

# A64FX™ specification

Fujitsu HPC Extension

Ver. 1

November 30, 2020

Fujitsu Limited



---

Copyright© 2009 – 2020 Fujitsu Limited, 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan. All rights reserved.

This product and related documentation are protected by copyright and distributed under licenses restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Fujitsu Limited and its licensors, if any.

The product(s) described in this book may be protected by one or more U.S. patents, foreign patents, or pending applications.



---

# Revision History

---

Date	Contents	Page
November 30, 2020	Ver.1 released	

---

# Contents

---

<b>1. Fujitsu HPC extensions.....</b>	<b>2</b>
1.1. HPC tag address override.....	2
1.1.1. Overview.....	2
1.1.2. Enabling of HPC tag address override function .....	3
1.1.3. Behavior of Sector cache and Hardware prefetch assistance while HPC tag address override function is disabled .....	3
Sector cache .....	4
Hardware prefetch assistance .....	4
1.1.4. System Register Description .....	4
1.1.4.1.1. IMP_FJ_TAG_ADDRESS_CTRL_EL1 .....	4
1.1.4.2. IMP_FJ_TAG_ADDRESS_CTRL_EL2 .....	5
1.1.4.3. IMP_FJ_TAG_ADDRESS_CTRL_EL3 .....	7
1.1.4.4. IMP_FJ_TAG_ADDRESS_CTRL_EL12 .....	8
1.1.5. Tag address allocation .....	9
1.2. Sector cache .....	10
1.2.1. Overview .....	10
1.2.2. Allocation of HPC tag address override function .....	10
1.2.3. System Register Description .....	11
1.2.3.1. IMP_SCCR_CTRL_EL1 .....	12
1.2.3.2. IMP_SCCR_ASSIGN_EL1 .....	13
1.2.3.3. IMP_SCCR_L1_EL0 .....	14
1.2.3.4. IMP_SCCR_SET0_L2_EL1 .....	15
1.2.3.5. IMP_SCCR_SET1_L2_EL1 .....	16
1.2.3.6. IMP_SCCR_VSCCR_L2_EL0 .....	17
1.3. Hardware prefetch assistance .....	19
1.3.1. Overview .....	19
1.3.2. Hardware prefetch operation mode .....	19
1.3.2.1. Stream detect mode .....	19
1.3.2.2. Prefetch injection mode.....	20
PFQ_ALLOCATE mode.....	21
PFQ_UNALLOCATE mode .....	21
1.3.3. Allocation of HPC tag address override function .....	21
1.3.4. System Register Description .....	24
1.3.4.1. IMP_PF_CTRL_EL1 .....	25
1.3.4.2. IMP_PF_STREAM_DETECT_CTRL_EL0 .....	26
1.3.4.3. IMP_PF_INJECTION_CTRL[0-7]_EL0 .....	28
1.3.4.4. IMP_PF_INJECTION_DISTANCE[0-7]_EL0 .....	29
1.4. Hardware barrier .....	30
1.4.1. Overview .....	30
1.4.2. Compatibility Note.....	30
1.4.3. System Register Description .....	30
1.4.3.1. IMP_BARRIER_CTRL_EL1 .....	32
1.4.3.2. IMP_BARRIER_INIT_SYNC_BB[0-5]_EL1 .....	33
1.4.3.3. IMP_BARRIER_ASSIGN_SYNC_W[0-3]_EL1 .....	35
1.4.3.4. IMP_BARRIER_BST_BIT_EL1 .....	36
1.4.3.5. IMP_BARRIER_BST_SYNC_W[0-3]_EL0(Write) IMP_BARRIER_LBSY_SYNC_W[0-3]_EL0(Read).....	37
Usage example .....	38

---

# Preface

---

The purpose of this manual is to explain the A64FX processor microarchitecture and provide reference information for software tuning.

The manual was written with reference to the following documents. They define terms used in this manual without any particular annotations.

