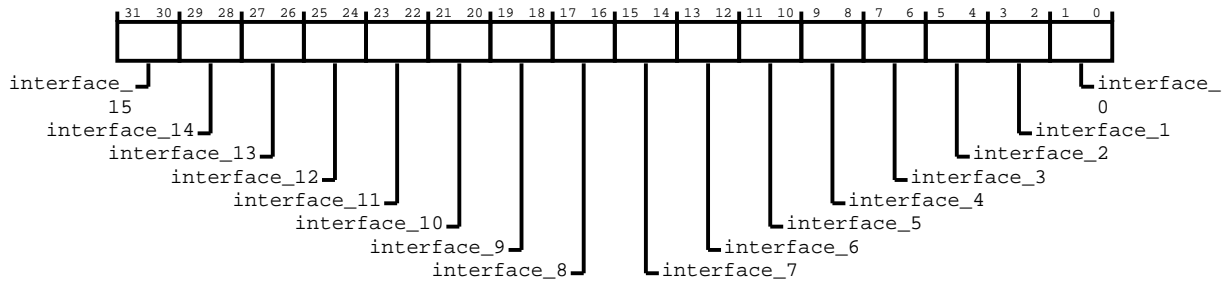
## Constraints

None.

## Bit descriptions

The following figure shows the secure\_info register bit assignments.

**Figure 16-363: Bit assignment diagram for the secure\_info register**



The following table shows the secure\_info register bit descriptions.

**Table 16-379: secure\_info bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	interface_15	Security attribute for interface 15.	RO	Configuration dependent
[29:28]	interface_14	Security attribute for interface 14.	RO	Configuration dependent
[27:26]	interface_13	Security attribute for interface 13.	RO	Configuration dependent
[25:24]	interface_12	Security attribute for interface 12.	RO	Configuration dependent
[23:22]	interface_11	Security attribute for interface 11.	RO	Configuration dependent
[21:20]	interface_10	Security attribute for interface 10.	RO	Configuration dependent
[19:18]	interface_9	Security attribute for interface 9.	RO	Configuration dependent
[17:16]	interface_8	Security attribute for interface 8.	RO	Configuration dependent
[15:14]	interface_7	Security attribute for interface 7.	RO	Configuration dependent
[13:12]	interface_6	Security attribute for interface 6.	RO	Configuration dependent
[11:10]	interface_5	Security attribute for interface 5.	RO	Configuration dependent
[9:8]	interface_4	Security attribute for interface 4.	RO	Configuration dependent
[7:6]	interface_3	Security attribute for interface 3.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[5:4]	interface_2	Security attribute for interface 2.	RO	Configuration dependent
[3:2]	interface_1	Security attribute for interface 1.	RO	Configuration dependent
[1:0]	interface_0	<p>Security attribute for interface 0:</p> <p><b>0b00</b> Software-programmable register to set the security attribute for the downstream completer.</p> <p><b>0b01</b> Pin exists and is used to pass the security attribute. Downstream filters out based on PPROT[1].</p> <p><b>0b02</b> Always Secure. Only Secure transactions access the completer attached to this APB requester interface.</p> <p><b>0b03</b> Always Non-secure. Both Secure and Non-secure transactions access the completer attached to this APB requester interface.</p>	RO	Configuration dependent

### 16.16.14 PMNI node\_features register

This register configures the node features. You can configure up to 16 APB interfaces for a PMNI. Use two bits to identify the APB protocol for a specific interface.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x40

##### Type

RO

##### Reset value

See individual bit resets.

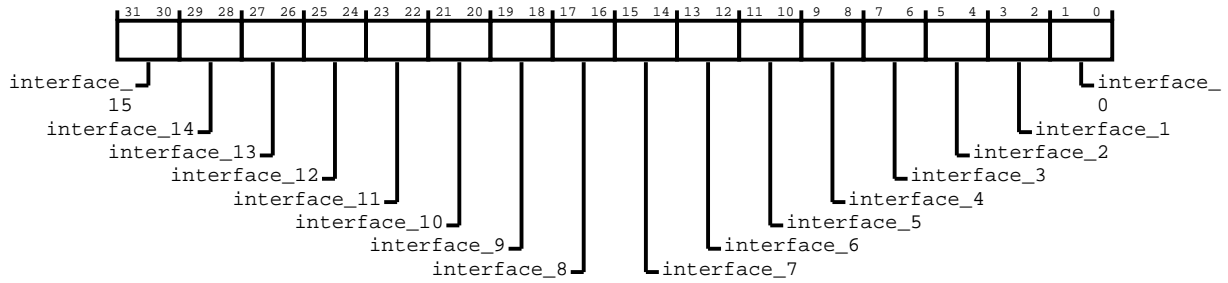
#### Constraints

None.

#### Bit descriptions

The following figure shows the node\_features register bit assignments.

**Figure 16-364: Bit assignment diagram for the node\_features register**



The following table shows the node\_features register bit descriptions.

**Table 16-380: node\_features bit descriptions**

Bits	Name	Description	Type	Reset
[31:30]	interface_15	Interface 15 APB protocol type.	RO	Configuration dependent
[29:28]	interface_14	Interface 14 APB protocol type.	RO	Configuration dependent
[27:26]	interface_13	Interface 13 APB protocol type.	RO	Configuration dependent
[25:24]	interface_12	Interface 12 APB protocol type.	RO	Configuration dependent
[23:22]	interface_11	Interface 11 APB protocol type.	RO	Configuration dependent
[21:20]	interface_10	Interface 10 APB protocol type.	RO	Configuration dependent
[19:18]	interface_9	Interface 9 APB protocol type.	RO	Configuration dependent
[17:16]	interface_8	Interface 8 APB protocol type.	RO	Configuration dependent
[15:14]	interface_7	Interface 7 APB protocol type.	RO	Configuration dependent
[13:12]	interface_6	Interface 6 APB protocol type.	RO	Configuration dependent
[11:10]	interface_5	Interface 5 APB protocol type.	RO	Configuration dependent
[9:8]	interface_4	Interface 4 APB protocol type.	RO	Configuration dependent
[7:6]	interface_3	Interface 3 APB protocol type.	RO	Configuration dependent
[5:4]	interface_2	Interface 2 APB protocol type.	RO	Configuration dependent
[3:2]	interface_1	Interface 1 APB protocol type.	RO	Configuration dependent
[1:0]	interface_0	Interface 0 APB protocol type. The encoding is common across all the interfaces:  <b>0b00</b> This value is reserved.  <b>0b01</b> APB3.  <b>0b10</b> APB4.  <b>0b11</b> APB5.	RO	Configuration dependent

16.16.15 PMNI node\_control register

This register indicates the security status, Secure or Non-secure, of APB interfaces that are attached to a PMNI.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x44

Type

RW

Reset value

See individual bit resets.

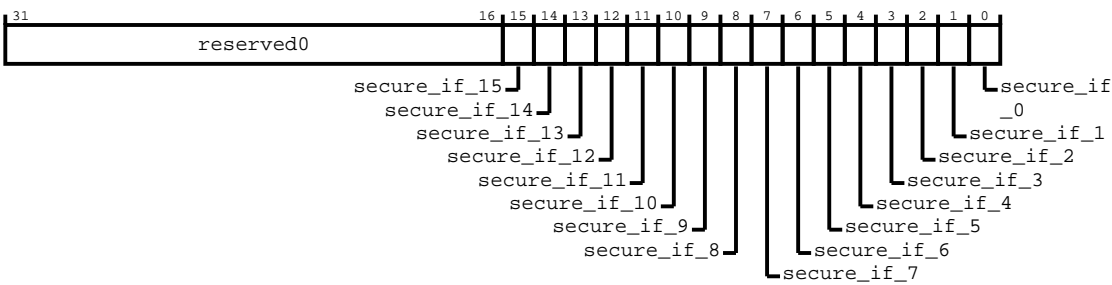
Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the node\_control register bit assignments.

Figure 16-365: Bit assignment diagram for the node\_control register



The following table shows the node\_control register bit descriptions.

Table 16-381: node\_control bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	reserved0	Bits within this register segment are reserved for future product development.	RO	0x0

Bits	Name	Description	Type	Reset
[15]	secure_if_15	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 15. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[14]	secure_if_14	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 14. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[13]	secure_if_13	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 13. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[12]	secure_if_12	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 12. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[11]	secure_if_11	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 11. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[10]	secure_if_10	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 10. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[9]	secure_if_9	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 9. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[8]	secure_if_8	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 8. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[7]	secure_if_7	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 7. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[6]	secure_if_6	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 6. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[5]	secure_if_5	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 5. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[4]	secure_if_4	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 4. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[3]	secure_if_3	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 3. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[2]	secure_if_2	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 2. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent
[1]	secure_if_1	Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 1. Configuration procedures are the same as those of PMNI APB interface 0.	RW	Configuration dependent



Bits	Name	Description	Type	Reset
[0]	secure_if_0	<p>Configure the security status, either Secure or Non-secure, of the downstream completer connected to PMNI APB interface 0:</p> <p><b>0</b></p> <p>Secure. Only Secure transactions can travel downstream.</p> <p><b>1</b></p> <p>Non-secure. Both Secure and Non-secure transactions can travel downstream.</p> <p>This register bit is relevant based on the secure_transfers field in the PMNI_SECURE_INFO register. If secure_transfers is 0b00, the PPROT pin is unavailable. This register bit determines the security attribute of the downstream completer. The security access permission check occurs within the PMNI. If secure_transfers is 0b01, the PPROT pin is supported downstream of the PMNI. The incoming security attribute is passed on to the pin, so this register bit is irrelevant. If the incoming request is Non-secure and the downstream completer is configured as Secure, then the transaction is not sent downstream. A Non-secure read transaction returns zero data. The data corresponding to a Non-secure write transaction is dropped but a protocol-compliant write response is returned. The read or write response does not contain an error indication. If secure_transfers is 0b02 or 0b03, then the PPROT pin is unavailable. However, the APB interface security attribute is set at build time to either Always Secure or Always Non-secure. This register bit becomes read-only. However, if secure_transfers is 0b03, the reset value is 1, and if secure_transfers is 0b02, the reset value is 0. The width of this PMNI Control register depends on the number of APB ports, which can be up to 16 ports. A single bit is assigned for each port to indicate the security status, either Secure or Non-secure, of the downstream completer.</p>	RW	Configuration dependent

### 16.16.16 PMNI silicon\_debug register

This register monitors the status of requester interface channels.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x80

##### Type

RW

##### Reset value

0x00000000

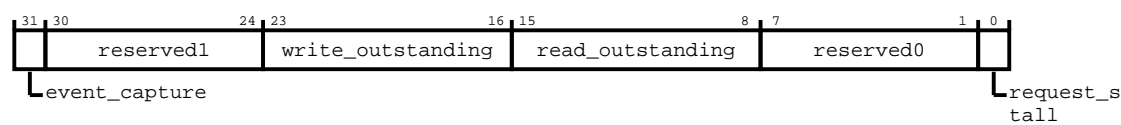
#### Constraints

Only accessible using Secure transactions, unless the ns\_debug\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

Bit descriptions

The following figure shows the silicon\_debug register bit assignments.

Figure 16-366: Bit assignment diagram for the silicon\_debug register



The following table shows the silicon\_debug register bit descriptions.

Table 16-382: silicon\_debug bit descriptions

Bits	Name	Description	Type	Reset
[31]	event_capture	Enable event capture	RW	0
[30:24]	reserved1	Bits within this register segment are reserved for future product development	RO	0b00000000
[23:16]	write_outstanding	Indicates that the interface has writes that are outstanding	RO	0x0
[15:8]	read_outstanding	Indicates that the interface has reads that are outstanding	RO	0x0
[7:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0b00000000
[0]	request_stall	Indicates that a read request has stalled	RO	0

16.16.17 PMNI idm\_device\_id register

This register indicates the statically configured device ID value and is implemented if IDM is enabled.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x100

Type

RO

Reset value

See individual bit resets.

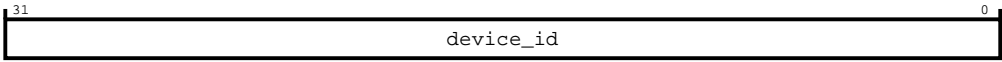
Constraints

None.

Bit descriptions

The following figure shows the `idm_device_id` register bit assignments.

Figure 16-367: Bit assignment diagram for the `idm_device_id` register



The following table shows the `idm_device_id` register bit descriptions.

Table 16-383: `idm_device_id` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	device_id	Returns the statically configured ID value	RO	Configuration dependent

16.16.18 PMNI `idm_config` register

This register enables transaction logging, error detection, timeout detection, access control, and reset control.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x104

Type

RW

Reset value

See individual bit resets.

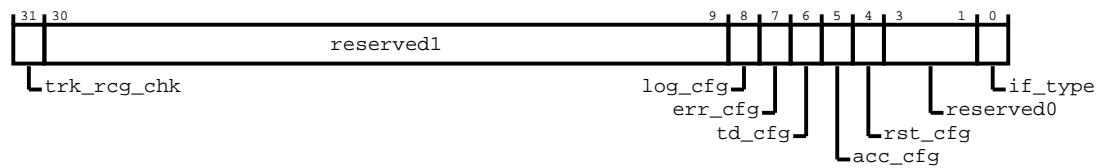
Constraints

None.

Bit descriptions

The following figure shows the `idm_config` register bit assignments.

**Figure 16-368: Bit assignment diagram for the idm\_config register**



The following table shows the idm\_config register bit descriptions.

**Table 16-384: idm\_config bit descriptions**

Bits	Name	Description	Type	Reset
[31]	trk_rcg_chk	Tracker Regional Clock Gating (RCG) chicken bit	RW	0
[30:9]	reserved1	Bits within this register segment are reserved for future product development	RO	0x0
[8]	log_cfg	Transaction logging present	RO	1
[7]	err_cfg	Error detection present	RO	1
[6]	td_cfg	Timeout detection present	RO	1
[5]	acc_cfg	Access control present	RO	1
[4]	rst_cfg	Reset control present	RO	1
[3:1]	reserved0	Bits within this register segment are reserved for future product development	RO	0b000
[0]	if_type	Interface type  0 Completer 1 Requester	RO	Configuration dependent

### 16.16.19 PMNI idm\_errctlr register

This register controls how errors are handled.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x108

##### Type

RW

## Reset value

0x00000000

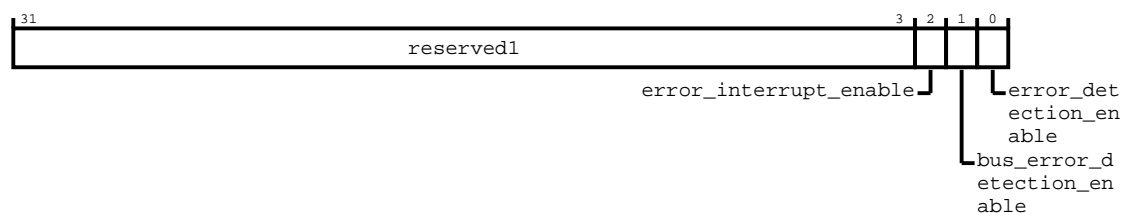
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the idm\_errctlr register bit assignments.

**Figure 16-369: Bit assignment diagram for the idm\_errctlr register**



The following table shows the idm\_errctlr register bit descriptions.

**Table 16-385: idm\_errctlr bit descriptions**

Bits	Name	Description	Type	Reset
[31:3]	reserved1	Reserved	RO	0x0
[2]	error_interrupt_enable	<p>A configurable register used to enable or disable error interrupt for an uncorrected error. The values are:</p> <p><b>0</b></p> <p>Disable error interrupt for an uncorrected error</p> <p><b>1</b></p> <p>Enable error interrupt for an uncorrected error</p>	RW	0
[1]	bus_error_detection_enable	<p>Enable bus error detection:</p> <p><b>0</b></p> <p>Disable bus error detection</p> <p><b>1</b></p> <p>Enabled when an error is detected and idm_errctlr [ed] is enabled. The error is logged if the transaction log is empty. If not, the logged transaction overflow bit is set. An error interrupt event is generated (unless masked).</p>	RW	0
[0]	error_detection_enable	<p>Error detection global enable</p> <p><b>0</b></p> <p>Disable error detection</p> <p><b>1</b></p> <p>Enable error detection</p>	RW	0

16.16.20 PMNI idm\_errstatus register

This register indicates the error status of Secure transactions. If timeout is configured, but error logging is not configured, then OF is never set and SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x110

Type

RW

Reset value

0x00000000

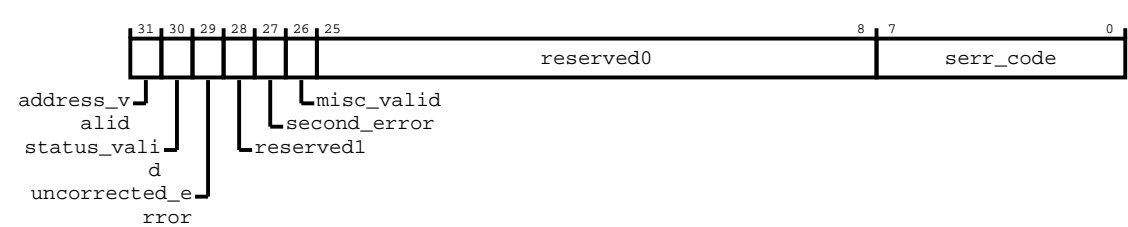
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the idm\_errstatus register bit assignments.

Figure 16-370: Bit assignment diagram for the idm\_errstatus register



The following table shows the idm\_errstatus register bit descriptions.

**Table 16-386: idm\_errstatus bit descriptions**

Bits	Name	Description	Type	Reset
[31]	address_valid	<p>Address valid. The values are:</p> <p><b>0</b></p> <p>ERRADDR is not valid.</p> <p><b>1</b></p> <p>ERRADDR contains an address that is associated with the highest priority error which this record records.</p> <p>This bit ignores writes if IDM_ERRSTATUS.UE is set to 1 and is not cleared to zero in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[30]	status_valid	<p>Status register is valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRSTATUS not valid</p> <p><b>1</b></p> <p>IDM_ERRSTATUS valid. At least one error has been recorded.</p> <p>This bit ignores writes if any of the following fields is set to 1 and is not being cleared to zero in the same write:</p> <ul style="list-style-type: none"> <li>IDM_ERRSTATUS.UE</li> <li>IDM_ERRSTATUS.AV</li> <li>IDM_ERRSTATUS.OF * IDM_ERRSTATUS.MV</li> </ul> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred</p> <p>This bit ignores writes if IDM_ERRSTATUS.OF is set to 1 and is not being cleared to zero in the same write. This bit is not valid and reads <b>UNKNOWN</b> if IDM_ERRSTATUS.V is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Reserved	RO	0

Bits	Name	Description	Type	Reset
[27]	second_error	Returns whether a second error has been received while handling a first error. The values are:  <b>1</b> Second error received  <b>0</b> No other error received  This bit is read, or write 1 to clear  Write 1 to clear.	RW	0
[26]	misc_valid	Miscellaneous registers valid. The values are:  <b>0</b> IDM_ERRMISC0 and IDM_ERRMISC1 not valid  <b>1</b> The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISC0 and IDM_ERRMISC1 registers contains additional information for an error that this record records.  This bit ignores writes if IDM_ERRSTATUS.UE is set to 1, and is not being cleared to 0 in the same write. This bit is a read, or write 1 to clear.  Write 1 to clear.	RW	0
[25:8]	reserved0	Reserved	RO	0x0
[7:0]	serr_code	Primary error code. Indicates the type of error. The values are:  <b>00</b> No error  <b>13</b> Illegal address - decode error  <b>18</b> Error response from completer  <b>20</b> Internal timeout	RO	0x0

### 16.16.21 PMNI idm\_erraddr\_lsb register

This register is the error log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit



Address offset

0x114

Type

RO

Reset value

0x00000000

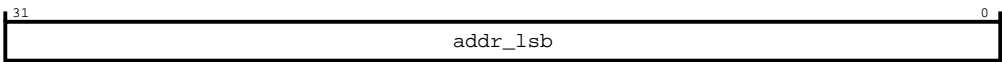
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_lsb` register bit assignments.

Figure 16-371: Bit assignment diagram for the `idm_erraddr_lsb` register



The following table shows the `idm_erraddr_lsb` register bit descriptions.

Table 16-387: `idm_erraddr_lsb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.16.22 PMNI `idm_erraddr_msb` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x118

Type

RO

Reset value

0x00000000

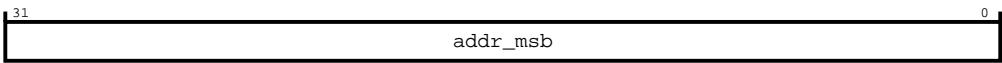
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_erraddr_msb` register bit assignments.