- *ARM® Architecture Reference Manual* (ARMv8, ARMv8.1, ARMv8.2, ARMv8.3)
- *ARM® Architecture Reference Manual Supplement The Scalable Vector Extension*

---

# 1. Fujitsu HPC extensions

---

This chapter explains the proprietary Fujitsu's HPC extensions.

The HPC extensions are implemented on the A64FX processor. These extensions can be used to achieve high performance and high efficiency in the HPC applications. These features have been succeeded and enhanced from SPARC64 VIIIfx, IXfx and XIfx processors which Fujitsu designed.

There are five features of Fujitsu HPC extension that A64FX processor implemented.

HPC tag address override - Function to control sector cache and hardware prefetch function from applications

Sector cache - Cache partitioning function which can control virtual cache capacity for each characteristics of data such as temporal locality.

The hardware prefetch assistance - Function that software can provide the hint for hardware prefetch mechanism to reduce the penalty of the memory access. By providing the access pattern through that function in advance, hardware prefetch mechanism can access the complex pattern.

Hardware barrier - Function to support the synchronization between threads of software with hardware

The next paragraph explains each in detail.

## 1.1. HPC tag address override

An HPC tag address override function is supported in the A64FX processor. This function is used to control the sector cache and the hardware prefetch assistance. It is possible to specify the operation of the sector cache and the hardware prefetch assistance by using the HPC tag address override function for each memory access instruction. The tuning of the application becomes possible with this function.

---

**Compatibility Note** This function corresponds to the referential function of `sector_id` and `dis_hwpf` of the Load/Store/Prefetch instructions enhanced with XAR in SPARC64VIIIfx, SPARC64IXfx, and SPARC64XIfx. This has been achieved by enhancing the function and enhancing the mechanism of Tagged addressing of ARMv8-A in the A64FX processor.

---

### 1.1.1. Overview

The HPC tag address override function is implemented in the A64FX processor.

Upper eight bits of 64 bit address calculated by the memory address calculation of the load instruction, the store instruction, and the prefetch instruction are ignored when the HPC tag address override function is valid, and it is treated as a hint to change the hardware behavior. This mechanism is compatible with the Tagged addressing function defined by ARMv8-A arm, and the behavior is the same from instruction set architecture point of view..

With the HPC tag address override, the sector cache and the hardware prefetch assistance can be controlled. When HPC tag address override is disabled, each function is valid as Default operation. For more information, see 1.1.3.

HPC tag address override function is valid only if Tagged addressing defined by ARMv8-A is enabled. Therefore, if a software using ARMv8-A defined Tagged addressing enables HPC tag address override function, the behavior of hardware can be changed and the performance can be affected. Moreover, it is difficult to tune performance of the application with Armv8-A defined Tagged addressing, by using the HPC tag address override function.

## 1.1.2. Enabling of HPC tag address override function

In order to use HPC tag address override function, both Tagged addressing of ARMv8-A and HPC tag address override must be enabled for each region. Table 1-1 shows the correspondence of the setting of each region.

Table 1-1 Memory area, Tagged addressing, and table for HPC tag address override setting bit

Region	Tagged addressing	HPC tag address override
TTBR0_EL1	TCR_EL1.TBI0	IMP_FJ_TAG_ADDRESS_CTRL_EL1.TBO0
TTBR1_EL1	TCR_EL1.TBI1	IMP_FJ_TAG_ADDRESS_CTRL_EL1.TBO1
TTBR0_EL2	TCR_EL2.TBI	IMP_FJ_TAG_ADDRESS_CTRL_EL2.TBO0
TTBR0_EL3	TCR_EL3.TBI	IMP_FJ_TAG_ADDRESS_CTRL_EL3.TBO0

Table 1-2 shows the addressing behaviors by the combination of TBI and TBO settings.

Table 1-2 TBI/TBO setting and operation to area

TBI	TBO	Behavior
0	-	Tagged addressing and the HPC tag address override function become invalid and all the data of 64 bits obtained by the address computation is used as an address.
1	0	Tagged addressing becomes valid, and upper 8 bits of the data of 64 bits obtained by the address computation is not used as an address.
1	1	The HPC tag address override function becomes valid, and upper 8 bits of the data of 64 bits obtained by the address computation is not used as an address.