Figure 16-372: Bit assignment diagram for the `idm_erraddr_msb` register



The following table shows the `idm_erraddr_msb` register bit descriptions.

Table 16-388: `idm_erraddr_msb` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.16.23 PMNI `idm_errmisc0` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x128

Type

RO

Reset value

0x00000000

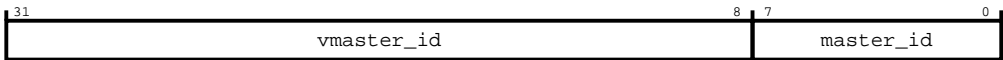
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_errmisc0` register bit assignments.

Figure 16-373: Bit assignment diagram for the `idm_errmisc0` register



The following table shows the `idm_errmisc0` register bit descriptions.

Table 16-389: `idm_errmisc0` bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming AXI AxiD into ASNI of the transaction causing an error. The assumption here is there is no manipulation of incoming AXI AxiD in ASNI.	RO	0x0
[7:0]	master_id	The ASNI Node ID of the transaction causing an error	RO	0x0

16.16.24 PMNI `idm_errmisc1` register

This register is the error log of Secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x12C

Type

RO

Reset value

0x00000000

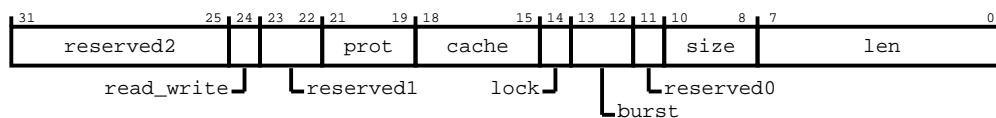
Constraints

Only accessible using Secure transactions.

Bit descriptions

The following figure shows the `idm_errmisc1` register bit assignments.

**Figure 16-374: Bit assignment diagram for the idm\_errmisc1 register**



The following table shows the idm\_errmisc1 register bit descriptions.

**Table 16-390: idm\_errmisc1 bit descriptions**

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Reserved	RO	0b0000000
[24]	read_write	The AXI read or write information of a transaction causing an error  1 Write  0 Read	RO	0
[23:22]	reserved1	Reserved	RO	0b00
[21:19]	prot	The AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	The AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	The AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	The AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Reserved	RO	0
[10:8]	size	The AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	The AXI len information of a transaction causing an error.	RO	0x0

### 16.16.25 PMNI idm\_access\_control register

This register controls the state, gated or ungated, of a device.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x130

##### Type

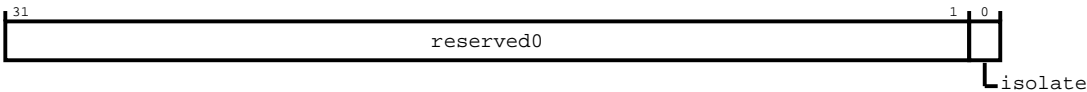
RW

**Reset value**  
0x00000000

**Constraints**  
Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

**Bit descriptions**  
The following figure shows the idm\_access\_control register bit assignments.

**Figure 16-375: Bit assignment diagram for the idm\_access\_control register**



The following table shows the idm\_access\_control register bit descriptions.

**Table 16-391: idm\_access\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	isolate	Perform gating off a device. Reading 1 indicates that the completer device is gated or isolated. Reading 0 indicates that the completer device is ungated or de-isolated. Write 1 to enter gated state. Write 0 to exit gated state. There is some delay to updating this field with the intended write value. Exit from gated state is only successful if there are no outstanding transactions and all error status register bits are cleared. Entry into gated state is only successful if there are no outstanding transactions. While in pending isolation entry state or in active isolation state, a write of 1 to this bit causes reentry to isolation state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 0 is ignored. While in pending isolation exit state, a write of 0 to this bit causes a re-exit to the exit state. The write causes the write_received and read_received fields of IDM_ACCESS_STATUS, and the IDM_access_readid and IDM_access_writeid registers to be cleared. A write of 1 is ignored.	RW	0

16.16.26 PMNI idm\_access\_status register

This register indicates the access status for Secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:  
**Width**  
32-bit

## Address offset

0x134

## Type

RO

## Reset value

0x00000002

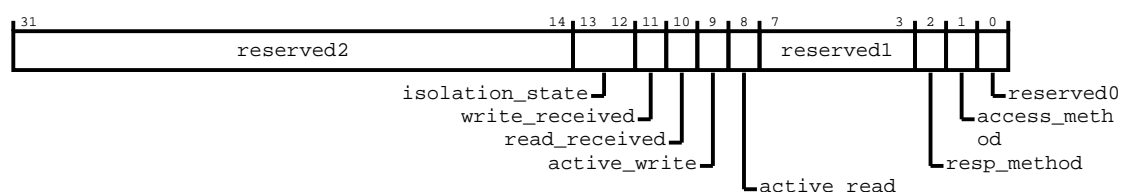
## Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the `idm_access_status` register bit assignments.

**Figure 16-376: Bit assignment diagram for the `idm_access_status` register**



The following table shows the `idm_access_status` register bit descriptions.

### Table 16-392: idm\_access\_status bit descriptions

Bits	Name	Description	Type	Reset
[31:14]	reserved2	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[13:12]	isolation_state	<p>Isolation status:</p> <p><b>00</b></p> <p>Isolation exit or entry is successful or not in gated or isolation state</p> <p><b>01</b></p> <p>Isolation exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus</p> <p><b>10</b></p> <p>Isolation entry is unsuccessful or pending because of outstanding transactions</p> <p><b>11</b></p> <p>Reserved</p>	RO	0b00
[11]	write_received	<p>A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on:</p> <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0

Bits	Name	Description	Type	Reset
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 0 to bit[0] of the IDM_ACCESS_CONTROL register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:3]	reserved1	Reserved, <b>UNDEFINED</b> , write as zero	RO	0b00000
[2]	resp_method	Indicates device generates errors in gated access.	RO	0
[1]	access_method	Wait for all outstanding to complete, then block input.	RO	1
[0]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0

### 16.16.27 PMNI idm\_access\_readid register

This register is the access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x138

##### Type

RO

##### Reset value

0x00000000

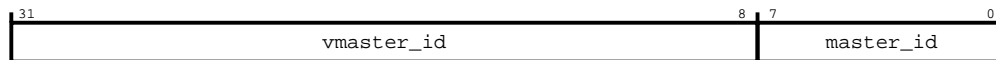
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_access\_readid register bit assignments.

**Figure 16-377: Bit assignment diagram for the `idm_access_readid` register**



The following table shows the `idm_access_readid` register bit descriptions.

**Table 16-393: `idm_access_readid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming signal into the endpoint of the first transaction to arrive after isolation when the <code>active_read</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the ARID of the transaction on ASNI and contains the HMASTER on HSNi. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNi of the first transaction to arrive after isolation when the <code>active_read</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

## 16.16.28 PMNI `idm_access_writeid` register

This register is the access log of Secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x13C

#### Type

RO

#### Reset value

0x00000000

### Constraints

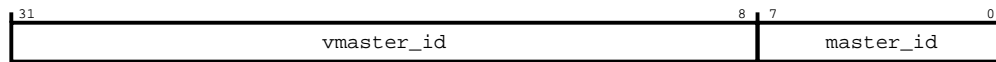
Only accessible using Secure transactions.

### Bit descriptions

The following figure shows the `idm_access_writeid` register bit assignments.



**Figure 16-378: Bit assignment diagram for the `idm_access_writeid` register**



The following table shows the `idm_access_writeid` register bit descriptions.

**Table 16-394: `idm_access_writeid` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AWID signal into the endpoint of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH. This field depends on the incoming endpoint. Therefore <code>vmaster_id</code> contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the <code>vmaster_id</code> matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the <code>active_write</code> field of the <code>IDM_ACCESS_STATUS</code> register is HIGH.	RO	0x0

## 16.16.29 PMNI `idm_reset_control` register

This register controls the reset of a device that is attached to the interconnect.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x140

#### Type

RW

#### Reset value

0x00000002

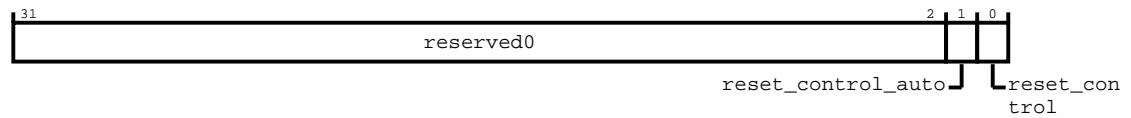
### Constraints

Only accessible using Secure transactions, unless the `ns_access_override` bit is set in the `secure_access` register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

### Bit descriptions

The following figure shows the `idm_reset_control` register bit assignments.

**Figure 16-379: Bit assignment diagram for the idm\_reset\_control register**



The following table shows the idm\_reset\_control register bit descriptions.

**Table 16-395: idm\_reset\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:2]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[1]	reset_control_auto	<p>Configures the device for auto or internal reset mode. For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. There are several constraints on this field:</p> <ul style="list-style-type: none"> <li>You can only change this field during initialization or when the interface is fully quiesced. * Arm does not support changing this field while the interface is active. If you change this field during runtime, behavior is <b>UNPREDICTABLE</b>.</li> </ul> <p>Reads have the following effect:</p> <p><b>1</b></p> <p>A read of 1 indicates that the device is in auto or internal reset mode.</p> <p><b>0</b></p> <p>A read of 0 indicates that the device is not in auto or internal reset mode.</p> <p>Writes have the following effect:</p> <p><b>1</b></p> <p>A write of 1 configures the device for auto or internal reset mode.</p> <p><b>0</b></p> <p>A write of 0 disables auto or internal reset mode.</p> <p>For more information on IDM soft reset modes, see the IDM soft reset mode section of the <i>Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual</i>. Bit[1] of the IDM_RESET_CONTROL register is 1 out of reset. This bit enables internal recovery mode out of reset. When not in auto reset mode and a timeout is detected, a write of 1 to the IDM_RESET_CONTROL.reset field initiates internal recovery mode. Changing this bit while the interface is not in idle mode results in <b>UNPREDICTABLE</b> behavior.</p>	RO	1

Bits	Name	Description	Type	Reset
[0]	reset_control	<p>Performs soft reset of attached device. If the auto bit is set to 1 the network interface gates the external interface, however the soft reset pin is not activated. If the auto bit is 0, the interfaces are not gated until there is a write to bit[0]. In this case, the soft reset pin is activated. Writes have the following effect:</p> <p><b>1</b></p> <p>Request the attached device to enter reset. If the write occurs before soft reset exit has occurred, the write is ignored.</p> <p><b>0</b></p> <p>Request the attached device to exit reset. If the write occurs before soft reset entry has occurred, the write is ignored.</p> <p>Software polls this register to determine if soft reset entry or exit has occurred, using the following values:</p> <p><b>1</b></p> <p>Indicates that the device is in reset.</p> <p><b>0</b></p> <p>Indicates that the device is not in reset.</p> <p>This register value updates to reflect a request for reset entry or reset exit, but the update can only occur after required internal conditions are met. Until these conditions are met, a read to this register returns the old value. For example, outstanding transactions currently being handled must complete before this register value updates. To ensure reset propagation within the device, it is the responsibility of the software to permit enough cycles after soft reset assertion is reflected in the IDM_RESET_CONTROL register before exiting soft reset by triggering a write of 0. If this responsibility is not met, the behavior is <b>UNDEFINED</b> or <b>UNPREDICTABLE</b>. When this register value is 1, the external soft reset pin that connects to the attached AXI requester or completer device is asserted, using the correct polarity of the reset pin. When this register value is 0, the external soft reset pin that connects to the attached AXI requester or completer device is deasserted, using the correct polarity of the reset pin. When in pending soft reset entry state or in active soft reset state, a write of 1 to this bit causes reentry to soft reset state. This write causes the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers to be cleared. A write of 0 is ignored. While in pending soft reset exit state, a write of 0 to this bit causes re-exit to exit state. A write of 0 also clears the write_received and read_received fields of the IDM_RESET_STATUS, IDM_RESET_READID, and IDM_RESET_WRITEID registers. A write of 1 is ignored.</p>	RW	0

### 16.16.30 PMNI idm\_reset\_status register

This register indicates mostly the reset status of Secure transactions. However, the rst\_exit\_state field indicates reset exit state of Secure or Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

#### Width

32-bit

## Address offset

0x144

## Type

RO

## Reset value

0x00000000

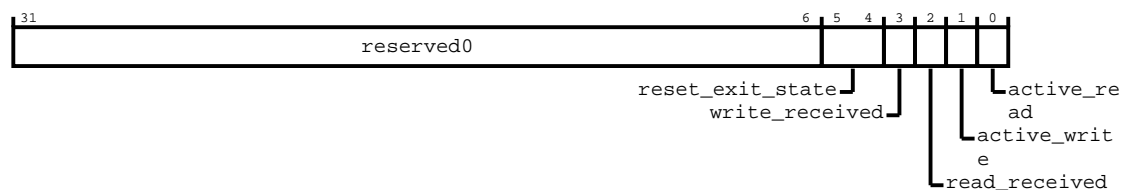
## Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

## Bit descriptions

The following figure shows the idm\_reset\_status register bit assignments.

**Figure 16-380: Bit assignment diagram for the idm\_reset\_status register**



The following table shows the idm\_reset\_status register bit descriptions.

**Table 16-396: idm\_reset\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:6]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[5:4]	reset_exit_state	Reset exit state  <b>00</b> Reset exit or entry is successful or not in reset state  <b>01</b> Reset exit is unsuccessful or pending because of uncleared error status bits, idm_errstatus  <b>10</b> Reset exit is unsuccessful or pending because of outstanding transactions  <b>11</b> Reset exit is unsuccessful or pending because of both uncleared error status bits and outstanding transactions	RO	0b00
[3]	write_received	A 1 indicates that an active Secure write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0

### 16.16.31 PMNI idm\_reset\_readid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x148

##### Type

RO

##### Reset value

0x00000000

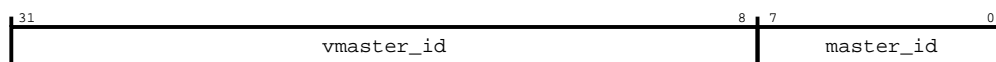
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_readid register bit assignments.

**Figure 16-381: Bit assignment diagram for the idm\_reset\_readid register**



The following table shows the idm\_reset\_readid register bit descriptions.

**Table 16-397: idm\_reset\_readid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.16.32 PMNI idm\_reset\_writeid register

This register is the reset access log of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x14C

##### Type

RO

##### Reset value

0x00000000

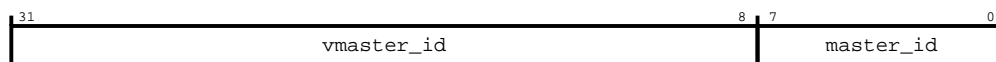
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid register bit assignments.

**Figure 16-382: Bit assignment diagram for the idm\_reset\_writeid register**



The following table shows the idm\_reset\_writeid register bit descriptions.

**Table 16-398: idm\_reset\_writeid bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS register is HIGH.	RO	0x0

### 16.16.33 PMNI idm\_timeout\_control register

This register is present when timeout detection is configured.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x150

##### Type

RW

##### Reset value

0x00000000

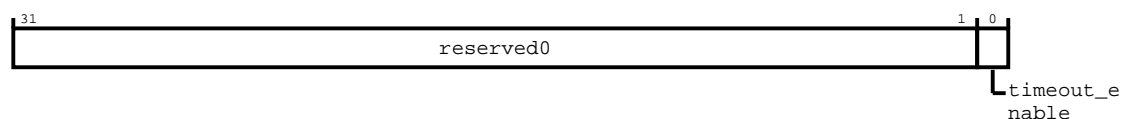
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the idm\_timeout\_control register bit assignments.

**Figure 16-383: Bit assignment diagram for the idm\_timeout\_control register**



The following table shows the idm\_timeout\_control register bit descriptions.

**Table 16-399: idm\_timeout\_control bit descriptions**

Bits	Name	Description	Type	Reset
[31:1]	reserved0	Reserved	RO	0x0
[0]	timeout_enable	<p>Timeout detection enable</p> <p><b>0</b></p> <p>Disabled</p> <p><b>1</b></p> <p>Enabled when a timeout is detected. The timeout is logged if the transaction log is empty. If not, the logged transaction overflow bit is set.</p> <p>A timeout interrupt event is generated, unless it is masked.</p>	RW	0

### 16.16.34 PMNI idm\_timeout\_value register

This register controls the duration that is used to determine if a transaction has timed out.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x154

##### Type

RW

##### Reset value

0x00000004

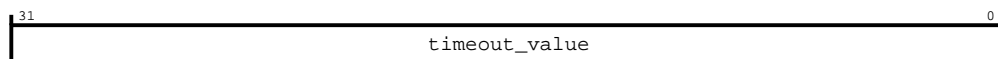
#### Constraints

Only accessible using Secure transactions, unless the ns\_access\_override bit is set in the secure\_access register of the relevant node or subfeature. If so, Non-secure accesses to this register are permitted.

#### Bit descriptions

The following figure shows the idm\_timeout\_value register bit assignments.

**Figure 16-384: Bit assignment diagram for the idm\_timeout\_value register**





The following table shows the `idm_timeout_value` register bit descriptions.

**Table 16-400: `idm_timeout_value` bit descriptions**

Bits	Name	Description	Type	Reset
[31:0]	<code>timeout_value</code>	Controls the duration that is used to determine if a transaction has timed out. The actual duration is $2^{\text{timeout\_exponent}}$ cycles. The minimum value is 4. Values of 0, 1, 2, or 3 are treated as 4. The maximum value is 30. Values greater than 30 are treated as 30.	RW	0x4

### 16.16.35 PMNI `idm_interrupt_status` register

This register indicates the interrupt status of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x158

##### Type

RW

##### Reset value

0x00000000

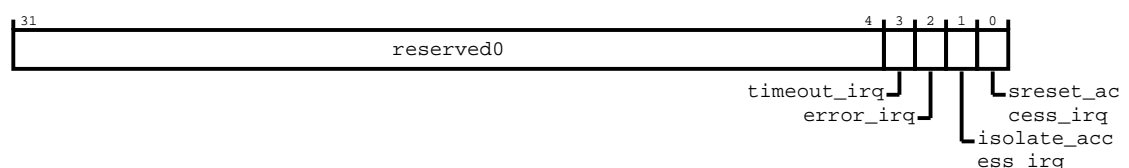
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the `idm_interrupt_status` register bit assignments.

**Figure 16-385: Bit assignment diagram for the `idm_interrupt_status` register**



The following table shows the `idm_interrupt_status` register bit descriptions.

**Table 16-401: idm\_interrupt\_status bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

### 16.16.36 PMNI idm\_interrupt\_mask register

This register is the interrupt mask of Secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x15C

##### Type

RW

##### Reset value

0x00000000

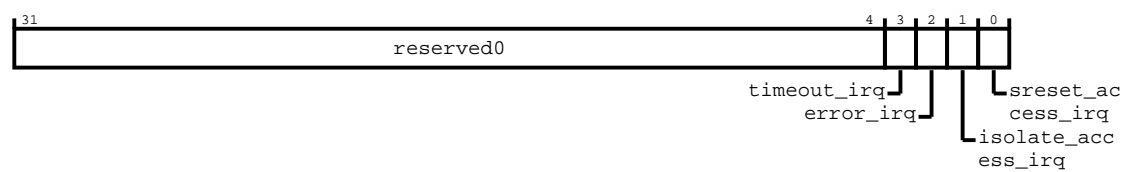
#### Constraints

Only accessible using Secure transactions.

#### Bit descriptions

The following figure shows the idm\_interrupt\_mask register bit assignments.

Figure 16-386: Bit assignment diagram for the idm\_interrupt\_mask register



The following table shows the idm\_interrupt\_mask register bit descriptions.

Table 16-402: idm\_interrupt\_mask bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

16.16.37 PMNI idm\_errstatus\_ns register

This register indicates the error status of Non-secure transactions. If timeout is configured, but error logging is not configured then OF is never set. Therefore SERR only reads as no error or timeout error.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x160

Type

RW

Reset value

0x00000000

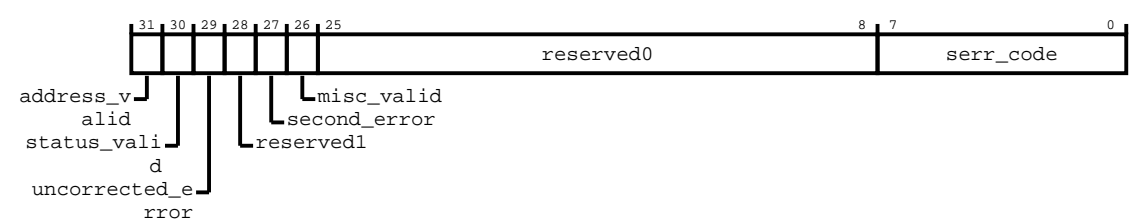
Constraints

None.