In addition to TBI and TBO, SCE (Sector Cache Enable) and PFE (hardware PreFetch Enable) that can specify Enable/Disable of each function exist about the setting of the HPC tag address override.

## 1.1.3. Behavior of Sector cache and Hardware prefetch assistance while HPC tag address override function is disabled

The sector and the hardware prefetch assistance become Default operation when either Tagged addressing or the HPC tag address override of ARMv8-A is invalid (When either of

TBI or TBO is 0). Moreover, when SCE and PFE are 0 in case of TBI=1 and TBO=1, the sector or the hardware prefetch assistance becomes Default operation.

The Default operation is brought together as follows.

## Sector cache

It operates as Default Sector specified by another register.

## Hardware prefetch assistance

It always operates as Stream detect mode. The operation of Stream detect mode is specified by IMP\_PF\_STREAM\_DETECT\_CTRL\_EL0.

## 1.1.4. System Register Description

The related registers are defined in IMPLEMENTATION DEFINED region of the system register (S3\_<op1>\_<Cn>\_<Cm>\_<op2>).

Table 1-37 shows the System Register list used for by the HPC tag address override function.

Table 1-3 HPC tag address override function System Register list

op0	op1	CRn	CRm	op2	Register Name	Shared Domain
11	000	1011	0010	000	IMP_FJ_TAG_ADDRESS_CTRL_EL1	PE
11	100	1011	0010	000	IMP_FJ_TAG_ADDRESS_CTRL_EL2	PE
11	110	1011	0010	000	IMP_FJ_TAG_ADDRESS_CTRL_EL3	PE
11	101	1011	0010	000	IMP_FJ_TAG_ADDRESS_CTRL_EL12	PE

ARMv8.1 virtualization host extension influences the IMP\_FJ\_TAG\_ADDRESS\_CTRL register.

If HCR\_EL2.E2H = 1 and SCR\_EL3.NS = 1, operation is changed as follows:

- PFE1, SCE1, and the TBO1 bit are added to IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL2, and the control to the TTBR1\_EL2 region is added.
- The access to IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL1 is changed to the access to IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL2 at CurrentEL=2.
- Alias is done by IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL1 as for the access to IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL12 at CurrentEL=2 and 3.

### 1.1.4.1.1. IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL1

Purpose HPC tag address override control register

Usage constraints IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	RW	RW	RW	RW	RW

Configuration This register is 32 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

31	14	13	12	11	10	9	8	7	2	1	0
RES0	PFE1	SCE1	RES0	PFE0	SCE0	RES0	TBO1	TBO0			

Bits	Name	Function
[31:14]	-	Reserved, RES0
[13]	PFE1	Prefetch assist enable 1: When it accesses the TTBR1_EL1 area, the pf_func instruction of the tagged address area is valid. 0: Prefetch assist operates as Default when it accesses the TTBR1_EL1 area.
[12]	SCE1	Sector cache enable 1: When it accesses the TTBR1_EL1 area, the sector_id instruction of the tagged address area is valid. 0: Sector cache operates as Default when it accesses the TTBR1_EL1 area.
[11:10]	-	Reserved, RES0
[9]	PFE0	Prefetch assist enable 1: When it accesses the TTBR0_EL1 area, the pf_func instruction of the tagged address area is valid. 0: Prefetch assist operates as Default when it accesses the TTBR0_EL1 area.
[8]	SCE0	Sector cache enable 1: When it accesses the TTBR0_EL1 area, the sector_id instruction of the tagged address area is valid. 0: Sector cache operates as Default when it accesses the TTBR0_EL1 area.
[7:2]	-	Reserved, RES0
[1]	TBO1	Top Byte override TTBR1 Upper 1-byte of the address of the access to the TTBR1_EL1 area is not used as an address at TBO1 = 1 and TCR_EL1.TBI1 = 1, and it is used as HPC tag address override function. The function of the HPC tag address override is invalid at TBO1 = 0.
[0]	TBO0	Top Byte override TTBR0 Upper 1-byte of the address of the access to the TTBR0_EL1 area is not used as an address at TBO0 = 1 and TCR_EL1.TBI0 = 1, and it is used as HPC tag address override function. The function of the HPC tag address override is invalid at TBO0 = 0.

Accessing MRS <Xt>, S3\_0\_C11\_C2\_0  
MSR S3\_0\_C11\_C2\_0, <Xt>

op0	op1	CRn	CRm	op2
11	000	1011	0010	000

#### 1.1.4.2. IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL2

Purpose HPC tag address override control register

Usage constraints IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL2 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
			RW	RW	RW

Configuration This register is 32 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

HCR\_EL2.E2H = 0

31	10	9	8	7	1	0
RES0	PFE0	SCE0	RES0	TBO0		

HCR\_EL2.E2H = 1

31	14	13	12	11	10	9	8	7	2	1	0
RES0	PFE1	SCE1	RES0	PFE0	SCE0	RES0	TBO1	TBO0			

HCR\_EL2.E2H = 0

Bits	Name	Function
[31:10]	-	Reserved, RES0
[9]	PFE0	Prefetch assist enable 1: When it accesses the TTBR0_EL2 area, the pf_func instruction of the tagged address area is valid. 0: Prefetch assist operates as Default when it accesses the TTBR0_EL2 area.
[8]	SCE0	Sector cache enable 1: When it accesses the TTBR0_EL2 area, the sector_id instruction of the tagged address area is valid. 0: Sector cache operates as Default when it accesses the TTBR0_EL2 area.
[7:1]	-	Reserved, RES0
[0]	TBO0	Top Byte override TTBR0 Upper 1-byte of the address of the access to the TTBR0_EL2 area is not used as an address at TBO0 = 1 and TCR_EL2.TBI = 1, and it is used as HPC tag address override function. The function of the HPC tag address override is invalid at TBO0 = 0.

HCR\_EL2.E2H = 1

Bits	Name	Function
[31:14]	-	Reserved, RES0
[13]	PFE1	Prefetch assist enable 1: When it accesses the TTBR1_EL2 area, the pf_func instruction of the tagged address area is valid. 0: Prefetch assist operates as Default when it accesses the TTBR1_EL2 area.
[12]	SCE1	Sector cache enable 1: When it accesses the TTBR1_EL2 area, the sector_id instruction of the tagged address area is valid. 0: Sector cache operates as Default when it accesses the TTBR1_EL2 area.
[11:10]	-	Reserved, RES0

[9]	PFE0	Prefetch assist enable 1: When it accesses the TTBR0_EL2 area, the pf_func instruction of the tagged address area is valid. 0: Prefetch assist operates as Default when it accesses the TTBR0_EL2 area.
[8]	SCE0	Sector cache enable 1: When it accesses the TTBR0_EL2 area, the sector_id instruction of the tagged address area is valid. 0: Sector cache operates as Default when it accesses the TTBR0_EL2 area.
[7:2]	-	Reserved, RES0
[1]	TBO1	Top Byte override TTBR1 Upper 1-byte of the address of the access to the TTBR1_EL2 area is not used as an address at TBO1 = 1 and TCR_EL2.TBI1 = 1, and it is used as HPC tag address override function. The function of the HPC tag address override is invalid at TBO0 = 1.
[0]	TBO0	Top Byte override TTBR0 Upper 1-byte of the address of the access to the TTBR0_EL2 area is not used as an address at TBO0 = 1 and TCR_EL2.TBI0 = 1, and it is used as HPC tag address override function. The function of the HPC tag address override is invalid at TBO0 = 0.