Bit descriptions

The following figure shows the `idm_errstatus_ns` register bit assignments.

Figure 16-387: Bit assignment diagram for the `idm_errstatus_ns` register



The following table shows the `idm_errstatus_ns` register bit descriptions.

Table 16-403: `idm_errstatus_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31]	<code>address_valid</code>	Address valid. The values are:  <b>0</b>  ERRADDR is not valid.  <b>1</b>  ERRADDR contains an address that is associated with the highest priority error that this record captures.  This bit ignores writes if the <code>ue</code> field of the <code>IDM_ERRSTATUS_NS</code> register is set to 1 and is not cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0
[30]	<code>status_valid</code>	Status register valid. The values are:  <b>0</b>  <code>IDM_ERRSTATUS_NS</code> is not valid.  <b>1</b>  <code>IDM_ERRSTATUS_NS</code> is valid. At least one error has been recorded.  This bit ignores writes if the <code>ue</code> field of the <code>IDM_ERRSTATUS_NS</code> register is set to 1 and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.  Write 1 to clear.	RW	0

Bits	Name	Description	Type	Reset
[29]	uncorrected_error	<p>Uncorrected error. The values are:</p> <p><b>0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred.</p> <p><b>1</b></p> <p>At least one detected error was not corrected and not deferred.</p> <p>This bit ignores writes if the oe field of the IDM_ERRSTATUS_NS register is set to 1 and is not being cleared to 0 in the same write. This bit is not valid and reads <b>UNKNOWN</b> if the v field of the IDM_ERRSTATUS_NS register is set to 0. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[28]	reserved1	Reserved	RO	0
[27]	second_error	<p>Returns whether a second error has been received while handling a first error. The values are:</p> <p><b>1</b></p> <p>Second error received</p> <p><b>0</b></p> <p>No other error received</p> <p>This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[26]	misc_valid	<p>Miscellaneous registers valid. The values are:</p> <p><b>0</b></p> <p>IDM_ERRMISCO_NS and IDM_ERRMISC1_NS are not valid.</p> <p><b>1</b></p> <p>The <b>IMPLEMENTATION DEFINED</b> contents of the IDM_ IDM_ERRMISCO_NS and IDM_ERRMISC1_NS registers contains additional information for an error that this record captures.</p> <p>This bit ignores writes if the ue field of the IDM_ERRSTATUS_NS register is set to 1, and is not being cleared to 0 in the same write. This bit is read, or write 1 to clear.</p> <p>Write 1 to clear.</p>	RW	0
[25:8]	reserved0	Reserved	RO	0x0
[7:0]	serr_code	<p>Primary error code, indicates the type of error. The values are:</p> <p><b>00</b></p> <p>No error</p> <p><b>13</b></p> <p>Illegal address - decode error</p> <p><b>18</b></p> <p>Error response from completer</p> <p><b>20</b></p> <p>Internal timeout</p>	RO	0x0

16.16.38 PMNI idm\_erraddr\_lsb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x164

Type

RO

Reset value

0x00000000

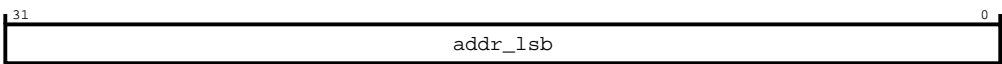
Constraints

None.

Bit descriptions

The following figure shows the idm\_erraddr\_lsb\_ns register bit assignments.

Figure 16-388: Bit assignment diagram for the idm\_erraddr\_lsb\_ns register



The following table shows the idm\_erraddr\_lsb\_ns register bit descriptions.

Table 16-404: idm\_erraddr\_lsb\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_lsb	Returns bits [31:0] of an address causing an error	RO	0x0

16.16.39 PMNI idm\_erraddr\_msb\_ns register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x168

Type

RO

Reset value

0x00000000

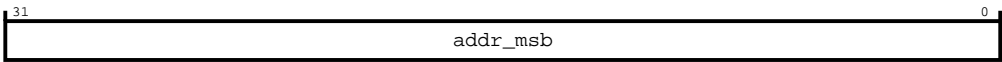
Constraints

None.

Bit descriptions

The following figure shows the `idm_erraddr_msb_ns` register bit assignments.

Figure 16-389: Bit assignment diagram for the `idm_erraddr_msb_ns` register



The following table shows the `idm_erraddr_msb_ns` register bit descriptions.

Table 16-405: `idm_erraddr_msb_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	addr_msb	Returns bits [63:32] of an address causing an error	RO	0x0

16.16.40 PMNI `idm_errmisc0_ns` register

This register is the error log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x178

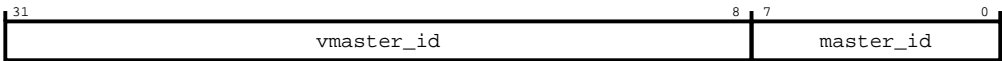
**Type**  
RO

**Reset value**  
0x00000000

**Constraints**  
None.

**Bit descriptions**  
The following figure shows the `idm_errmisc0_ns` register bit assignments.

**Figure 16-390: Bit assignment diagram for the `idm_errmisc0_ns` register**



The following table shows the `idm_errmisc0_ns` register bit descriptions.

**Table 16-406: `idm_errmisc0_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	<code>vmaster_id</code>	The incoming AXI AxiD into ASNI of the transaction causing an error. The assumption is no manipulation of incoming AXI AxiD in ASNI.	RO	0x0
[7:0]	<code>master_id</code>	The ASNI Node ID of the transaction causing an error.	RO	0x0

16.16.41 PMNI `idm_errmisc1_ns` register

This register is the error log of Non-secure transactions.

**Configurations**  
This register is available in all configurations.

**Attributes**  
Its characteristics are:

**Width**  
32-bit

**Address offset**  
0x17C

**Type**  
RO

**Reset value**  
0x00000000



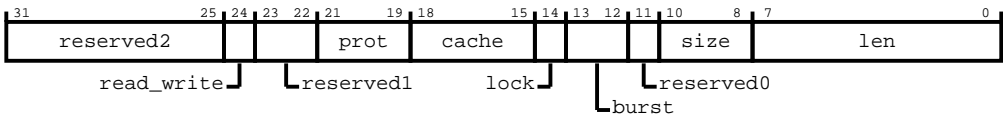
Constraints

None.

Bit descriptions

The following figure shows the `idm_errmisc1_ns` register bit assignments.

Figure 16-391: Bit assignment diagram for the `idm_errmisc1_ns` register



The following table shows the `idm_errmisc1_ns` register bit descriptions.

Table 16-407: `idm_errmisc1_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:25]	reserved2	Reserved	RO	0b0000000
[24]	read_write	Returns the AXI read or write information of a transaction causing an error:  1           Write  0           Read	RO	0
[23:22]	reserved1	Reserved	RO	0b00
[21:19]	prot	Returns the AXI prot information of a transaction causing an error.	RO	0b000
[18:15]	cache	Returns the AXI cache information of a transaction causing an error.	RO	0b0000
[14]	lock	Returns the AXI lock information of a transaction causing an error.	RO	0
[13:12]	burst	Returns the AXI burst information of a transaction causing an error.	RO	0b00
[11]	reserved0	Reserved	RO	0
[10:8]	size	Returns the AXI size information of a transaction causing an error.	RO	0b000
[7:0]	len	Returns the AXI len information of a transaction causing an error.	RO	0x0

16.16.42 PMNI `idm_access_status_ns` register

This register indicates the access status for Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

## Address offset

0x184

## Type

RO

## Reset value

0x00000000

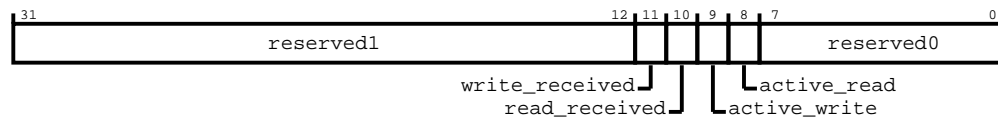
## Constraints

None.

## Bit descriptions

The following figure shows the `idm_access_status_ns` register bit assignments.

**Figure 16-392: Bit assignment diagram for the `idm_access_status_ns` register**



The following table shows the `idm_access_status_ns` register bit descriptions.

**Table 16-408: `idm_access_status_ns` bit descriptions**

Bits	Name	Description	Type	Reset
[31:12]	reserved1	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[11]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of the <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[10]	read_received	A 1 indicates that an active read transaction has occurred since the IDM entered the isolation state. This bit is cleared to zero on: <ul style="list-style-type: none"> <li>Reentry to isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation entry state, or isolation active state.</li> <li>Re-exit from isolation state. Write 1 into bit 0 of <code>IDM_ACCESS_CONTROL</code> register when already in pending isolation exit state.</li> </ul>	RO	0
[9]	active_write	Active write transactions. A 1 indicates there is at least one write transaction currently in progress.	RO	0
[8]	active_read	Active read transactions. A 1 indicates there is at least one read transaction currently in progress.	RO	0
[7:0]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0

16.16.43 PMNI idm\_access\_readid\_ns register

This register is the access log of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x188

Type

RO

Reset value

0x00000000

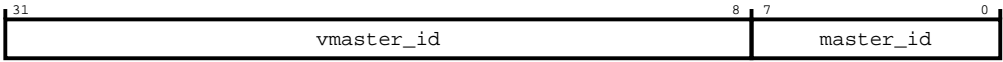
Constraints

None.

Bit descriptions

The following figure shows the idm\_access\_readid\_ns register bit assignments.

Figure 16-393: Bit assignment diagram for the idm\_access\_readid\_ns register



The following table shows the idm\_access\_readid\_ns register bit descriptions.

Table 16-409: idm\_access\_readid\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

### 16.16.44 PMNI idm\_access\_writeid\_ns register

This register is the access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x18C

##### Type

RO

##### Reset value

0x00000000

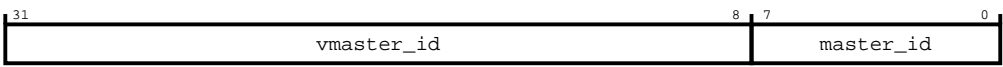
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_access\_writeid\_ns register bit assignments.

**Figure 16-394: Bit assignment diagram for the idm\_access\_writeid\_ns register**



The following table shows the idm\_access\_writeid\_ns register bit descriptions.

**Table 16-410: idm\_access\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the IDM_ACCESS_STATUS_NS register field active_write is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_write field of the IDM_ACCESS_STATUS_NS register is HIGH.	RO	0x0

16.16.45 PMNI idm\_reset\_status\_ns register

This register indicates the reset status of Non-secure transactions.

Configurations

This register is available in all configurations.

Attributes

Its characteristics are:

Width

32-bit

Address offset

0x194

Type

RO

Reset value

0x00000000

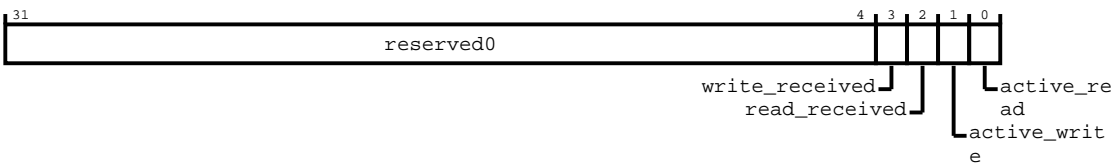
Constraints

None.

Bit descriptions

The following figure shows the idm\_reset\_status\_ns register bit assignments.

Figure 16-395: Bit assignment diagram for the idm\_reset\_status\_ns register



The following table shows the idm\_reset\_status\_ns register bit descriptions.

Table 16-411: idm\_reset\_status\_ns bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved, <b>UNDEFINED</b> , write as zero	RO	0x0
[3]	write_received	A 1 indicates that an active write transaction has occurred since the IDM entered the soft reset state. This bit is cleared to zero on: <ul style="list-style-type: none"><li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li><li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li></ul>	RO	0

Bits	Name	Description	Type	Reset
[2]	read_received	A 1 indicates that there has been an active read transaction since a write of 1 to the IDM_RESET_CONTROL register. This bit is cleared to 0 on: <ul style="list-style-type: none"> <li>Reentry to soft reset state. Write 1 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset entry state, or soft reset active state.</li> <li>Re-exit from soft reset state. Write 0 to bit[0] of the IDM_RESET_CONTROL register when already in pending soft reset exit state.</li> </ul>	RO	0
[1]	active_write	Active write transactions. A 1 indicates that there is at least one write transaction currently in progress.	RO	0
[0]	active_read	Active read transactions. A 1 indicates that there is at least one read transaction currently in progress.	RO	0

### 16.16.46 PMNI idm\_reset\_readid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x198

##### Type

RO

##### Reset value

0x00000000

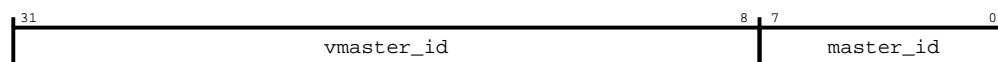
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_readid\_ns register bit assignments.

**Figure 16-396: Bit assignment diagram for the idm\_reset\_readid\_ns register**



The following table shows the idm\_reset\_readid\_ns register bit descriptions.

**Table 16-412: idm\_reset\_readid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the ARID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating ARID or HMASTER transaction. There is no manipulation of the incoming AXI ARID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when the active_read field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

### 16.16.47 PMNI idm\_reset\_writeid\_ns register

This register is the reset access log of Non-secure transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

Its characteristics are:

##### Width

32-bit

##### Address offset

0x19C

##### Type

RO

##### Reset value

0x00000000

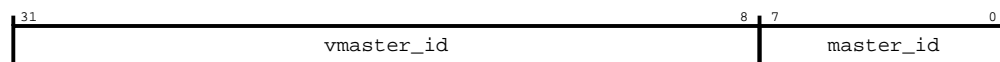
#### Constraints

None.

#### Bit descriptions

The following figure shows the idm\_reset\_writeid\_ns register bit assignments.

**Figure 16-397: Bit assignment diagram for the idm\_reset\_writeid\_ns register**



The following table shows the idm\_reset\_writeid\_ns register bit descriptions.

**Table 16-413: idm\_reset\_writeid\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:8]	vmaster_id	The incoming signal into the endpoint of the first transaction to arrive after isolation when the active_write field of the IDM_RESET_STATUS_NS register is HIGH. This field depends on the incoming endpoint. Therefore vmaster_id contains the AWID of the transaction on ASNI and contains the HMASTER on HSNI. For AMNI, PMNI, and HMNI the vmaster_id matches the ID of the originating AWID or HMASTER transaction. There is no manipulation of the incoming AXI AWID signal in ASNI.	RO	0x0
[7:0]	master_id	The originating Node ID of the ASNI or HSNI of the first transaction to arrive after isolation when active_write field of the IDM_RESET_STATUS_NS register is HIGH.	RO	0x0

## 16.16.48 PMNI idm\_interrupt\_status\_ns register

This register indicates the interrupt status of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1A8

#### Type

RW

#### Reset value

0x00000000

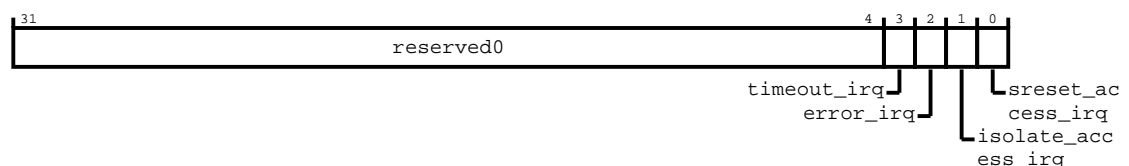
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_status\_ns register bit assignments.

**Figure 16-398: Bit assignment diagram for the idm\_interrupt\_status\_ns register**



The following table shows the idm\_interrupt\_status\_ns register bit descriptions.



**Table 16-414: idm\_interrupt\_status\_ns bit descriptions**

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved	RO	0x0
[3]	timeout_irq	Timeout detection event. Interface has detected a timeout.  Write 1 to clear.	RW	0
[2]	error_irq	Error detection event. Interface has detected a protocol error.  Write 1 to clear.	RW	0
[1]	isolate_access_irq	Isolation access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0
[0]	sreset_access_irq	Reset access event. Interface access while the IDM is closed.  Write 1 to clear.	RW	0

## 16.16.49 PMNI idm\_interrupt\_mask\_ns register

This register is the interrupt mask of Non-secure transactions.

### Configurations

This register is available in all configurations.

### Attributes

Its characteristics are:

#### Width

32-bit

#### Address offset

0x1AC

#### Type

RW

#### Reset value

0x00000000

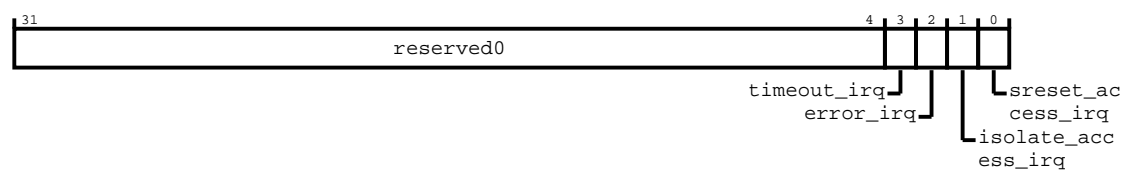
### Constraints

None.

### Bit descriptions

The following figure shows the idm\_interrupt\_mask\_ns register bit assignments.

Figure 16-399: Bit assignment diagram for the `idm_interrupt_mask_ns` register



The following table shows the `idm_interrupt_mask_ns` register bit descriptions.

Table 16-415: `idm_interrupt_mask_ns` bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	reserved0	Reserved	RO	0x0
[3]	timeout_irq	Timeout detection event mask	RW	0
[2]	error_irq	Error detection event mask	RW	0
[1]	isolate_access_irq	Isolation access event mask	RW	0
[0]	sreset_access_irq	Reset access event mask	RW	0

# Appendix A Signal descriptions

NI-710AE components provide a number of external signals.

Signal timing constraints and clock associations depend on the NI-710AE clock domain configuration. For more information, see [Signal timing constraints and clock associations](#).

The following signal groups are included in NI-710AE:

- [ASNI external interface types and associated signal groups](#)
- [AMNI external interface types and associated signal groups](#)
- [HSNI external interface types and associated signal groups](#)
- [HMNI external interface types and associated signal groups](#)
- [PMNI external interface types and associated signal groups](#)
- [Miscellaneous AXI interface signals](#)
- [Clock and reset signals](#)
- [Clock management signals](#)
- [Power management signals](#)
- [IDM interface signals](#)
- [Interrupt signals](#)
- [Configuration strap signals](#)
- [DFT interface signals](#)
- [Debug and Performance Monitoring Unit interface signals](#)
- [Fault Management Unit interface signals](#)
- [Access Protection Unit interface signals](#)



- Unless specified otherwise, signals are active-HIGH.
  - When making external device connections, unused signal bits must be tied LOW while unused check signal bits must be tied HIGH.
- 

## A.1 Signal timing constraints and clock associations

Every endpoint in NI-710AE is associated with a specific clock domain and its clock signal. Therefore, the timing constraints and clock associations of the signals on the endpoint interface depend on the NI-710AE clock domain configuration.

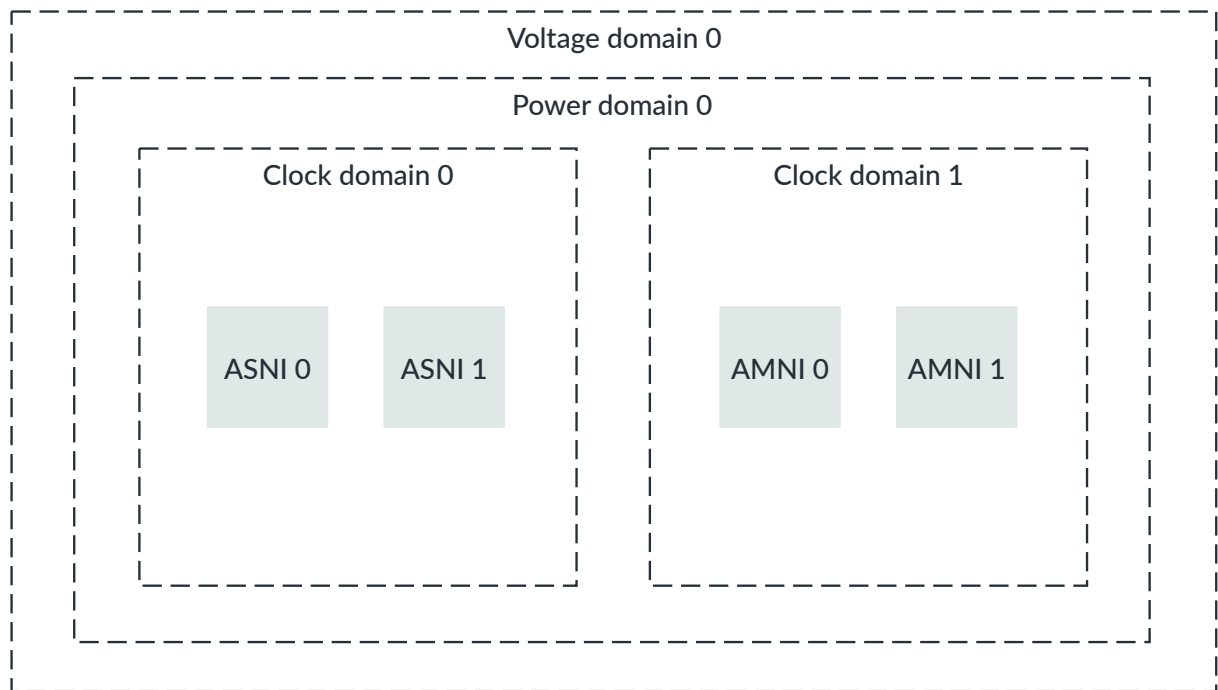
The voltage, power, and clock supply to NI-710AE is organized into voltage, power, and clock domains. You can configure the number of these domains according to your system requirements.

Every NI-710AE functional unit, for example, an endpoint, is contained by a single clock domain. Every clock domain is controlled by its own external Q-Channel and has its own clock signal. Therefore, depending on the clock domain they belong to, different functional units and their interfaces can have a different clock supply. As a result, the timing constraints and clock associations for a specific signal or interface is configuration-dependent. If you enable IDM on an endpoint, the same is true for the IDM-specific block-level signals.

The clock signal for a clock domain supplies all the functional units in the clock domain. Therefore, the signals for an interface are synchronous to the clock signal for the clock domain that the interface belongs to.

The following figure shows a simple example configuration with two clock domains and several endpoints.

**Figure A-1: Example domain hierarchy**



Clock domain 0 contains two ASNIs, ASNI 0 and ASNI 1, and clock domain 1 contains two AMNIs, AMNI 0 and AMNI 1. The ASNI 0 and ASNI 1 AXI completer interfaces and all associated signals are synchronous to the clock signal for clock domain 0. Similarly, the AMNI 0 and AMNI 1 AXI requester interfaces and all associated signals are synchronous to the clock signal for clock domain 1.

For more information about the voltage, power, and clock architecture of NI-710AE, see the *Power, clock, and reset management* chapter of the Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect Technical Reference Manual.

## A.2 ASNI external interface types and associated signal groups

You can configure an ASNI to have either an AXI5 or ACE5-Lite external completer interface. The ACE5-Lite interface has an extra set of signal groups compared to the AXI5 interface.



Check and Code signals are not present when `ambalInterfaceProtection` is disabled.

### AXI5 external interface signal groups

If your ASNI has an AXI5 interface, see the following sections to find the details of the AXI signals:

- [ASNI AXI4 write address channel signals](#)
- [ASNI AXI5 extension write address channel signals](#)
- [ASNI AXI4 write data channel signals](#)
- [ASNI AXI5 extension write data channel signals](#)
- [ASNI AXI4 write response channel signals](#)
- [ASNI AXI5 extension write response channel signals](#)
- [ASNI AXI4 read address channel signals](#)
- [ASNI AXI5 extension read address channel signals](#)
- [ASNI AXI4 read data channel signals](#)
- [ASNI AXI5 extension read data channel signals](#)
- [Other ASNI signals](#)

### ACE5-Lite external interface signal groups

If your ASNI has an ACE5-Lite interface, see the following sections to find the details of the AXI and ACE-Lite signals:

- [ASNI AXI4 write address channel signals](#)
- [ASNI AXI5 extension write address channel signals](#)
- [ASNI ACE-Lite write address channel signals](#)
- [ASNI ACE5-Lite extension write address channel signals](#)
- [ASNI AXI4 write data channel signals](#)
- [ASNI AXI5 extension write data channel signals](#)
- [ASNI AXI4 write response channel signals](#)
- [ASNI AXI5 extension write response channel signals](#)
- [ASNI ACE5-Lite extension write response channel signals](#)

- [ASNI AXI4 read address channel signals](#)
- [ASNI AXI5 extension read address channel signals](#)
- [ASNI ACE-Lite read address channel signals](#)
- [ASNI AXI4 read data channel signals](#)
- [ASNI AXI5 extension read data channel signals](#)
- [Other ASNI signals](#)

## Cortex-R52 and Cortex-R52+ external interface signal groups

If your ASNI has a Cortex-R52 or Cortex-R52+ bus interface, see the following sections to find the details of the signals:

- [ASNI Cortex-R52 and Cortex-R52+ AXIM interface signals](#)
- [ASNI LLPP interface signals](#)
- [ASNI Flash interface signals](#)

### A.2.1 ASNI AXI4 write address channel signals

All ASNI interface configurations contain a set of AXI4 write address channel signals. These signals transport AXI4 write address information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-1: ASNI AXI4 write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWID[n:0]	<prefix>_AWIDCHK	Input	Write address ID. Width is configurable.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWADDR[n:0]	<prefix>_AWADDRCHK	Input	Write address. Width is configurable from 32 bits to 64 bits.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWLEN[7:0]	<prefix>_AWLENCHK	Input	Write Burst length	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWSIZE[2:0]	<prefix>_AWCTLCHK0	Input	Write Burst size	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWBURST[1:0]	<prefix>_AWCTLCHK0	Input	Write Burst type	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWLOCK	<prefix>_AWCTLCHK0	Input	Write lock type	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWCACHE[3:0]	<prefix>_AWCTLCHK1	Input	Write cache type	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWPROT[2:0]	<prefix>_AWCTLCHK0	Input	Write protection type	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWQOS[3:0]	<prefix>_AWCTLCHK1	Input	Write (QoS) value	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWREGION[3:0]	<prefix>_AWCTLCHK1	Input	Write region identifier	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWUSER[n:0]	<prefix>_AWUSERCHK	Input	User-specified extension to AW payload	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWVALID	<prefix>_AWVALIDCHK	Input	Write address valid	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWREADY	<prefix>_AWREADYCHK	Output	Write address ready	Connect to the corresponding requester device, if populated.
<prefix>_AWNSAID[3:0]	<prefix>_AWNSAIDCHK	Input	NSAID signal associated with write address channel	

## A.2.2 ASNI AXI5 extension write address channel signals

All ASNI interface configurations contain a set of AXI5 extensions to the write address channel signals. These signals transport AXI5 write address information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-2: ASNI AXI5 extension write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWATOP	<prefix>_AWCTLCHK3	Input	AW atomic operation.  Indicates the type and endianness of atomic transactions.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWTRACE	<prefix>_AWTRACECHK	Input	Trace signals that are associated with the AW write address channel. AXI5 and ACE-Lite only.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWLOOP	<prefix>_AWLOOPCHK	Input	LOOP signal associated with the AW write address channel.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWMPAM	<prefix>_AWMPAMCHK	Input	Write address channel MPAM information.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWIDUNQ	<prefix>_AWIDCHK	Input	Write address channel unique ID indicator, active-HIGH.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWTAGOP	<prefix>_AWCTLCHK3	Input	Write request tag operation. Encoded as:  <b>00</b> Invalid  <b>01</b> Transfer  <b>10</b> Update  <b>11</b> Match	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

### A.2.3 ASNI ACE-Lite write address channel signals

ACE5-Lite ASNI interface configurations contain a set of ACE-Lite write address channel signals. These signals transport ACE-Lite write address information between an upstream ACE-Lite device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-3: ASNI ACE-Lite write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWSNOOP[3:0]	<prefix>_AWCTLCHK2	Input	Transaction type for shareable write transactions	Connect to the corresponding requester device, if populated.
<prefix>_AWDOMAIN[1:0]	<prefix>_AWCTLCHK2	Input	Indicates the Shareability domain of a write transaction	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

### A.2.4 ASNI ACE5-Lite extension write address channel signals

ACE5-Lite ASNI interface configurations contain a set of ACE5-Lite extensions to the write address channel signals. These signals transport ACE5-Lite write address information between an upstream ACE-Lite device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.



## Signal definitions

**Table A-4: ASNI ACE5-Lite extension write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWSTASHNID	<prefix>_AWSTASHNIDCHK	Input	Indicates the node identifier of the physical interface. This interface is the target interface for the cache stashing operation.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWSTASHNIDEN	<prefix>_AWSTASHNIDCHK	Input	When asserted, this signal indicates the AWSTASHNID signal is valid and must be used.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWSTASHLPID	<prefix>_AWSTASHLPIDCHK	Input	Indicates the logical processor subunit associated with the physical interface that is the target for the cache stashing operation.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWSTASHLPIDEN	<prefix>_AWSTASHLPIDENCHK	Input	When asserted, this signal indicates the AWSTASHLPID signal is enabled and must be used.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_AWCMO	<prefix>_AWCTLCHK3	Input	Indicates the type of CMO.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.2.5 ASNI AXI4 write data channel signals

All ASNI interface configurations contain a set of AXI4 write data channel signals. These signals transport AXI4 write data information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

## Signal definitions

**Table A-5: ASNI AXI4 write data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_WDATA[DATA_WIDTH-1:0]	<prefix>_WDATACHK	Input	Write data	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_WSTRB[(DATA_WIDTH/8)-1:0]	<prefix>_WSTRBCHK	Input	Write byte lane strobes	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_WLAST	<prefix>_WLASTCHK	Input	Write data last transfer indication	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_WUSER[n:0]	<prefix>_WUSERCHK	Input	User-specified extension to W payload	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_WVALID	<prefix>_WVALIDCHK	Input	Write data valid	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_WREADY	<prefix>_WREADYCHK	Output	Write data ready	Connect to the corresponding requester device, if populated.

## A.2.6 ASNI AXI5 extension write data channel signals

All ASNI interface configurations contain a set of AXI5 extensions to the write data channel signals. These signals transport AXI5 write data information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-6: ASNI AXI5 extension write data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_WTRACE	<prefix>_WTRACECHK	Input	Trace signals that are associated with the write data channel	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_WTAG	<prefix>_WTAGCHK	Input	<p>The tag associated with write data.</p> <p>There is a 4-bit tag for each 128 bits of data, with a minimum of 4 bits.</p> <p><math>WTAG[(((4 \times n)-1):4 \times (n-1))]</math> corresponds to <math>WDATA[(((128 \times n)-1):128 \times (n-1))]</math></p> <p><b>Note:</b> WTAG has the same validity rules as WDATA.</p>	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_WTAGUPDATE	<prefix>_WTAGUPDATECHK	Input	<p>Indicates which tags must be written to memory when an Update operation occurs.</p> <ul style="list-style-type: none"> <li>If a bit is asserted, then the corresponding tags must be written to memory.</li> <li>If a bit is deasserted, then the corresponding tags are invalid.</li> </ul> <p>There is 1 bit for each 4 bits of tag. WTAGUPDATE[n] corresponds to WTAG[(4n)+3:(4n)] .</p> <p>WTAGUPDATE bits outside of the transaction container must be deasserted.</p> <p>For operations other than Update, WTAGUPDATE must be deasserted. It can be asserted or deasserted for Update operations.</p>	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.2.7 ASNI AXI4 write response channel signals

All ASNI interface configurations contain a set of AXI4 write response channel signals. These signals transport AXI4 write response information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

The following channel signal widths are defined as:

- BID[n:0] - user configurable
- BIDCHK[n:0] - use the formula
- BUSER[n:0] - user configurable
- BUSERCHK[n:0] - use the formula

### Signal definitions

**Table A-7: ASNI AXI4 write response channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_BID[n:0]	<prefix>_BIDCHK	Output	Write response ID, width is configurable	Connect to the corresponding requester device, if populated.
<prefix>_BRESP[1:0]	<prefix>_BRESPCHK	Output	Write response	Connect to the corresponding requester device, if populated.
<prefix>_BUSER[n:0]	<prefix>_BUSERCHK	Output	User-specified extension to write response payload	Connect to the corresponding requester device, if populated.

Signal	Check signal	Direction	Description	Connection information
<prefix>_BVALID	<prefix>_BVALIDCHK	Output	Write response valid	Connect to the corresponding requester device, if populated.
<prefix>_BREADY	<prefix>_BREADYCHK	Input	Write response ready	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.2.8 ASNI AXI5 extension write response channel signals

All ASNI interface configurations contain a set of AXI5 extensions to the write response channel signals. These signals transport AXI5 write response information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-8: ASNI AXI5 extension write response channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_BTRACE	<prefix>_BTRACECHK	Output	Trace signals that are associated with the write response channel	Connect to the corresponding requester device, if populated.
<prefix>_BLOOP	<prefix>_BLOOPCHK	Output	LOOP signal associated with the write response channel	Connect to the corresponding requester device, if populated.
<prefix>_BIDUNQ	<prefix>_BIDCHK	Output	Write response channel unique ID indicator, active-HIGH	Connect to the corresponding requester device, if populated.
<prefix>_BTAGMATCH	<prefix>_BRESPCHK	Output	Indicates the result of a tag comparison on a write transaction:  <b>00</b> Not a match transaction  <b>01</b> No match result  <b>10</b> Fail  <b>11</b> Pass	Connect to the corresponding requester device, if populated.

## A.2.9 ASNI ACE5-Lite extension write response channel signals

ACE5-Lite ASNI interface configurations contain a set of ACE5-Lite extensions to the write response channel signals. These signals transport ACE5-Lite write response information between an upstream ACE5-Lite device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

## Signal definitions

**Table A-9: ASNI ACE5-Lite extension write response channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_BPERSIST	<prefix>_BRESPCHK	Output	Indicates that the write data is updated in persistent memory. Can only be asserted for transactions where AWCMO is CleanSharedPersist or CleanSharedDeepPersist.	Connect to the corresponding requester device, if populated.

## A.2.10 ASNI AXI4 read address channel signals

All ASNI interface configurations contain a set of AXI4 read address channel signals. These signals transport AXI4 read address information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

## Signal definitions

**Table A-10: ASNI AXI4 read address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARID[n:0]	<prefix>_ARIDCHK	Input	Read data ID. Width is configurable.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARADDR	<prefix>_ARADDRCHK	Input	Address of the first transfer in a read transaction	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARLEN	<prefix>_ARLENCHK	Input	Length. The exact number of data transfers in a read transaction.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARSIZE	<prefix>_ARCTLCHK0	Input	Size. The number of bytes in each data transfer in a read transaction.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARBURST	<prefix>_ARCTLCHK0	Input	Burst type. Indicates how address changes between each transfer in a read transaction.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARLOCK	<prefix>_ARCTLCHK0	Input	Information about the atomic characteristics of a read transaction	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARCACHE	<prefix>_ARCTLCHK1	Input	Indicates how a read transaction is required to progress through a system	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARPROT	<prefix>_ARCTLCHK0	Input	Protection attributes of a read transaction: privilege, security level, and access type	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARQOS	<prefix>_ARCTLCHK1	Input	QoS identifier for a read transaction	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>[3:0]	<prefix>_ARCTLCHK1	Input	Read region identifier	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARUSER	<prefix>_ARUSERCHK	Input	User-defined extension for the read address channel	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARVALID	<prefix>_ARVALIDCHK	Input	Indicates that the read address channel signals are valid	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARREADY	<prefix>_ARREADYCHK	Output	Indicates that a transfer on the read address channel can be accepted	Connect to the corresponding requester device, if populated.
<prefix>_ARNSAID	<prefix>_ARNSAIDCHK	Input	NSAID associated with the read address channel	Connect to the corresponding requester device, if populated.

## A.2.11 ASNI AXI5 extension read address channel signals

All ASNI interface configurations contain a set of AXI5 extensions to the read address channel signals. These signals transport AXI5 read address information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-11: ASNI AXI5 extension read address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARTRACE	<prefix>_ARTRACECHK	Input	Trace signal that is associated with the AR read address channel	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARLOOP	<prefix>_ARLOOPCHK	Input	LOOP signal associated with the AR read address channel	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARMPAM	<prefix>_ARMPAMCHK	Input	Read address channel MPAM information	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARIDUNQ	<prefix>_ARIDCHK	Input	Read address channel unique ID indicator, active-HIGH	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_ARCHUNKEN	<prefix>_ARCTLCHK3	Input	If this signal is asserted, read data for this transaction can be returned out of order, in 128-bit chunks.	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARTAGOP	<prefix>_ARCTLCHK3	Input	Read request tag operation. Encoded as:  <b>0b00</b> Invalid  <b>0b01</b> Transfer  <b>0b10</b> Reserved  <b>0b11</b> Fetch	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.2.12 ASNI ACE-Lite read address channel signals

ACE5-Lite ASNI interface configurations contain a set of ACE-Lite read address channel signals. These signals transport ACE-Lite read address information between an upstream ACE5-Lite device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-12: ASNI ACE-Lite read address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARSNOOP[3:0]	<prefix>_ARCTLCHK2	Input	Transaction type for shareable read transactions	Connect to the corresponding requester device, if populated
<prefix>_ARDOMAIN[1:0]	<prefix>_ARCTLCHK2	Input	Shareability domain of a read transaction	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.2.13 ASNI AXI4 read data channel signals

All ASNI interface configurations contain a set of AXI4 read data channel signals. These signals transport AXI read data information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-13: ASNI AXI4 read data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_RID[n:0]	<prefix>_RIDCHK	Output	Read data ID, width is configurable	Connect to the corresponding requester device, if populated.
<prefix>_RDATA[DATA_WIDTH-1:0]	<prefix>_RDATACHK	Output	Read data	Connect to the corresponding requester device, if populated.
<prefix>_RRESP[3:0]	<prefix>_RRESPCHK	Output	Read data response	Connect to the corresponding requester device, if populated.

Signal	Check signal	Direction	Description	Connection information
<prefix>_RLAST	<prefix>_RLASTCHK	Output	Read data last transfer indication	Connect to the corresponding requester device, if populated.
<prefix>_RUSER[n:0]	<prefix>_RUSERCHK	Output	User-specified extension to read data payload	Connect to the corresponding requester device, if populated.
<prefix>_RVALID	<prefix>_RVALIDCHK	Output	Read data valid	Connect to the corresponding requester device, if populated.
<prefix>_RREADY	<prefix>_RREADYCHK	Input	Read data ready	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.2.14 ASNI AXI5 extension read data channel signals

All ASNI interface configurations contain a set of AXI5 extensions to the read data channel signals. These signals transport AXI5 read data information between an upstream AXI device and the downstream ASNI.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-14: ASNI AXI5 extension read data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_RTRACE	<prefix>_RTRACECHK	Output	Trace signal that is associated with the read data channel	Connect to the corresponding requester device, if populated.
<prefix>_RLOOP	<prefix>_RLOOPCHK	Output	LOOP signal associated with the read data channel	Connect to the corresponding requester device, if populated.
<prefix>_RIDUNQ	<prefix>_RIDUNQ	Output	Read data channel unique ID indicator, active-HIGH	Connect to the corresponding requester device, if populated.
<prefix>_RCHUNKV	<prefix>_RCHUNKCHK	Output	If this signal is asserted, RCHUNKNUM and RCHUNKSTRB are valid for this transfer.	Connect to the corresponding requester device, if populated.
<prefix>_RCHUNKNUM	<prefix>_RCHUNKCHK	Output	Indicates the chunk number being transferred. Chunks are numbered incrementally from zero, according to the data width and base address of the transaction.	Connect to the corresponding requester device, if populated.



Signal	Check signal	Direction	Description	Connection information
<prefix>_RCHUNKSTRB	<prefix>_RCHUNKCHK	Output	Indicates which part of read data is valid for this transfer, each bit corresponds to 128 bits of data. For example: <ul style="list-style-type: none"> <li>RCHUNKSTRB[0] corresponds to RDATA[127:0]</li> <li>RCHUNKSTRB[1] corresponds to RDATA[255:128]</li> </ul>	Connect to the corresponding requester device, if populated.
<prefix>_RTAG	<prefix>_RTAGCHK	Output	The tag associated with read data.  There is a 4-bit tag for each 128 bits of data, with a minimum of 4 bits.  RTAG[ $((4 \times n) - 1) : 4 \times (n - 1)$ ] corresponds to RDATA[ $((128 \times n) - 1) : 128 \times (n - 1)$ ]  <b>Note:</b> RTAG has the same validity rules as RDATA.	Connect to the corresponding requester device, if populated.

## A.2.15 Other ASNI signals

ASNI configurations have a set of interface signals that are not related to a specific AXI channel.