Accessing MRS <Xt>, S3\_4\_C11\_C2\_0  
MSR S3\_4\_C11\_C2\_0, <Xt>

op0	op1	CRn	CRm	op2
11	100	1011	0010	000

#### 1.1.4.3. IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL3

Purpose HPC tag address override control register

Usage constraints IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL3 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
				RW	RW

Configuration This register is 32 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

31	10	9	8	7	1	0
RES0	PFE0	SCE0	RES0	TBO0		

Bits	Name	Function
[31:10]	-	Reserved, RES0
[9]	PFE0	Prefetch assist enable 1: When it accesses the TTBR0_EL3 area, the pf_func instruction of the tagged address area is valid. 0: Prefetch assist operates as Default when it accesses the TTBR0_EL3 area.
[8]	SCE0	Sector cache enable 1: When it accesses the TTBR0_EL3 area, the sector_id instruction of the tagged address area is valid. 0: Sector cache operates as Default when it accesses the TTBR0_EL3 area.
[7:1]	-	Reserved, RES0
[0]	TBO0	Top Byte override TTBR0 Upper 1-byte of the address of the access to the TTBR0_EL3 area is not used as an address at TBO0 = 1 and TCR_EL3.TBI = 1, and it is used as HPC tag address override function. The function of the HPC tag address override is invalid at TBO0 = 0.

Accessing MRS <Xt>, S3\_6\_C11\_C2\_0  
MSR S3\_6\_C11\_C2\_0, <Xt>

op0	op1	CRn	CRm	op2
11	110	1011	0010	000

#### 1.1.4.4. IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL12

Purpose HPC tag address override control register

Usage constraints IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL12 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
			RW	RW	

Configuration This register is 32 bits wide. When HCR\_EL2.E2H = 1 and SCR\_EL3.NS = 1, this register can operate as the alias register of IMP\_FJ\_TAG\_ADDRESS\_CTRL\_EL1. When HCR\_EL2.E2H = 0 or SCR\_EL3.NS = 0, the access from all EL generates UNDEFINED.

Attributes The allocation of the register and implementation in the A64FX core are shown.

Accessing MRS <Xt>, S3\_5\_C11\_C2\_0  
MSR S3\_5\_C11\_C2\_0, <Xt>

op0	op1	CRn	CRm	op2
11	101	1011	0010	000

---

## 1.1.5. Tag address allocation

When the HPC tag address override function is enabled, eight bits of the tag address (upper eight bits of the address) are treated as following.

63	60	59	58	57	56
pf_func		SBZ	SBZ	sector_id	

---

Bits	Name	Function
[63:60]	pf_func	When PFE=1: the operation of the hardware prefetch assistance is specified. Refer to details to the hardware prefetch assistance. When PFE=0: this field is ignored, and the hardware prefetch assistance operates as Default.
[59:58]	-	Software should set 0 in preparation for the future extension. However, even if non-0 value is set, the value is ignored.
[57:56]	sector_id	When SCE=1: sector ID of the sector cache is specified. Refer to details to the sector cache. When SCE=0: this field is ignored, and sector ID operates as Default.

---

This setting is valid only for the load instruction, the store instruction, and the prefetch instruction, and it does not influence the instruction fetch.

---

## 1.2. Sector cache

The A64FX processor has a mechanism that split cache into multiple sectors and control them separately. This mechanism is called sector cache.

### 1.2.1. Overview

The sector cache is a function to split cache into multiple sectors, and control them separately. The maximum capacity is specified for each sector respectively, and cache is controlled so that data in the sector is not evicted from cache as long as the quantity consumed of the sector is smaller than the maximum capacity. Because multiple sectors can be made in cache and the maximum capacity of each sector can be set independently of the maximum capacity of other sectors, the degree of freedom of the application is high.

The sector cache mechanism is implemented on the L1D cache and L2 cache in the A64FX processor, and sector cache function can be controlled individually for L1D cache and L2 cache. In the A64FX processor, there are sector cache controls for each L1D cache of PE and each L2 cache of CMG.