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-15: Other ASNI signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_QOSOVERRIDE	<prefix>_QOSOVERRIDECHK	Input	Sample at reset QoS override. For more information, see the <a href="#">QoS value override programmable registers</a> .	To enable QoS override, tie HIGH. Otherwise tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_ORDERED_WRITE_OBSERVATION	<prefix>_ORDERED_WRITE_OBSERVATIONCHK	Input	Enables OWO on this completer interface if asserted. Refer to the OWO feature in the <i>AMBA® AXI and ACE Protocol Specification</i> . The CDAS has a subsection on OWO. For more information, see <a href="#">Ordered Write Observation</a> .	To enable OWO, tie HIGH. Otherwise tie LOW.

## A.2.16 ASNI Cortex-R52 and Cortex-R52+ AXIM interface signals

The ASNI AXIM interface connects to the AXIM interface of a Cortex-R52 or Cortex-R52+ processor.

The ASNI AXIM interface contains the following signal groups:

- [ASNI AXIM read address channel signals](#)
- [ASNI AXIM read data channel signals](#)
- [ASNI AXIM write address channel signals](#)
- [ASNI AXIM write data channel signals](#)
- [ASNI AXIM write response channel signals](#)

In this section, <prefix> represents <PROTOCOL>\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

When BUSPROTECTION3 parameter is selected, CODE signals will be present only on read data and write data channels.

### A.2.16.1 ASNI AXIM read address channel signals

The ASNI AXIM interface read address channel signals transport read address information between NI-710AE and the AXIM interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read address channel signal for the ASNI AXIM interface.

**Table A-16: ASNI AXIM read address channel signal**

Signal	Code Signal	Direction	Description	Connection information
<prefix>_ARREADY	<prefix>_ARREADYCODE	Output	Read address ready.	Connect to the corresponding requester device, if populated.
<prefix>_ARADDR[31:0]	<prefix>_ARADDRCODE	Input	Read address.	Connect to the corresponding requester device, if populated.
<prefix>_ARBURST[1:0]	<prefix>_ARCTL1CODE	Input	Read burst type.	Connect to the corresponding requester device, if populated.
<prefix>_ARCACHE[3:0]	<prefix>_ARCTL0CODE	Output	Read cache type.	Connect to the corresponding requester device, if populated.
<prefix>_ARID[3:0]	<prefix>_ARIDCODE	Input	Read request ID.	Connect to the corresponding requester device, if populated.
<prefix>_ARLEN[7:0]	<prefix>_ARCTL1CODE	Input	Read burst length.	Connect to the corresponding requester device, if populated.
<prefix>_ARLOCK	<prefix>_ARCTL1CODE	Input	Read lock type.	Connect to the corresponding requester device, if populated.
<prefix>_ARPROT[2:0]	<prefix>_ARCTL0CODE	Input	Read protection type.	Connect to the corresponding requester device, if populated.

Signal	Code Signal	Direction	Description	Connection information
<prefix>_ARVALID	<prefix>_ARVALIDCODE	Input	Read address valid.	Connect to the corresponding requester device, if populated.
<prefix>_ARSIZE[2:0]	<prefix>_ARCTL1CODE	Input	Read burst size.	Connect to the corresponding requester device, if populated.
<prefix>_ARQOS[3:0]	<prefix>_ARCTL0CODE	Input	Quality of service read identifier.	Connect to the corresponding requester device, if populated.
<prefix>_ARVMID[n:0]	<prefix>_ARVMIDCODE	Input	<p>EL2 Exception level (&lt;prefix&gt;_ARVMID[n]) and Virtual Machine Identifier (&lt;prefix&gt;_ARVMID[n-1:0]), where uniquely identifiable. For example, for Device and Normal Non-cacheable memory transfers, the Virtual Machine Identifier (VMID) is indicated. When a transfer does not have a uniquely identifiable VMID, the signal is driven as 0x000.</p> <p><b>Note:</b> n = 7 for Cortex-R52  n = 8 for Cortex-R52+</p>	Connect to the corresponding requester device, if populated.

### A.2.16.2 ASNI AXIM read data channel signals

The ASNI AXIM interface read data channel signals transport read data information between NI-710AE and the AXIM interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read data channel signal for the ASNI AXIM interface.

**Table A-17: ASNI AXIM read data channel signal**

Signal	Code Signal	Direction	Description	Connection information
<prefix>_RDATA[127:0]	<prefix>_RDATACODE	Output	Read data.	Connect to the corresponding requester device, if populated.
<prefix>_RID[3:0]	<prefix>_RIDCODE	Output	Read data ID.	Connect to the corresponding requester device, if populated.
<prefix>_RLAST	<prefix>_RCTLCODE	Output	Read last.	Connect to the corresponding requester device, if populated.
<prefix>_RRESP[1:0]	<prefix>_RCTLCODE	Output	Read data response.	Connect to the corresponding requester device, if populated.

Signal	Code Signal	Direction	Description	Connection information
<prefix>_RVALID	<prefix>_RVALIDCODE	Output	Read data valid.	Connect to the corresponding requester device, if populated.
<prefix>_RREADY	<prefix>_RREADYCODE	Input	Read data ready.	Connect to the corresponding requester device, if populated.

### A.2.16.3 ASNI AXIM write address channel signals

The ASNI AXIM interface write address channel signals transport write address information between NI-710AE and the AXIM interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write address channel signals for the ASNI AXIM interface.

**Table A-18: ASNI AXIM write address channel signals**

Signal	Code Signal	Direction	Description	Connection information
<prefix>_AWREADY	<prefix>_AWREADYCODE	Output	Write address ready.	Connect to the corresponding requester device, if populated.
<prefix>_AWADDR[31:0]	<prefix>_AWADDRCODE	Input	Write address.	Connect to the corresponding requester device, if populated.
<prefix>_AWBURST[1:0]	<prefix>_AWCTLCODE1	Input	Write burst type.	Connect to the corresponding requester device, if populated.
<prefix>_AWCACHE[3:0]	<prefix>_AWCTLCODE0	Input	Write memory type.	Connect to the corresponding requester device, if populated.
<prefix>_AWID[2:0]	<prefix>_AWIDCODE	Input	Write request ID.	Connect to the corresponding requester device, if populated.

Signal	Code Signal	Direction	Description	Connection information
<prefix>_AWLEN[7:0]	<prefix>_AWCTLCODE1	Input	Write burst length. <prefix>_AWLEN[7:2] is always 0b000000.	Connect to the corresponding requester device, if populated.
<prefix>_AWLOCK	<prefix>_AWCTLCODE1	Input	Write lock type.	Connect to the corresponding requester device, if populated.
<prefix>_AWPROT[2:0]	<prefix>_AWCTLCODE0	Input	Write protection type.	Connect to the corresponding requester device, if populated.
<prefix>_AWQOS[3:0]	<prefix>_AWCTLCODE0	Input	Quality of service write identifier.	Connect to the corresponding requester device, if populated.
<prefix>_AWSIZE[2:0]	<prefix>_AWCTLCODE1	Input	Write burst size.	Connect to the corresponding requester device, if populated.
<prefix>_AWVALID	<prefix>_AWVALIDCODE	Input	Write address valid.	Connect to the corresponding requester device, if populated.
<prefix>_AWVMID[n:0]	<prefix>_AWVMIDCODE	Input	<p>EL2 Exception level (&lt;prefix&gt;_AWVMID[n]) and VMID (&lt;prefix&gt;_AWVMID[n-1:0]), where uniquely identifiable. For example, for Device and Normal Non-cacheable memory transfers, the VMID is indicated. When a transfer does not have a uniquely identifiable VMID, the signal is driven as 0x00.</p> <p><b>Note:</b> n = 7 for Cortex-R52  n = 8 for Cortex-R52+</p>	Connect to the corresponding requester device, if populated.

#### A.2.16.4 ASNI AXIM write data channel signals

The ASNI AXIM interface write data channel signals transport write data information between NI-710AE and the AXIM interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write data channel signals for the ASNI AXIM interface.

**Table A-19: ASNI AXIM write data channel signals**

Signal	Code Signal	Direction	Description	Connection information
<prefix>_WREADY	<prefix>_WREADYCODE	Output	Write data ready.	Connect to the corresponding requester device, if populated.
<prefix>_WDATA[127:0]	<prefix>_WDATACODE	Input	Write data.	Connect to the corresponding requester device, if populated.
<prefix>_WSTRB[15:0]	<prefix>_WCTLCODE	Input	Write byte-lane strobes.	Connect to the corresponding requester device, if populated.
<prefix>_WLAST	<prefix>_WCTLCODE	Input	Indicates the last transfer in a write burst.	Connect to the corresponding requester device, if populated.
<prefix>_WVALID	<prefix>_WVALIDCODE	Input	Write data valid.	Connect to the corresponding requester device, if populated.

#### A.2.16.5 ASNI AXIM write response channel signals

The ASNI AXIM interface write response channel signals transport write response information between NI-710AE and the AXIM interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write response channel signals for the ASNI AXIM interface.

**Table A-20: ASNI AXIM write response channel signals**

Signal	Code Signal	Direction	Description	Connection information
<prefix>_BID[2:0]	<prefix>_BIDCODE	Output	Write response ID.	Connect to the corresponding requester device, if populated.
<prefix>_BRESP[1:0]	<prefix>_BCTLCODE	Output	Write response.	Connect to the corresponding requester device, if populated.
<prefix>_BVALID	<prefix>_BVALIDCODE	Output	Write response valid.	Connect to the corresponding requester device, if populated.
<prefix>_BREADY	<prefix>_BREADYCODE	Input	Write response ready.	Connect to the corresponding requester device, if populated.

### A.2.17 ASNI LLPP interface signals

The ASNI LLPP interface connects to the LLPP interface of a Cortex-R52 or Cortex-R52+ processor.

The ASNI LLPP interface contains the following signal groups:

- [ASNI LLPP read address channel signals](#)

- [ASNI LLPP read data channel signals](#)
- [ASNI LLPP write address channel signals](#)
- [ASNI LLPP write data channel signals](#)
- [ASNI LLPP write response channel signals](#)

### A.2.17.1 ASNI LLPP read address channel signals

The ASNI LLPP interface read address channel signals transport read address information between NI-710AE and the LLPP interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read address channel signals for the ASNI LLPP interface.

**Table A-21: ASNI LLPP read address channel signals**

Signal	Code signal	Direction	Description	Connection information
<prefix>_ARREADYPx	ARREADYCODEPx	Output	Read address ready.	Connect to the corresponding requester device, if populated.
ARADDRPx[31:0]	ARADDRCODEPx	Input	Read address.	Connect to the corresponding requester device, if populated.
ARBURSTPx[1:0]	ARCTL1CODEPx	Input	Read burst type.	Connect to the corresponding requester device, if populated.
ARCACHEPx[3:0]	ARCTL0CODEPx	Input	Read cache type.	Connect to the corresponding requester device, if populated.
ARIDPx	ARIDCODEPx	Input	Read address ID.	Connect to the corresponding requester device, if populated.
ARLENPx[7:0]	ARCTL1CODEPx	Input	Read burst length.	Connect to the corresponding requester device, if populated.
ARLOCKPx	ARCTL1CODEPx	Input	Read lock type.	Connect to the corresponding requester device, if populated.
ARSIZEPx[2:0]	ARCTL1CODEPx	Input	Read burst size.	Connect to the corresponding requester device, if populated.
ARVALIDPx	ARVALIDCODEPx	Input	Read address valid.	Connect to the corresponding requester device, if populated.
ARPROTPx[2:0]	ARCTL0CODEPx	Input	Read protection type.	Connect to the corresponding requester device, if populated.
ARQOSPx[3:0]	ARCTL0CODEPx	Input	Quality of service read identifier.	Connect to the corresponding requester device, if populated.



Signal	Code signal	Direction	Description	Connection information
ARVMIDPx[n:0]	ARVMIDCODEPx	Input	<p>EL2 Exception level (ARVMIDPx[n]) and VMID (ARVMIDPx[n-1:0]). If the access was performed in EL2, bit[8] is set HIGH.</p> <p><b>Note:</b> n = 7 for Cortex-R52  n = 8 for Cortex-R52+</p>	Connect to the corresponding requester device, if populated.

### A.2.17.2 ASNI LLPP read data channel signals

The ASNI LLPP interface read data channel signals transport read data information between NI-710AE and the LLPP interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read data channel signals for the ASNI LLPP interface.

**Table A-22: ASNI LLPP read data channel signals**

Signal	Code Signal	Direction	Description	Connection information
RREADYPx	RREADYCODEPx	Input	Read data ready.	Connect to the corresponding requester device, if populated.
RLASTPx	RCTLCODEPx	Output	Read data last transfer indication.	Connect to the corresponding requester device, if populated.
RRESPPx[1:0]	RCTLCODEPx	Output	Read response.	Connect to the corresponding requester device, if populated.
RVALIDPx	RVALIDCODEPx	Output	Read data valid.	Connect to the corresponding requester device, if populated.
RDATAPx[31:0]	RDATACODEPx	Output	Read data.	Connect to the corresponding requester device, if populated.
RIDPx[n:0]	RIDCODEPx	Output	<p>Read data ID.</p> <p><b>Note:</b> n is configurable for Cortex-R52.  n is fixed at 15 for Cortex-R52+.</p>	Connect to the corresponding requester device, if populated.

### A.2.17.3 ASNI LLPP write address channel signals

The ASNI LLPP interface write address channel signals transport write address information between NI-710AE and the LLPP interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write address channel signal for the ASNI LLPP interface.

**Table A-23: ASNI LLPP write address channel signal**

Signal	Code signal	Direction	Description	Connection information
AWREADYPx	AWREADYCODEPx	Output	Write address ready.	
AWBURSTPx[1:0]	AWCTL1CODEPx	Input	Write burst type.	
AWCACHEPx[3:0]	AWCTL0CODEPx	Input	Write cache type.	
AWIDPx	AWIDCODEPx	Input	Write address ID.	
AWLENPx[7:0]	AWCTL1CODEPx	Input	Write burst length.	
AWLOCKPx	AWCTL1CODEPx	Input	Write lock type.	
AWPROTPx[2:0]	AWCTL0CODEPx	Input	Write protection type.	
AWVALIDPx	AWVALIDCODEPx	Input	Write address valid.	
AWSIZEPx[2:0]	AWCTL1CODEPx	Input	Write burst size.	
AWADDRPx[31:0]	AWADDRCODEPx	Input	Write address.	
AWQOSPx[3:0]	AWCTL0CODEPx	Input	Quality of service write identifier.	
AWVMIDPx[n:0]	AWVMIDCODEPx	Input	<p>EL2 Exception level (AWVMIDPx[n]) and VMID (AWVMIDPx[n-1:0]). On LLPP, the VMID field is always uniquely identifiable. If the access was performed in EL2, bit[8] is set HIGH.</p> <p><b>Note:</b> n = 7 for Cortex-R52  n = 8 for Cortex-R52+</p>	

#### A.2.17.4 ASNI LLPP write data channel signals

The ASNI LLPP interface write data channel signals transport write data information between NI-710AE and the LLPP interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write data channel signal for the ASNI LLPP interface.

**Table A-24: ASNI LLPP write data channel signal**

Signal	Code Signal	Direction	Description	Connection information
WREADYPx	WREADYCODEPx	Output	Write data Ready.	Connect to the corresponding requester device, if populated.
WVALIDPx	WVALIDCODEPx	Input	Write data valid.	Connect to the corresponding requester device, if populated.
WDATAPx[31:0]	WDATACODEPx	Input	Write data.	Connect to the corresponding requester device, if populated.
WLASTPx	WCTLCODEPx	Input	Write data last transfer indication.	Connect to the corresponding requester device, if populated.
WSTRBPx[3:0]	WCTLCODEPx	Input	Write response.	Connect to the corresponding requester device, if populated.

### A.2.17.5 ASNI LLPP write response channel signals

The ASNI LLPP interface write response channel signals transport write response information between NI-710AE and the LLPP interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write response channel signals for the ASNI LLPP interface.

**Table A-25: ASNI LLPP write response channel signals**

Signal	Code Signal	Direction	Description	Connection information
BIDPx	BIDCODEPx	Output	Write response ID.	Connect to the corresponding requester device, if populated.
BREADYPx	BREADYCODEPx	Input	Write response ready.	Connect to the corresponding requester device, if populated.
BRESPPx[1:0]	BCTLCODEPx	Output	Write response.	Connect to the corresponding requester device, if populated.
BVALIDPx	BVALIDCODEPx	Output	Write response valid.	Connect to the corresponding requester device, if populated.

## A.2.18 ASNI Flash interface signals

The ASNI flash interface connects to the flash interface of a Cortex-R52 or Cortex-R52+ processor.

The ASNI flash interface contains the following signal groups:

- [ASNI Flash read address channel signals](#)
- [ASNI Flash read data channel signals](#)

### A.2.18.1 ASNI Flash read address channel signals

The ASNI flash interface read address channel signals transport read address information between NI-710AE and the flash interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read address channel signals for the ASNI flash interface.

**Table A-26: ASNI Flash read address channel signals**

Signal	Code signal	Direction	Description	Connection information
ARREADYFx	ARREADYCODEFx	Output	Read address ready.	Connect to the corresponding requester device, if populated.
ARLENFx[7:0]	ARCTL1CODEFx	Input	Read burst length.	Connect to the corresponding requester device, if populated.
ARIDFx	ARIDCODEFx	Input	Read address ID.	Connect to the corresponding requester device, if populated.
ARADDRFx[31:0]	ARADDRCODEFx	Input	Read address.	Connect to the corresponding requester device, if populated.
ARPROTFx[2:0]	ARCTL0CODEFx	Input	Read protection type.	Connect to the corresponding requester device, if populated.
ARVALIDFx	ARVALIDCODEFx	Input	Read address valid.	Connect to the corresponding requester device, if populated.
ARBURSTFx[1:0]	ARCTL1CODEFx	Input	Read burst type.	Connect to the corresponding requester device, if populated.

### A.2.18.2 ASNI Flash read data channel signals

The ASNI flash interface read data channel signals transport read data information between NI-710AE and the flash interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read data channel signal for the ASNI flash interface.

**Table A-27: ASNI Flash read data channel signal**

Signal	Code Signal	Direction	Description	Connection information
RIDFx	RIDCODEFx	Output	Read data ID.	Connect to the corresponding requester device, if populated.
RDATAFx[127:0]	RDATACODEFx	Output	Read data.	Connect to the corresponding requester device, if populated.
RRESPFx[1:0]	RCTLCODEFx	Output	Read data response.	Connect to the corresponding requester device, if populated.
RVALIDFx	RVALIDCODEFx	Output	Read data valid.	Connect to the corresponding requester device, if populated.
RLASTFx	RCTLCODEFx	Output	Indicates the last transfer in a read burst.	Connect to the corresponding requester device, if populated.
RREADYFx	RREADYCODEFx	Input	Read data ready.	Connect to the corresponding requester device, if populated.

## A.3 AMNI external interface types and associated signal groups

You can configure an AMNI to have an AXI5, AXI3, ACE5-Lite, or ACE5-LiteACP external requester interface. Each interface type has a different set of AXI signal groups.



Note

- Check and Code signals are not present when ambalInterfaceProtection is disabled.
- When BUSPROTECTION3 parameter is selected, Code signals are present only for Read data and Write data channels.

### AXI5 external interface signal groups

If your AMNI has an AXI5 interface, see the following sections to find the details of the AXI signals:

- [AMNI AXI4 write address channel signals](#)
- [AMNI AXI5 extension write address channel signals](#)
- [AMNI AXI4 write data channel signals](#)
- [AMNI AXI5 extension write data channel signals](#)
- [AMNI AXI4 write response channel signals](#)
- [AMNI AXI5 extension write response channel signals](#)

- [AMNI AXI4 read address channel signals](#)
- [AMNI AXI5 extension read address channel signals](#)
- [AMNI AXI4 read data channel signals](#)
- [AMNI AXI5 extension read data channel signals](#)

### **AXI3 external interface signal groups**

If your AMNI has an AXI3 interface, some of the signals that are also present in the AXI5 configuration have different widths. For information about these changes, see [AMNI AXI3 interface configuration signal changes](#). See the following sections to find the details of the other AXI signals:

- [AMNI AXI4 write address channel signals](#)
- [AMNI AXI4 write data channel signals](#)
- [AMNI AXI4 write response channel signals](#)
- [AMNI AXI4 read address channel signals](#)
- [AMNI AXI4 read data channel signals](#)

### **ACE5-Lite and ACE5-LiteACP external interface signal groups**

If your AMNI has an ACE5-Lite or ACE5-LiteACP interface, see the following sections to find the details of the AXI and ACE-Lite signals:

- [AMNI AXI4 write address channel signals](#)
- [AMNI AXI5 extension write address channel signals](#)
- [AMNI ACE-Lite write address channel signals](#)
- [AMNI ACE5-Lite extension write address channel signals](#)
- [AMNI AXI4 write data channel signals](#)
- [AMNI AXI5 extension write data channel signals](#)
- [AMNI AXI4 write response channel signals](#)
- [AMNI AXI5 extension write response channel signals](#)
- [AMNI AXI4 read address channel signals](#)
- [AMNI AXI5 extension read address channel signals](#)
- [AMNI ACE-Lite read address channel signals](#)
- [AMNI AXI4 read data channel signals](#)
- [AMNI AXI5 extension read data channel signals](#)

### **Cortex-R52 and Cortex-R52+ external interface signal groups**

If your AMNI has an Cortex-R52 or Cortex-R52+ bus interface, see the following sections to find the details of the signals:

- [AMNI AXI5 read address channel signals](#)

### A.3.1 AMNI AXI4 write address channel signals

All AMNI interface configurations contain a set of AXI4 write address channel signals. These signals transport AXI4 write address information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-28: AMNI AXI4 write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWID[n:0]	<prefix>_AWIDCHK  Signal width is $\text{ceil}((\text{ID\_W\_WIDTH}+1)/8)$ .  If AWIDUNQ is not present, then the signal width is $\text{ceil}(\text{ID\_W\_WIDTH}/8)$ .	Output	Write address ID	Connect to the corresponding completer device, if populated.
<prefix>_AWADDR[n:0]	<prefix>_AWADDRCHK  Signal width is $\text{ceil}(\text{ADDR\_WIDTH}/8)$	Output	Write address. The width is configurable from 32-64.	Connect to the corresponding completer device, if populated.
<prefix>_AWLEN[7:0]	<prefix>_AWLENCHK	Output	Write burst length	Connect to the corresponding completer device, if populated.
<prefix>_AWSIZE[2:0]	<prefix>_AWCTLCHK0	Output	Write burst size	Connect to the corresponding completer device, if populated.
<prefix>_AWBURST[1:0]	<prefix>_AWCTLCHK0	Output	Write burst type	Connect to the corresponding completer device, if populated.
<prefix>_AWLOCK	<prefix>_AWCTLCHK0	Output	Write lock type	Connect to the corresponding completer device, if populated.
<prefix>_AWCACHE[3:0]	<prefix>_AWCTLCHK1	Output	Write cache type	Connect to the corresponding completer device, if populated.
<prefix>_AWPROT[2:0]	<prefix>_AWCTLCHK0	Output	Write protection type	Connect to the corresponding completer device, if populated.
<prefix>_AWQOS[3:0]	<prefix>_AWCTLCHK1	Output	Write QoS value	Connect to the corresponding completer device, if populated.
<prefix>_AWREGION[3:0]	<prefix>_AWCTLCHK1	Output	Write region identifier	Connect to the corresponding completer device, if populated.
<prefix>_AWUSER[n:0]	<prefix>_AWUSERCHK  Signal width is $\text{ceil}(\text{USER\_REQ\_WIDTH})/8$ .	Output	User-specified extension to write address payload	Connect to the corresponding completer device, if populated.
<prefix>_AWVALID	<prefix>_AWVALIDCHK	Output	Write address valid	Connect to the corresponding completer device, if populated.
<prefix>_AWNSAID[3:0]	<prefix>_AWNNAIDCHK	Output	NSAID signal that is associated with the write address channel	Connect to the corresponding completer device, if populated.

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWREADY	<prefix>_AWREADYCHK	Input	Write address ready	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

### A.3.2 AMNI AXI5 extension write address channel signals

All AMNI interface configurations except AXI3 configurations contain a set of AXI5 extensions to the write address channel signals. These signals transport AXI5 write address information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-29: AMNI AXI5 extension write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWATOP	<prefix>_AWACTLCHK3	Output	AW atomic operation.  Indicates the type and endianness of atomic transactions.	Connect to the corresponding completer device, if populated
<prefix>_AWTRACE	<prefix>_AWTRACECHK	Output	Trace signals that are associated with the write address channel	Connect to the corresponding completer device, if populated
<prefix>_AWLOOP	<prefix>_AWLOOPCHK	Output	LOOP signal that is associated with the write address channel	Connect to the corresponding completer device, if populated
<prefix>_AWMPAM	<prefix>_AWMPAMCHK	Output	Write address channel MPAM information	Connect to the corresponding completer device, if populated
<prefix>_AWIDUNQ	<prefix>_AWIDCHK  Signal width is $\text{ceil}((\text{ID\_W\_WIDTH}/1)/8)$  If AWIDUNQ is not present, then width is $\text{ceil}(\text{ID\_W\_WIDTH}/8)$ .	Output	Write address channel unique ID indicator, active-HIGH	Connect to the corresponding completer device, if populated
<prefix>_AWTAGOP	<prefix>_AWTCTLCHK3	Output	Write request tag operation. Encoded as:  <b>00</b> Invalid  <b>01</b> Transfer  <b>10</b> Update  <b>11</b> Match	Connect to the corresponding completer device, if populated

### A.3.3 AMNI ACE-Lite write address channel signals

ACE5-Lite and ACE5-LiteACP AMNI interface configurations contain a set of ACE-Lite write address channel signals. These signals transport ACE-Lite write address information between the upstream AMNI and the downstream ACE-Lite device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-30: AMNI ACE-Lite write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWSNOOP[3:0]	<prefix>_AWCTLCHK2	Output	The transaction type for shareable write transactions	Connect to the corresponding completer device, if populated.
<prefix>_AWDOMAIN[1:0]	<prefix>_AWCTLCHK2	Output	Indicates the shareability domain of a write transaction	Connect to the corresponding completer device, if populated.

### A.3.4 AMNI ACE5-Lite extension write address channel signals

ACE5-Lite and ACE5-LiteACP AMNI interface configurations contain a set of ACE5-Lite extensions to the write address channel signals. These signals transport ACE5-Lite write address information between the upstream AMNI and the downstream ACE5-Lite device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-31: AMNI ACE5-Lite extension write address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_AWSTASHNID	<prefix>_AWSTASHNIDCHK	Output	Indicates the node identifier of the physical interface that is the target interface for the cache stashing operation	Connect to the corresponding completer device, if populated.
<prefix>_AWSTASHNIDEN	<prefix>_AWSTASHNIDENCHK	Output	When asserted, this signal indicates that the AWSTASHNID signal is valid and must be used.	Connect to the corresponding completer device, if populated.
<prefix>_AWSTASHLPID	<prefix>_AWSTASHLPIDCHK	Output	Indicates the logical processor subunit associated with the physical interface that is the target for the cache stashing operation	Connect to the corresponding completer device, if populated.
<prefix>_AWSTASHLPIDEN	<prefix>_AWSTASHLPIDENCHK	Output	When asserted, this signal indicates that the AWSTASHLPID signal is enabled and must be used.	Connect to the corresponding completer device, if populated.
<prefix>_AWCMO	<prefix>_AWCTLCHK3	Output	Indicates the type of CMO	Connect to the corresponding completer device, if populated.



### A.3.5 AMNI AXI4 write data channel signals

All AMNI interface configurations contain a set of AXI4 write data channel signals. These signals transport AXI write data information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-32: AMNI AXI4 write data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_WID[n:0]	<prefix>_WID	Output	The output write data ID.  <b>Note:</b> NI-710AE does not perform write data interleaving across transactions. The signal exists only for integration purposes.	Connect to the corresponding completer device, if populated.
<prefix>_WDATA[n:0]	<prefix>_WDATACHK  Signal width is DATA_WIDTH/8.	Output	Write data	Connect to the corresponding completer device, if populated.
<prefix>_WSTRB[(DATA_WIDTH/8)-1:0]	<prefix>_WSTRBCHK  Signal width is ceil(DATA_WIDTH/64)	Output	Write byte lane strobes	Connect to the corresponding completer device, if populated.
<prefix>_WLAST	<prefix>_WLASTCHK	Output	Write data last transfer indication	Connect to the corresponding completer device, if populated.
<prefix>_WUSER[n:0]	<prefix>_WUSERCHK  Signal width is ceil(USER_DATA_WIDTH/8)	Output	User-specified extension to write data payload	Connect to the corresponding completer device, if populated.
<prefix>_WVALID	<prefix>_WVALIDCHK	Output	Write data valid	Connect to the corresponding completer device, if populated.
<prefix>_WREADY	<prefix>_WREADYCHK	Input	Write data ready	Connect to the corresponding completer device, if populated.

## A.3.6 AMNI AXI5 extension write data channel signals

All AMNI interface configurations except AXI3 configurations contain a set of AXI5 extensions to the write data channel signals. These signals transport AXI5 write data information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-33: AMNI AXI5 extension write data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_WTRACE	<prefix>_WTRACECHK	Output	Trace signals associated with the write data channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_WTAG	<prefix>_WTAGCHK  Signal width is $\text{ceil}(\text{DATA\_WIDTH}/128)$  WTAGCHK[n] is the parity of WTAGUPDATE[n], WTAG[(4n) + 3:(4n)]	Output	The tag associated with write data.  There is a 4-bit tag for each 128 bits of data, with a minimum of 4 bits.  WTAG[(((4 × n) – 1):4 × (n – 1))] corresponds to WDATA[(((128 × n) – 1):128 × (n – 1))]  <b>Note:</b> WTAG has the same validity rules as WDATA.	Connect to the corresponding completer device, if populated.
<prefix>_WTAGUPDATE	<prefix>_WTAGCHK	Output	Indicates which tags must be written to memory when an Update operation occurs: <ul style="list-style-type: none"> <li>If a bit is asserted, then the corresponding tags must be written to memory.</li> <li>If a bit is deasserted, then the corresponding tags are invalid.</li> </ul> There is 1 bit for each 4 bits of tag. WTAGUPDATE[n] corresponds to WTAG[(4n) + 3:(4n)].  WTAGUPDATE bits outside of the transaction container must be deasserted.  For operations other than Update, WTAGUPDATE must be deasserted. It can be asserted or deasserted for Update operations.	Connect to the corresponding completer device, if populated.

### A.3.7 AMNI AXI4 write response channel signals

All AMNI interface configurations contain a set of AXI4 write response channel signals. These signals transport AXI write data information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-34: AMNI AXI4 write response channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_BID[n:0]	<prefix>_BIDCHK  Signal width is $\text{ceil}((\text{ID\_W\_WIDTH} + 1)/8)$ .  If BIDUNQ is not present, then signal width is $\text{ceil}(\text{ID\_W\_WIDTH}/8)$ .	Input	Write response ID. Width is configurable.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BRESP[1:0]	<prefix>_BRESPCHK	Input	Write response	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BUSER[n:0]	<prefix>_BUSERCHK  Signal width is $\text{ceil}(\text{USER\_RESP\_WIDTH}/8)$	Input	User-specified extension to write response payload	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BVALID	<prefix>_BVALIDCHK	Input	Write response valid	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BREADY	<prefix>_BREADYCHK	Output	Write response ready	Connect to the corresponding completer device, if populated.

### A.3.8 AMNI AXI5 extension write response channel signals

All AMNI interface configurations except AXI3 configurations contain a set of AXI5 extensions to the write response channel signals. These signals transport AXI5 write data information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-35: AMNI AXI5 extension write response channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_BTRACE	<prefix>_BTRACECHK	Input	Trace signal that is associated with the write response channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_BLOOP	<prefix>_BLOOPCHK	Input	LOOP signal that is associated with the write response channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BIDUNQ	<prefix>_BIDCHK  Signal width is $\text{ceil}((\text{ID\_W\_WIDTH} + 1)/8)$ .  If BIDUNQ is not present then, width is $\text{ceil}(\text{ID\_WIDTH}/8)$ .	Input	Write response channel unique ID indicator, active-HIGH	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BCOMP	<prefix>_BRESPCHK	Input	Indicates that the write is observable	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BPERSIST	<prefix>_BRESPCHK	Input	Indicates that the write data is updated in persistent memory. Can only be asserted for transactions where AWCMO is CleanSharedPersist or CleanSharedDePersist.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_BTAGMATCH	<prefix>_BRESPCHK	Input	Indicates the result of a tag comparison on a write transaction:  <b>00</b> Not a match transaction  <b>01</b> No match result  <b>10</b> Fail  <b>11</b> Pass	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

### A.3.9 AMNI AXI4 read address channel signals

All AMNI interface configurations contain a set of AXI4 read address channel signals. These signals transport AXI read address information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

## Signal definitions

**Table A-36: AMNI AXI4 read address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARID[n:0]	<prefix>_ARIDCHK[n:0]  Signal width is $\text{ceil}((\text{ID\_R\_WIDTH}+1)/8)$  If ARIDUNQ is not present, then signal width is $\text{ceil}(\text{ID\_R\_WIDTH})/8$ .	Output	Read data ID. Width is configurable.	Connect to the corresponding completer device, if populated.
<prefix>_ARADDR[n:0]	<prefix>_ARADDRCHK  Signal width is $\text{ceil}(\text{ADDR\_WIDTH}/8)$	Output	Address of the first transfer in a read transaction	Connect to the corresponding completer device, if populated.
<prefix>_ARLEN[7:0]	<prefix>_ARLENCHK	Output	Length. The exact number of data transfers in a read transaction.	Connect to the corresponding completer device, if populated.
<prefix>_ARSIZE[2:0]	<prefix>_ARCTLCHK0	Output	Size. The number of bytes in each data transfer in a read transaction.	Connect to the corresponding completer device, if populated.
<prefix>_ARBURST[1:0]	<prefix>_ARCTLCHK0	Output	Burst type. Indicates how address changes between each transfer in a read transaction.	Connect to the corresponding completer device, if populated.
<prefix>_ARLOCK	<prefix>_ARCTLCHK0	Output	Information about the atomic characteristics of a read transaction	Connect to the corresponding completer device, if populated.
<prefix>_ARCACHE[3:0]	<prefix>_ARCCTLCHK1	Output	Indicates how a read transaction is required to progress through a system	Connect to the corresponding completer device, if populated.
<prefix>_ARPROT[2:0]	<prefix>_ARCTLCHK0	Output	Protection attributes of a read transaction: <ul style="list-style-type: none"> <li>Privilege</li> <li>Security level</li> <li>Access type</li> </ul>	Connect to the corresponding completer device, if populated.
<prefix>_ARQOS[3:0]	<prefix>_ARCTLCHK1	Output	QoS identifier for a read transaction	Connect to the corresponding completer device, if populated.
<prefix>_ARREGION[3:0]	<prefix>_ARCTLCHK1	Output	Read region identifier	Connect to the corresponding completer device, if populated.
<prefix>_ARUSER[n:0]	<prefix>_ARUSERCHK  Signal width is $\text{ceil}(\text{USER\_REQ\_WIDTH}/8)$	Output	User-defined extension for the read address channel	Connect to the corresponding completer device, if populated.
<prefix>_ARVALID	<prefix>_ARVALIDCHK	Output	Indicates that the read address channel signals are valid	Connect to the corresponding completer device, if populated.
<prefix>_ARREADY	<prefix>_ARREADYCHK	Output	Indicates that a transfer on the read address channel can be accepted	Connect to the corresponding completer device, if populated.
<prefix>_ARNSAID[3:0]	<prefix>_ARNSAIDCHK	Output	NSAID associated with the read address channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

### A.3.10 AMNI AXI5 extension read address channel signals

All AMNI interface configurations except AXI3 configurations contain a set of AXI5 extensions to the read address channel signals. These signals transport AXI5 read address information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-37: AMNI AXI5 extension read address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARTRACE	<prefix>_ARTRACECHK	Output	Trace signal that is associated with the read address channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_ARLOOP	<prefix>_ARLOOPCHK	Output	The LOOP signal that is associated with the read address channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_ARMPAM	<prefix>_ARMPAMCHK	Output	Read address channel MPAM information	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_ARIDUNQ	<prefix>_ARIDCHK  Width is $\text{ceil}((\text{ID\_W\_WIDTH} + 1)/8)$ .  If BIDUNQ is not present, then width is $(\text{ID\_W\_WIDTH})/8$ .	Output	Read address channel unique ID indicator, active-HIGH	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_ARCHUNKEN	<prefix>_ARCTLCHK3	Output	If this signal is asserted, read data for this transaction can be returned out of order, in 128-bit chunks.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_ARTAGOP	<prefix>_ARCTLCHK3	Output	Read request tag operation. Encoded as:  <b>0b00</b> Invalid  <b>0b01</b> Transfer  <b>0b10</b> Reserved  <b>0b11</b> Fetch	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

### A.3.11 AMNI ACE-Lite read address channel signals

ACE5-Lite and ACE5-LiteACP AMNI interface configurations contain a set of ACE-Lite read address channel signals. These signals transport ACE-Lite read address information between the upstream AMNI and the downstream ACE5-Lite device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-38: AMNI ACE-Lite read address channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARSNOOP[3:0]	<prefix>_ARCTLCHK2	Output	Transaction type for shareable read transactions	Connect to the corresponding completer device, if populated.
<prefix>_ARDOMAIN[1:0]	<prefix>_ARCTLCHK2	Output	Shareability domain of a read transaction	Connect to the corresponding completer device, if populated.

### A.3.12 AMNI AXI4 read data channel signals

All AMNI interface configurations contain a set of AXI4 read data channel signals. These signals transport AXI read data information between the upstream AMNI and the downstream AXI device.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-39: AMNI AXI4 read data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_RID[n:0]	<prefix>_RIDCHK  Signal width is $\text{ceil}((\text{ID\_R\_WIDTH}+1)/8)$ .  If RIDUNQ is not present then, signal width is $\text{ceil}(\text{ID\_R\_WIDTH}/8)$ .	Input	Read data ID. Width is configurable.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RDATA[DATA_WIDTH-1:0]	<prefix>_RDATACHK[DATA_WIDTH-1:0]	Input	Read data	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RRESP[3:0]	<prefix>_RRESPCHK	Input	Read data response	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RLAST	<prefix>_RLASTCHK	Input	Read data last transfer indication	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RUSER[n:0]	<prefix>_RUSERCHK  Signal width is $\text{ceil}(\text{USER\_DATA\_WIDTH} + \text{USER\_RESP\_WIDTH}/8)$	Input	User-specified extension to read data payload	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_RVALID	<prefix>_RVALID	Input	Read data valid	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RREADY	<prefix>_RREADY	Output	Read data ready	Connect to the corresponding completer device, if populated.

### A.3.13 AMNI AXI5 extension read data channel signals

All AMNI interface configurations except AXI3 configurations contain a set of AXI5 extensions to the read data channel signals. These signals transport AXI5 read data information between the upstream AMNI and the downstream AXI device.

In this table, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-40: AMNI AXI5 extension read data channel signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_RTRACE	<prefix>_RTRACECHK	Input	Trace signal that is associated with the read data channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RLOOP	<prefix>_RLOOPCHK	Input	LOOP signal associated with the read data channel	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RIDUNQ	<prefix>_RIDCHK  Signal width is $\text{ceil}((\text{ID\_R\_WIDTH})+1)/8$  If RIDUNQ is not present then width is $(\text{ID\_R\_WIDTH}/8)$ .	Input	Read data channel unique ID indicator, active-HIGH	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RCHUNKV	<prefix>_RCHUNKCHK	Input	If this signal is asserted, RCHUNKNUM and RCHUNKSTRB are valid for this transfer.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RCHUNKNUM	<prefix>_RCHUNKCHK	Input	Indicates the chunk number being transferred. Chunks are numbered incrementally from zero, according to the data width and base address of the transaction.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.



Signal	Check signal	Direction	Description	Connection information
<prefix>_RCHUNKSTRB	<prefix>_RCHUNKCHK	Input	Indicates which part of read data is valid for this transfer, each bit corresponds to 128 bits of data. For example: <ul style="list-style-type: none"> <li>RCHUNKSTRB[0] corresponds to RDATA[127 :0]</li> <li>RCHUNKSTRB[1] corresponds to RDATA[255 :128]</li> </ul>	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_RTAG	<prefix>_RTAGCHK  Signal width is $\text{ceil}(\text{DATAWIDTH}/128)$ .  RTAGCHK <sub>n</sub> is the parity of RTAG[4n+3:4n]	Input	The tag associated with read data.  There is a 4-bit tag for each 128 bits of data, with a minimum of 4 bits.  RTAG[ $((4 \times n)-1) : 4 \times (n-1)$ ] corresponds to RDATA[ $((128 \times n)-1) : 128 \times (n-1)$ ]  <b>Note:</b> RTAG has the same validity rules as RDATA.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

### A.3.14 AMNI AXI3 interface configuration signal changes

Configuring the external interface type of the AMNI to AXI3 changes the width of some of the AXI signals. This configuration affects the read address, write address, and write data channels.

In this section, <prefix> represents <PROTOCOL>\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

For comparison with the AXI4 interface configuration, see the following sections:

- [AMNI AXI4 read address channel signals](#)
- [AMNI AXI4 write address channel signals](#)
- [AMNI AXI4 write data channel signals](#)

### Signal definitions

**Table A-41: AMNI AXI3 interface configuration signal changes**

Signal	Check signal	Direction	Description	Connection information
<prefix>_ARLEN[3:0]	<prefix>_ARLENCHK	Output	Length. The exact number of data transfers in a read transaction.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_ARLOCK[1:0]	<prefix>_ARCTLCHK0	Output	Information about the atomic characteristics of a read transaction	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_AWLEN[3:0]	<prefix>_AWLENCHK	Output	Write burst length	Connect to the corresponding completer device, if populated.
<prefix>_AWLOCK[1:0]	<prefix>_AWCTLCHK0	Output	Write lock type	Connect to the corresponding completer device, if populated.
<prefix>_AWID[n:0]	<prefix>_AWIDCHK	Output	WID pin	Connect to the corresponding completer device, if populated.

## A.3.15 AXIS interface signals

The AMNI AXIS interface connects to the AXIS interface of a Cortex-R52 or Cortex-R52+ processor.

The AMNI AXIS interface contains the following signal groups:

- AMNI AXIS read address channel signals
- AMNI AXIS read data channel signals
- AMNI AXIS write address channel signals
- AMNI AXIS write data channel signals
- AMNI AXIS write response channel signals

### A.3.15.1 AMNI AXIS read address channel signals

The AMNI AXIS interface read address channel signals transport read address information between NI-710AE and the AXIS interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read address channel signals for the AMNI AXIS interface between NI-710AE and Cortex-R52 or Cortex-R52+ processors.

**Table A-42: AMNI AXIS read address channel signals**

Signal	Code Signal	Direction	Description	Connection information
<prefix>_ARADDRS[31:0]	<prefix>_ARADDRCODES[6:0]	Output	Read address.	
<prefix>_ARIDS[n:0]	ARIDCODES[4:0]	Output	Read address ID.  <b>Note:</b> n is configurable for Cortex-R52.  n is fixed at 15 for Cortex-R52+.	Connect to the corresponding completer device, if populated.
ARLENS[7:0]	ARCTL1CODES[4:0]	Output	Instruction fetch burst length.	Connect to the corresponding completer device, if populated.
ARSIZES[2:0]	ARCTL1CODES[4:0]	Output	Read burst size.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.

Signal	Code Signal	Direction	Description	Connection information
ARBURSTS[1:0]	ARCTL1CODES[4:0]	Output	Read burst type.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
ARLOCKS	ARCTL1CODES[4:0]	Output	Read lock type.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
ARCACHES[3:0]	ARCTL0CODES[4:0]	Output	Read memory type.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
ARPROTS[2:0]	ARCTL0CODES[4:0]	Output	Protection information, privileged/normal access.	Connect to the corresponding completer device, if populated.
ARQOSS[3:0]	ARCTL0CODES[4:0]	Output	Quality of service read identifier.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
ARVALIDS	ARVALIDCODES	Output	Read address valid.	Connect to the corresponding completer device, if populated.
ARREADYS	ARREADYCODES	Input	Read address ready.	Connect to the corresponding completer device, if populated.

### A.3.15.2 AMNI AXIS read data channel signals

The AMNI AXIS interface read data channel signals transport read data information between NI-710AE and the AXIS interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the read data channel signals for the AMNI AXIS interface.

**Table A-43: AMNI AXIS read data channel signals**

Signal	Code Signal	Direction	Description	Connection information
RREADY	RREADYCODES	Output	Read data ready.	Connect to the corresponding completer device, if populated.
RDATAS[127:0]	RDATA[15:0]	Input	Read data.	Connect to the corresponding completer device, if populated.

Signal	Code Signal	Direction	Description	Connection information
RIDS[n:0]	RIDCODES[4:0]	Input	Read data ID.  <b>Note:</b> n is configurable for Cortex-R52.  n is fixed at 15 for Cortex-R52+.	Connect to the corresponding completer device, if populated.
RLASTS	RCTLCODES[2:0]	Input	Indicates the last transfer in a read burst.	Connect to the corresponding completer device, if populated.
RRESPS[1:0]	RCTLCODES[2:0]	Input	Read response.	Connect to the corresponding completer device, if populated.
RVALIDS	RVALIDCODES	Input	Read data valid.	Connect to the corresponding completer device, if populated.

### A.3.15.3 AMNI AXIS write address channel signals

The AMNI AXIS interface write address channel signals transport write address information between NI-710AE and the AXIS interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write address channel signals for the AMNI AXIS interface.

**Table A-44: AMNI AXIS write address channel signals**

Signal	Code Signal	Direction	Description	Connection information
AWADDRS[31:0]	AWADDRCODES[6:0]	Output	Write address.	Connect to the corresponding completer device, if populated.
AWIDS[n:0]	AWIDCODES[4:0]	Output	Write address ID.  <b>Note:</b> n is configurable for Cortex-R52.  n is fixed at 15 for Cortex-R52+.	Connect to the corresponding completer device, if populated.
AWLENS[7:0]	AWCTL1CODES[4:0]	Output	Write transfer burst length.	Connect to the corresponding completer device, if populated.
AWSIZES[2:0]	AWCTL1CODES[4:0]	Output	Write burst size.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
AWBURSTS[1:0]	AWCTL1CODES[4:0]	Output	Write burst type.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.

Signal	Code Signal	Direction	Description	Connection information
AWLOCKS	AWCTL1CODES[4:0]	Output	Write lock type.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
AWCACHES[3:0]	AWCTL0CODES[4:0]	Output	Write cache type.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
AWPROTS[2:0]	AWCTL0CODES[4:0]	Output	Protection information, privileged/normal access.	Connect to the corresponding completer device, if populated.
AWQOSS[3:0]	AWCTL0CODES[4:0]	Output	Quality of service write identifier.  <b>Note:</b> Only present when NI-710AE xface is configured for Cortex-R52+	Connect to the corresponding completer device, if populated.
AWVALIDS	AWVALIDCODES	Output	Write address valid.	Connect to the corresponding completer device, if populated.
AWREADYS	AWREADYCODES	Output	Write address ready.	Connect to the corresponding completer device, if populated.

#### A.3.15.4 AMNI AXIS write data channel signals

The AMNI AXIS interface write data channel signals transport write data information between NI-710AE and the AXIS interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write data channel signals for the AMNI AXIS interface.

**Table A-45: AMNI AXIS write data channel signals**

Signal	Code Signal	Direction	Description	Connection information
WDATAS[127:0]	WDATACODES[15:0]	Output	Write data.	Connect to the corresponding completer device, if populated.
WLASTS	WCTLCODES[4:0]	Output	Indicates the last data transfer of a burst.	Connect to the corresponding completer device, if populated.
WSTRBS[15:0]	WCTLCODES[4:0]	Output	Write byte-lane strobes.	Connect to the corresponding completer device, if populated.
WVALIDS	WVALIDCODES	Output	Read data valid.	Connect to the corresponding completer device, if populated.
WREADYS	WREADYCODES	Input	Read data ready.	Connect to the corresponding completer device, if populated.

### A.3.15.5 AMNI AXIS write response channel signals

The AMNI AXIS interface write response channel signals transport write response information between NI-710AE and the AXIS interface of a Cortex-R52 or Cortex-R52+ processor.

The following table shows the write response channel signals for the AXIS interface.

**Table A-46: AXIS write response channel signals**

Signal	Code Signal	Direction	Description	Connection information
BREADYs	BREADYCODES	Output	Write response ready.	Connect to the corresponding completer device, if populated.
BIDS[m:0]	BIDCODES[4:0]	Input	Write response ID.  <b>Note:</b> n is configurable for Cortex-R52.  n is fixed at 15 for Cortex-R52+.	Connect to the corresponding completer device, if populated.
BRESPs[1:0]	BCTLCODES[2:0]	Input	Write response.	Connect to the corresponding completer device, if populated.
BVALIDs	BVALIDCODES	Output	Write response valid.	Connect to the corresponding completer device, if populated.

## A.4 HSNi external interface types and associated signal groups

You can configure an HSNi to have either an AHB5 or AHB5 mirrored requester interface. The AHB5 interface has an extra group of signals compared to the AHB5 mirrored requester interface.

### AHB5 external interface signal groups

If your HSNi has an AHB5 interface, see the following sections to find the details of the AHB signals:

- [HSNi AHB-Lite request signals](#)
- [HSNi AHB5 extension request signals](#)
- [HSNi AHB-Lite response signals](#)
- [HSNi AHB5 extension response signals](#)
- [Other HSNi AHB signals](#)

### AHB5 mirrored requester external interface signal groups

If your HSNi has an AHB5 mirrored requester interface, see the following sections to find the details of the AHB signals:

- [HSNI AHB-Lite request signals](#)
- [HSNI AHB5 extension request signals](#)
- [HSNI AHB-Lite response signals](#)
- [HSNI AHB5 extension response signals](#)

## A.4.1 HSNI AHB-Lite request signals

All HSNI interface configurations have a set of AHB-Lite request signals. These signals transport AHB-Lite request information between an upstream AHB requester device and the downstream HSNI.

In this section, <prefix> represents AHB\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-47: HSNI AHB-Lite request signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HADDR	<prefix>_HADDRCHK	Input	AHB address bus	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HBURST	<prefix>_HCTRLCHK1	Input	Burst type	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HMASTLOCK	<prefix>_HCTRLCHK1	Input	When HIGH, indicates that the current transfer is part of a locked sequence	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HPROT[3:0]	<prefix>_HPROTCHK	Input	The protection control signals	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HSIZE	<prefix>_HCTRLCHK1	Input	Indicates the size of the transfer	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HTRANS	<prefix>_HTRANSCHK	Input	Indicates the transfer type of the current transfer	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HWDATA	<prefix>_HWDATACHK	Input	The write data	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HWRITE	<prefix>_HCTRLCHK1	Input	Indicates the transfer direction being write or read	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HAUSER	<prefix>_HAUSERCHK	Input	Address channel User signals	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HWUSER	<prefix>_HWUSERCHK	Input	Write data channel User signals	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.4.2 HSNi AHB5 extension request signals

All HSNi interface configurations have a set of AHB5 extensions to the request signals. These signals transport AHB5 request information between the upstream AHB requester device and the downstream HSNi.

In this section, <prefix> represents AHB\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-48: HSNi AHB5 extension request signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HPROT	<prefix>_HPROTCHK	Input	The 3-bit extension of the HPROT signal that adds extended memory types	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HNONSEC	<prefix>_HCTRLCHK1	Input	Indicates whether the transfer is Secure or Non-secure	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HEXCL	<prefix>_HCTRLCHK2	Input	Exclusive transfer	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HMASTER	<prefix>_HCTRLCHK2	Input	The requester identifier which is only used for exclusive transfer	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.4.3 HSNi AHB-Lite response signals

All HSNi interface configurations have a set of AHB-Lite response signals. These signals transport AHB-Lite response information between the upstream AHB requester device and the downstream HSNi.

In this section, <prefix> represents AHB\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-49: HSNi AHB-Lite response signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HRDATA	<prefix>_HRDATACHK	Output	The read data from the multiplexer	Connect to the corresponding requester device, if populated.
<prefix>_HREADY	<prefix>_HREADYCHK	Output	Ready output from HSNi core	Connect to the corresponding requester device, if populated.
<prefix>_HRESP	<prefix>_HRESPCHK	Output	The transfer response from the multiplexer	Connect to the corresponding requester device, if populated.
<prefix>_HRUSER	<prefix>_HRUSERCHK	Output	The read data channel User signal from the multiplexer	Connect to the corresponding requester device, if populated.



## A.4.4 HSNI AHB5 extension response signals

All HSNI interface configurations have a set of AHB5 extensions to the response signals. These signals transport AHB5 response information between the upstream AHB requester device and the downstream HSNI.

In this section, <prefix> represents AHB\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-50: HSNI AHB5 extension response signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HEXOKAY	<prefix>_HRESPCHK	Output	Exclusive Okay response	Connect to the corresponding requester device, if populated.

## A.4.5 Other HSNI AHB signals

If you configure a HSNI to have a full AHB interface, instead of a requester mirror interface, the interface has an extra set of signals. These signals transport control information between the upstream AHB completer device and the downstream HSNI.

In this section, <prefix> represents AHB\_SLAVE\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-51: Other HSNI AHB signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HREADY	<prefix>_HREADYCHK	Input	The HREADY from the multiplexer going to all requesters and completers	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.
<prefix>_HSEL	<prefix>_HSELCHK	Input	The completer select signal from the decoder	Connect to the corresponding requester device, if populated. Otherwise, tie LOW.

## A.5 HMNI external interface types and associated signal groups

You can configure an HMNI to have either an AHB5 or AHB5 mirrored completer interface. The AHB5 interface has an extra group of signals compared to the AHB5 mirrored completer interface.

### AHB5 external interface signal groups

If your HMNI has an AHB5 interface, see the following sections to find the details of the AHB signals:

- [HMNI AHB-Lite request signals](#)
- [HMNI AHB5 extension request signals](#)
- [HMNI AHB-Lite response signals](#)

- [HMNI AHB5 extension response signals](#)
- [Other HMNI AHB signals](#)

## AHB5 mirrored completer external interface signal groups

If your HMNI has an AHB5 mirrored completer interface, see the following sections to find the details of the AHB signals:

- [HMNI AHB-Lite request signals](#)
- [HMNI AHB5 extension request signals](#)
- [HMNI AHB-Lite response signals](#)
- [HMNI AHB5 extension response signals](#)

### A.5.1 HMNI AHB-Lite request signals

All HMNI configurations have a set of AHB-Lite request signals. These signals transmit AHB-Lite request information between the upstream HMNI and the downstream AHB completer.

In this section, <prefix> represents AHB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

#### Signal definitions

**Table A-52: HMNI AHB-Lite request signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HADDR	<prefix>_HADDRCHK  Signal width is $\text{ceil}(\text{ADDR\_WIDTH}/8)$ .	Output	AHB address bus	Connect to the corresponding completer device, if populated.
<prefix>_HBURST	<prefix>_HCTRLCHK1	Output	Burst type	Connect to the corresponding completer device, if populated.
<prefix>_HMASTLOCK	<prefix>_HCTRLCHK1	Output	When HIGH, indicates that the current transfer is part of a locked sequence	Connect to the corresponding completer device, if populated.
<prefix>_HPROT[3:0]	<prefix>_HPROTCHK	Output	Protection control signals	Connect to the corresponding completer device, if populated.
<prefix>_HSIZE	<prefix>_HCTRLCHK1	Output	Indicates the size of the transfer	Connect to the corresponding completer device, if populated.
<prefix>_HTRANS	<prefix>_HTRANSCHK	Output	Indicates the transfer type of the current transfer	Connect to the corresponding completer device, if populated.
<prefix>_HWDATA	<prefix>_HWDATACHK  Signal width is $\text{DATA\_WIDTH}/8$ .	Output	Write data	Connect to the corresponding completer device, if populated.
<prefix>_HWRITE	<prefix>_HCTRLCHK1	Output	Indicates the transfer direction being write or read	Connect to the corresponding completer device, if populated.
<prefix>_HAUSER	<prefix>_HAUSERCHK  Signal width is $\text{ceil}(\text{USER\_REQ\_WIDTH}/8)$ .	Output	Address channel User signals	Connect to the corresponding completer device, if populated.

Signal	Check signal	Direction	Description	Connection information
<prefix>_HWUSER	<prefix>_HWUSERCHK  Signal width is ceil(USER_REQ_WIDTH/8)	Output	Write data channel User signals	Connect to the corresponding completer device, if populated.

## A.5.2 HMNI AHB5 extension request signals

All HMNI configurations have a set of AHB5 extensions to the request signals. These signals transmit AHB5 request information between the upstream HMNI and the downstream AHB completer.

In this section, <prefix> represents AHB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-53: HMNI AHB5 extension request signals**

Signal	Check Signal	Direction	Description	Connection information
<prefix>_HPROT[6:4]	<prefix>_HPROTCHK[6:4]	Output	The 3-bit extension of the HPROT signal that adds extended memory types	Connect to the corresponding completer device, if populated.
<prefix>_HNONSEC	<prefix>_HCTRLCHK1	Output	Indicates whether the transfer is Secure or Non-secure	Connect to the corresponding completer device, if populated.
<prefix>_HEXCL	<prefix>_HCTRLCHK2	Output	Exclusive transfer	Connect to the corresponding completer device, if populated.
<prefix>_HMASTER	<prefix>_HCTRLCHK2	Output	Requester identifier which is only used for Exclusive transfer	Connect to the corresponding completer device, if populated.

## A.5.3 HMNI AHB-Lite response signals

All HMNI configurations have a set of AHB-Lite response signals. These signals transmit AHB-Lite response information between the upstream HMNI and the downstream AHB completer.

In this section, <prefix> represents AHB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-54: HMNI AHB-Lite response signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HRDATA	<prefix>_HRDATACHK  Signal width is DATA_WIDTH/8.	Input	The read data from the multiplexer.  HRDATACHK must be driven with the correct parity value corresponding to HRDATA even when the responder is in wait state, that is, irrespective of HREADYOUT.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

Signal	Check signal	Direction	Description	Connection information
<prefix>_HREADY/ HREADYOUT	<prefix>_HREADYCHK/ HREADYOUTCHK	Input	If the interface is an HMNI interface, this signal is the HREADY signal from the multiplexer.  In AHB mirror mode, this signal is the HREADYOUT signal from the completer.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_HRESP	<prefix>_HRESPCHK	Input	The transfer response from the multiplexer	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.
<prefix>_HRUSER	<prefix>_HRUSERCHK  Signal width is ceil(USER_DATA_WIDTH/8).	Input	The read data channel User signal from the multiplexer	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

## A.5.4 HMNI AHB5 extension response signals

All HMNI configurations have a set of AHB5 extensions to the response signals. These signals transmit AHB5 response information between the upstream HMNI and the downstream AHB completer.

In the section, <prefix> represents AHB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-55: HMNI AHB5 extension response signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HEXOKAY	<prefix>_HRESPCHK	Input	Exclusive Okay response.  HEXOKAY and HRESP must always be valid and the corresponding parity signal HRESPCHK must be driven at all times.	Connect to the corresponding completer device, if populated. Otherwise, tie LOW.

## A.5.5 Other HMNI AHB signals

If you configure an HMNI to have a full AHB interface, instead of a completer mirror interface, the interface has an extra set of signals. These signals transport control information between the upstream HMNI and the downstream AHB completer device.

In this section, <prefix> represents AHB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

## Signal definitions

**Table A-56: Other HMNI AHB signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_HREADY	<prefix>_HREADYCHK	Output	The HREADY from the multiplexer, which goes to all requesters and completers	Connect to the corresponding completer device, if populated.
<prefix>_HSEL	<prefix>_HSELCHK	Output	The completer select signal from the decoder	Connect to the corresponding completer device, if populated.

## A.6 PMNI external interface types and associated signal groups

You can configure a PMNI to have either an APB3 or APB4 external requester interface. Support for APB5 in NI-710AE is limited to the signal parity protection feature only.

### APB3 external interface signal groups

If your PMNI has an APB3 interface, see the following sections to find the details of the APB signals:

- [PMNI APB signals](#)
- [PMNI APB3 signals](#)

### APB4 external interface signal groups

If your PMNI has an APB4 interface, see the following sections to find the details of the APB signals:

- [PMNI APB signals](#)
- [PMNI APB3 signals](#)
- [PMNI APB4 signals](#)

### APB5 external interface signal groups

If your PMNI has an APB5 interface, see the following sections to find the details of the APB signals:

- [PMNI APB signals](#)
- [PMNI APB3 signals](#)
- [PMNI APB4 signals](#)

The only APB5 feature that is supported in NI-710AE is signal parity protection. When APB5 support is enabled on a NI-710AE PMNI, extra check signals are added at the interfaces.

## A.6.1 PMNI APB signals

You can configure the PMNI to have an APB3, APB4, or APB 5 requester interface. These APB signals that are always present in the PMNI regardless of the interface configuration.

In this section, <prefix> represents APB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>.

### Signal definitions

**Table A-57: PMNI APB signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_PADDR_{0-15}	<prefix>_PADDRCHK_{0-15}	Output	APB address bus	Connect to the corresponding completer devices, if populated.
<prefix>_PSEL_{0-15}	<prefix>_PSELxCHK_{0-15}	Output	APB completer device select. PMNI supports up to 16 APB completers.	Connect to the corresponding completer devices, if populated.
<prefix>_PENABLE_{0-15}	<prefix>_PENABLECHK_{0-15}	Output	Enable. This signal indicates the second and subsequent cycles of an APB transfer.	Connect to the corresponding completer devices, if populated.
<prefix>_PWRITE_{0-15}	<prefix>_PCTRLCHK_{0-15}	Output	This signal indicates an APB read or write access:  <b>0</b> APB read access  <b>1</b> APB write access	Connect to the corresponding completer devices, if populated.
<prefix>_PWRITE_{0-15}	<prefix>_PWRITECHK_{0-15}	Output	Write data	Connect to the corresponding completer devices, if populated.
<prefix>_PRDATA_{0-15}	<prefix>_PRDATACHK_{0-15}	Input	APB read data	Connect to the corresponding completer devices, if populated.

## A.6.2 PMNI APB3 signals

You can configure the PMNI to have an APB3, APB4, or APB5 requester interface. These APB signals that are always present in the PMNI regardless of the interface configuration.

In this section, <prefix> represents APB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>. For more information about signal timing constraints and clock associations, see [Signal timing constraints and clock associations](#).

## Signal definitions

**Table A-58: PMNI APB3 signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_PREADY_{0-15}	<prefix>_PREADYCHK_{0-15}	Input	Ready. The APB completer uses this signal to extend an APB transfer (wait states).	Connect to the corresponding completer devices, if populated. Otherwise, tie LOW.
<prefix>_PSLVERR_{0-15}	<prefix>_PSLVERRCHK_{0-15}	Input	This signal indicates a transfer failure. APB peripherals are not required to support the PSLVERR pin. Where a peripheral does not include this pin, then the appropriate input to the PMNI is tied LOW.	Connect to the corresponding completer devices, if populated. Otherwise, tie LOW.

## A.6.3 PMNI APB4 signals

You can configure the PMNI to have an APB3, APB4, or APB5 requester interface. APB4 introduced new signals that were not present in APB3, so these signals are only present in PMNIs with APB4 interfaces.

In this section, <prefix> represents APB\_MASTER\_<ENDPOINT\_INTERFACE\_NAME>. For more information about signal timing constraints and clock associations, see [Signal timing constraints and clock associations](#).

## Signal definitions

**Table A-59: PMNI APB4 signals**

Signal	Check signal	Direction	Description	Connection information
<prefix>_PPROT_{0-15}	<prefix>_PCTRLCHK_{0-15}	Output	Protection type  <b>Note:</b> NI-710AE only supports Secure or Non-secure access indication corresponding to PPROT[1]. NI-710AE does not transport normal or privileged access and data or instruction access.	Connect to the corresponding completer devices, if populated.
<prefix>_PSTRB_{0-15}	<prefix>_PSTRBCHK_{0-15}	Output	APB write data strobes. This signal indicates which byte lanes to update during a write transfer. One write strobe for each 8-bit of the write data bus.  Therefore, PSTRB[n] corresponds to PWDATA[(8n + 7):(8n)]. Write strobes must not be active during a read transfer.	Connect to the corresponding completer devices, if populated.

## A.7 Miscellaneous AXI interface signals

NI-710AE provides protection on miscellaneous AXI interface signals.

### Signal definitions

**Table A-60: Miscellaneous AXI interface signals and check signals**

Signal	Check signal	Direction	Description	Connection information
AXI_MASTER_<ENDPOINT_INTERFACE_NAME>_AWAKEUP	AXI_MASTER_<ENDPOINT_INTERFACE_NAME>_AWAKEUPCHK	Output	Indicates that the requester interface has active transactions. It can be used as an indicator to turn on the clock to downstream components.  AWAKEUP and AWAKEUPCHK signals are also present at Cortex-R52 and Cortex-R52+ interfaces. These inputs must be driven appropriately.	-
AXI_SLAVE_<ENDPOINT_INTERFACE_NAME>_AWAKEUP	AXI_SLAVE_<ENDPOINT_INTERFACE_NAME>_AWAKEUPCHK	Input	Indicates that the AXI or ACE-Lite completer interface has pending active transactions. This signal requests a clock for the NI-710AE.  AWAKEUP and AWAKEUPCHK signals are also present at Cortex-R52 and Cortex-R52+ interfaces. These inputs must be driven appropriately.	-
AXI_MASTER_<ENDPOINT_INTERFACE_NAME>_WID	AXI_MASTER_<ENDPOINT_INTERFACE_NAME>_WIDCHK	Output	Same cycle W-channel ID signal on AXI interface. Only applicable for AXI3.	-

## A.8 Clock and reset signals

The NI-710AE provides protection on the clock and reset signals.

### Signal definitions

**Table A-61: Clock and reset signals and check signals**

Signal	Check signal	Direction	Protection policy	Timing	Description
<CLKNAME>_CLK	<CLKNAME>_CLKCHK	Input	The clock input for that clock domain.		
<CLKNAME>_RESETn	<CLKNAME>_RESETCHKn	Input	Reset signal that is associated with the clock domain. Active-LOW.		
<CLKNAME>_AON_CLK	<CLKNAME>_AON_CLKCHK	Input	Feeds the HSNI buffer stage and must be on before the initial transaction ingresses into the device so the transaction is not lost.		



Signal	Check signal	Direction	Protection policy	Timing	Description
<CLKNAME>_AON_RESETh	<CLKNAME>_AON_RESECHKn	Input	The reset signal that feeds the HSNI buffer stage.		

## A.9 Clock management signals

NI-710AE provides protection on the clock management signals.

### Signal definitions

**Table A-62: Clock management signals and check signals**

Signal	Check signal	Direction	Description	Connection information
<CLKNAME>_QREQn	<CLKNAME>_QREQCHKn	Input	Request to disable the <CLKNAME>_CLK input. Active-LOW.	
<CLKNAME>_QACCEPTn	<CLKNAME>_QACCEPTCHKn	Output	Clock disable acceptance response. Active-LOW.	
<CLKNAME>_QDENY	<CLKNAME>_QDENYCHK	Output	Clock disable denial response.	
<CLKNAME>_QACTIVE	<CLKNAME>_QACTIVECHK	Output	Indicates that the NI-710AE requires the <CLKNAME>_CLK input to run.	

## A.10 Power management signals

NI-710AE provides protection on the power management signals.

### Signal definitions

**Table A-63: Power management signals and check signals**

Signal	Check signal	Direction	Description	Connection information
<PDOMAIN>_PREQ	<PDOMAIN>_PREQCHK	Input	Request to change power state for power domain <PDOMAIN>.	
<PDOMAIN>_PSTATE[4:0]	<PDOMAIN>_PSTATECHK	Input	Required power state.	
<PDOMAIN>_PACCEPT	<PDOMAIN>_PACCEPTCHK	Output	Power state transition acceptance.	
<PDOMAIN>_PDENY	<PDOMAIN>_PDENYCHK	Output	Power state transition denial.	
<PDOMAIN>_PACTIVE[x]	<PDOMAIN>_PACTIVECHK[x]	Output	Indicates the available power states for the NI-710AE.	
<PDOMAIN>_INTERRUPT	<PDOMAIN>_INTERRUPTCHK	Output	Secure interrupt for each power domain that is used to indicate specific conditions (IDM or non-IDM) within upstream or downstream interface. See the <a href="#">Programmers model</a> for the conditions.	
<PDOMAIN>_NS_INTERRUPT	<PDOMAIN>_NS_INTERRUPTCHK	Output	Non-secure interrupt for each power domain that is used to indicate specific conditions (IDM or non-IDM) within upstream or downstream interface. See the <a href="#">Programmers model</a> for the conditions.	

## A.11 IDM interface signals

NI-710AE provides protection on the IDM soft reset interface signals.

In this section:

- <INTF> represents <ENDPOINT\_INTERFACE\_NAME>
- <INST> represents the instance number of the APB peripheral

### Signal definitions

**Table A-64: IDM soft reset interface signals and check signals**

Signal	Check signal	Direction	Description	Connection information
<PROTOCOL>_MASTER_<INTF>_SRESETN	<PROTOCOL>_MASTER_<INTF>_SRESETNCHK	Output	External IDM soft reset	
<PROTOCOL>_SLAVE_<INTF>_SRESETN	<PROTOCOL>_SLAVE_<INTF>_SRESETNCHK	Output	External IDM soft reset	
APB_MASTER_<INST>_SRESETN	APB_MASTER_<INST>_SRESETNCHK	Output	External IDM soft reset	
<PROTOCOL>_MASTER_<INTF>_IDM_EXIT_SRESET_ENTRY_ACK	<PROTOCOL>_MASTER_<INTF>_IDM_EXT_SRESET_ENTRY_REQCHK	Input	External IDM soft reset	
<PROTOCOL>_MASTER_<INTF>_IDM_EXT_SRESET_ENTRY_ACK	<PROTOCOL>_MASTER_<INTF>_IDM_EXT_SRESET_ENTRY_ACKCHK	Output	External IDM soft reset	

## A.12 Interrupt signals

NI-710AE provides protection on the interrupt signals.

In this section, <PDOMAIN> represents the power domain.

### Signal definitions

**Table A-65: Interrupt signals and check signals**

Signal	Check signal	Direction	Description	Connection information
<PDOMAIN>_INTERRUPT	<PDOMAIN>_INTERRUPTCHK	Output	Secure interrupt for each power domain.  This interrupt is used to indicate specific conditions (IDM or non-IDM) within a requester or completer interface.  This interrupt is rising edge triggered.	

Signal	Check signal	Direction	Description	Connection information
<PDOMAIN>_NS_INTERRUPT	<PDOMAIN>_NS_INTERRUPTCHK	Output	<p>Non-secure interrupt for each power domain.</p> <p>This interrupt is used to indicate specific conditions (IDM or non-IDM) within a requester or completer interface.</p> <p>This interrupt is rising edge triggered.</p>	

## A.13 Configuration strap signals

NI-710AE provides protection on the configuration strap signals.

In this section:

- <INTF> represents <ENDPOINT\_INTERFACE\_NAME>
- <INST> represents the instance number of the APB peripheral

### Signal definitions

**Table A-66: Configuration strap signals and check signals**

Signal	Check signal	Direction	Description	Connection information
APB_MASTER_<INST>_IDM_SRESET_STRAP	APB_MASTER_<INST>_IDM_SRESET_STRAPCHK	Input	External IDM soft reset strap	
<PROTOCOL>_MASTER_<INTF>_IDM_SRESET_STRAP	<PROTOCOL>_MASTER_<INTF>_IDM_SRESET_STRAPCHK	Input	External IDM soft reset strap	
<PROTOCOL>_SLAVE_<INTF>_IDM_SRESET_STRAP	<PROTOCOL>_SLAVE_<INTF>_IDM_SRESET_STRAPCHK	Input	External IDM soft reset strap	
ECOREVNUM	ECOREVNUMCHK	Input	Same width as ECOREVNUM	
<prefix>_QOSOVERRIDE	<prefix>_QOSOVERRIDECHK	Input	Sample at reset QoS override	
<prefix>_ORDERED_WRITE_OBSERVATION	<prefix>_ORDERED_WRITE_OBSERVATIONCHK	Input	Enables Ordered Write Observation (OWO) on this completer interface if asserted	
<INTF>_CONFIG_ACCESS	<INTF>_CONFIG_ACCESSCHK	Input	Sample-at-reset input pin for each completer interface to indicate the completer interfaces that are permitted to accept new transactions in the CONFIG power state.	

## A.14 DFT interface signals

NI-710AE provides protection for DFT signals.



Each clock domain in a NI-710AE configuration is assigned a separate bit of DFT<CLKNAME>CLKDISABLE.

### Signal definitions

**Table A-67: Design for Test signals**

Signal	Check signal	Direction	Description	Connection information
DFTRSTDISABLE[1:0]	DFTRSTDISABLECHK	Input	Disables internal resets during scan operation.	
DFTCGEN	DFTCGENCHK	Input	Enables architectural clock gates for CLKNAME clocks. Assert HIGH during scan shift.	
DFT<CLKNAME>CLKDISABLE	DFT<CLKNAME>CLKDISABLECHK	Input	Disable clock.	

## A.15 Debug and Performance Monitoring Unit interface signals

In NI-710AE each clock domain can count, export, and report performance monitoring events. Each clock domain can report either Secure or Non-secure events.

In the following section, <CLKNAME> represents the name of the clock domain.



There are no functional safety check signals for the Performance Monitoring Unit (PMU).

### Signal definitions

**Table A-68: PMU and debug signals**

Signal	Direction	Description
<CLKNAME>_NIDEN	Input	Non-invasive debug enable. If HIGH, the signal enables counting and export of PMU events.
<CLKNAME>_SPNIDEN	Input	Secure privileged non-invasive debug enable. When HIGH, this signal enables the counting of both Non-secure and Secure events, provided NIDEN is also HIGH.
<CLKNAME>_DBGEN	Input	Invasive debug enable. If HIGH, enables the counting and export of PMU events.
<CLKNAME>_SPIDEN	Input	Secure privileged invasive debug enable. When HIGH, this signal enables the counting of both Non-secure and Secure events, provided that DBGEN is also HIGH.

Signal	Direction	Description
<CLKNAME>_PMUSNAPSHOTREQ	Input	Four-phase request to initiate snapshot of PMU counters.
<CLKNAME>_PMUSNAPSHOTACK	Output	Acknowledgment of PMU snapshot capture.
<CLKNAME>_nPMUIINTERRUPT	Output	Active-LOW level-sensitive interrupt to indicate a counter, event, or cycle has overflowed.

## A.16 Fault Management Unit interface signals

NI-710AE provides protection on the Fault Management Unit (FMU). The following table shows the FMU signals and their respective check signals.

### Signal definitions

**Table A-69: FMU signals and relevant check signals**

Signal	Check signal	Direction	Description	Connection information
FMU_PADDR	FMU_PADDRCHK	Input	APB address bus	-
FMU_PSEL	FMU_PSELCHK	Input	APB completer device select	-
FMU_PENABLE	FMU_PENABLECHK	Input	Enable.  This signal indicates the second and subsequent cycle of an APB transfer.	-
FMU_PWRITE	FMU_PCTRLCHK	Input	This signal indicates an APB read or write access.  <b>0b0</b> Read access  <b>0b1</b> Write access	-
FMU_PWDATA	FMU_PWDATACHK	Input	Write data	-
FMU_PPROT	FMU_PCTRLCHK	Input	Secure or Non-secure access	-
FMU_PSTRB	FMU_PSTRBCHK	Output	APB write data strobes	-
FMU_PRDATA	FMU_PRDATACHK	Output	APB read data	-
FMU_PRREADY	FMU_PRREADYCHK	Input	Ready.  The APB completer uses this signal to extend an APB transfer (wait states).	-
FMU_PSLVERR	FMU_PSLVERRCHK	Output	This signal indicates a transfer failure. APB peripherals are not required to support the PSLVERR pin. Where a peripheral does not include this pin then the appropriate input to PMNI is tied LOW.	-
FMU_FHI_INT	FMU_FHICLK	Output	Fault handling interrupt	-
FMU_ERI_INT	FMU_ERICLK	Output	Error recovery interrupt	-
FMU_CRI_INT	FMU_CRICLK	Output	Critical error interrupt	-

Signal	Check signal	Direction	Description	Connection information
FMU_LOCKSTRAP	FMU_LOCKSTRAPCHK	Input	Strap input for FMU lock and key mechanism.  <b>0b0</b> Lock and key mechanism is not enforced.  <b>0b1</b> Lock and key mechanism is enforced.	-
FMU_RESETh	FMU_RESEThCHK	Input	FMU reset to clear FMU error records	-

## A.17 Access Protection Unit interface signals

NI-710AE provides protection on the Access Protection Unit (APU) interface signals. The following table shows the APU signals and their respective check signals.

In this section:

- <INTF> represents <ENDPOINT\_INTERFACE\_NAME>
- <INST> represents the instance number of the APB peripheral

### Signal definitions

**Table A-70: APU signals and check signals**

Signal	Check signal	Direction	Description	Connection information
<PROTOCOL>_SLAVE_<INTF>_APUID	<PROTOCOL>_SLAVE_<INTF>_APUIDCHK	Input	Sideband signal for APU ID (sub systemID or completer ID)	
<PROTOCOL>_MASTER_<INTF>_APUID	<PROTOCOL>_MASTER_<INTF>_APUIDCHK	Output	Sideband signal for APU ID (sub systemID or completer ID)	
<PROTOCOL>_MASTER/ SLAVE_<INTF>_APU_ENABLE_RESET_STRAP	<PROTOCOL>_MASTER/ SLAVE_<INTF>_APU_ENABLE_RESET_STRAPCHK	Input	The reset value for apu_enable.	

# Appendix B Revisions

This appendix describes the technical changes between released issues of this manual.

**Table B-1: Issue 0000-01**

Change	Location
First Confidential release for r0p0	–

**Table B-2: Differences between issue 0000-01 and issue 0000-02**

Change	Location
Reorganized Clock and reset protection section and removed redundant information	Clock protection Reset protection
Reorganized Fault Management Unit section, removed redundant information, and added new topics: <ul style="list-style-type: none"> <li>FMU structure</li> <li>FMU error logging process</li> <li>Controlling which safety mechanisms report errors</li> <li>FMU interrupts</li> <li>FMU resets</li> </ul>	Fault Management Unit

**Table B-3: Differences between issue 0000-02 and issue 0001-03**

Change	Location
Moved FuSa parameters topic into <a href="#">Configurable options</a> section.	FuSa parameters
Corrected default value of <code>apuAddrRegions</code> to 8. The previous release incorrectly stated that the default value was 32.	
Added <code>idmWireInterface</code> to list of parameters that must be disabled when <code>dlsLogicProtection</code> is disabled.	
Added recommendation that Ordered Write Observation property should be used for ASNIs that are connected to PCIe requesters	ASNI configuration options
Removed Ordered Write Observation option from the list of AMNI configuration options, because this option is only supported on ASNIs	AMNI configuration options
Removed incorrect HMNI configuration option to enable Secure access support	HMNI configuration options
Described required actions to reprogram the APU address region registers after they have been set	Order of programming for APU address region registers
Added warning stating that the FMU_SMEN register must not be reprogrammed at runtime	Controlling which safety mechanisms report errors to the FMU
Corrected names of FMU interrupt signals	FMU interrupts Fault Management Unit interface signals

Change	Location
Update description of power domain OFF→ON transition to clarify that a Cold reset is required and add details of how to achieve it	Power control sequences
Extended power domain setup requirements for AHB address phase buffering in HSNI. To improve power saving, you must include the AHB requester in the same power domain as the HSNI buffer, and HSNI core. Previously, the requirement was that only the HSNI buffer and HSNI core must be in the same power domain.	AHB address phase buffering in HSNI
Added description of how to calculate NI-710AE node ID and interface ID values	Node ID calculation  Interface ID calculation
Updated USER_REQ_WIDTH parameter range from 0–64 bits to 0–256 bits	User signals
Updated details describing using NI-710AE to transport data parity, Error Correcting Code (ECC), and poison information, to refer to the relevant tooling option and value	Support for transporting data parity, ECC, and poison information
Corrected statement about supported read data reorder buffer sizes. NI-710AE read data reorder buffers can be configured at 1–256 data beats, but the previous release incorrectly stated that the available range was 1–64.	Transaction reorder buffers
Corrected statement about subfeature configuration node sizes. NI-710AE only supports 4KB configuration nodes, but the previous release incorrectly stated that there was an option for 64KB subfeature configuration nodes.	Subfeature configuration register region
Updated reset values of Global peripheral_id2 register, peripheral_id2.revision field, FMU FMU_ERRIIDR register, FMU_ERRIIDR.Variant field, FMU FMU_ERRPIDR2 register, and FMU_ERRPIDR2.Revision field	Global peripheral_id2 register  FMU FMU_ERRIIDR register  FMU FMU_ERRPIDR2 register
Corrected position of read_data_aggregation_enable field in the AMNI node_info register to bit[10], with bit[9] now reserved. The previous release incorrectly showed the read_data_aggregation_enable field at bit[9] in the AMNI node_info register.	AMNI node_info register
Updated description of HRDATACHK to state that the signal must be driven with the correct parity value corresponding to HRDATA even when the responder is in wait state	HMNI AHB-Lite response signals
Added information about AWAKEUP and AWAKEUPCHK input signals when integrating with Cortex-R52 and Cortex-R52+ processors	Miscellaneous AXI interface signals
Added FMU_CRI_INT signal and corrected direction of FMU_FHI_INT and FMU_ERI_INT signals to Output. The previous release incorrectly stated that these signals are Inputs.	Fault Management Unit interface signals
Removed Stuck at faults section	–

**Table B-4: Differences between issue 0001-03 and issue 0001-04**

Change	Location
Added requirement to lock APU registers to prevent modification due to faults	Unlocked and locked APU address regions
Removed reference to the FMU_ERR<n>_ADDR register, which is not present in NI-710AE	FMU error record table
Revised warning not to disable error reporting for safety mechanisms	Controlling which safety mechanisms report errors to the FMU



Change	Location
Added formula to calculate address offsets for registers that are repeated for each entry in the FMU error record table	FMU register summary
	FMU FMU_ERR_STATUS register
	FMU FMU_ERR_MISCO register
	FMU FMU_ERR_FR register
	FMU FMU_ERR_CTLR register
Corrected descriptions of FMU_ERR_MISCO and FMU_ERRDEVID registers	FMU register summary
	FMU FMU_ERR_MISCO register
	FMU FMU_ERRDEVID register