Each L1D cache and L2 cache has 4 sectors. The 4 sectors in L2 cache are divided into 2 parts. Each PE can use 2 sectors in L2 cache by selecting one from 2 parts.

---

**Note** It is recommended that all CMGs have common sector cache setting beforehand by the software when all 48 computing cores are used by one process.

**Note** It is recommended that the sector cache setting in the same CMG is the same between the processes when 12 cores in CMG are divided and used in two or more processes.

---

The sector is identified by the number. The sector number is added to the access of all the memories of the instruction fetch and the load/store instruction, etc. As for the sector number, software is able to specify the number expressly, or implicit number is used when not specifying it.

If there is no data to be used in cache, it is read from the memory, and it is stored in the cache with the sector number. At this time, the data to be evicted from the cache is selected taking into consideration of the capacity of the sector, when the sector cache is valid.

On the other hand, when data to be used is in cache, the data is read or updated. At this time, data on all sectors can be used regardless of the sector number added.

### 1.2.2. Allocation of HPC tag address override function

In the A64FX processor implementation, the sector ID is given to the access of the cacheable memory. These sector IDs are decided by `IMP_SCCR_ASSIGN_EL1.assign` and `IMP_SCCR_ASSIGN_EL1.default_sector<1:0>` and `TagAddress.sector_id<1:0>`.

Table 74 Sector ID used by each cache hierarchy

		L1I/L1D cache	L2 cache
<b>Instruction access</b>		-	assign::default_sector<0>
<b>Data access</b>	<b>HPC Tagged override Access to invalid region</b>	default_sector<1:0>	assign::default_sector<0>
	<b>HPC Tagged override Access to valid region</b>	sector_id<1:0>	assign::sector_id<0>
<b>Programming Note</b> Sector cache of L1I cache is not applied. However, Sector is decided by request from the L1I cache to L2 cache using assign and default_sector<0>.			

### 1.2.3. System Register Description

All registers of the sector cache are defined in IMPLEMENTATION DEFINED region (S3\_<op1>\_<Cn>\_<Cm>\_<op2>).

Table 1-6 shows the list of all setting registers concerning the sector cache. All the registers are 64 bits wide.

Table 1-6 Sector cache register list

op0	op1	CRn	CRm	op2	Register Name	Shared domain
11	000	1011	1000	000	IMP_SCCR_CTRL_EL1	PE
11	000	1011	1000	001	IMP_SCCR_ASSIGN_EL1	PE
11	000	1111	1000	010	IMP_SCCR_SET0_L2_EL1	CMG
11	000	1111	1000	011	IMP_SCCR_SET1_L2_EL1	CMG
11	011	1011	1000	010	IMP_SCCR_L1_EL0	PE
11	011	1111	1000	010	IMP_SCCR_VSCCR_L2_EL0	PE(CMG) <sup>i</sup>

The access by EL1 and EL0 to a register of the sector cache is controlled from system register IMP\_SCCR\_CTRL\_EL1.

<sup>i</sup> The substance of the register to be updated is either IMP\_SCCR\_SET0\_L2\_EL1 or IMP\_SCCR\_SET1\_L2\_EL1 specified with IMP\_SCCR\_ASSIGN\_EL1, though the register is a resource of PE.

Table 1-7 Sector cache register access right

Register Name	el1ae=0			el1ae=1 and el0ae=0			el1ae=1 and el0ae=1		
	EL0	EL1 (NS)	EL1(S)	EL0	EL1 (NS)	EL1(S)	EL0	EL1 (NS)	EL1(S)
IMP_SCCR_CTRL_EL1		RO	RW		RW	RW		RW	RW
IMP_SCCR_ASSIGN_EL1			RW		RW	RW		RW	RW
IMP_SCCR_L1_EL0			RW		RW	RW	RW	RW	RW
IMP_SCCR_SET0_L2_EL1			RW		RW	RW		RW	RW
IMP_SCCR_SET1_L2_EL1			RW		RW	RW		RW	RW
IMP_SCCR_VSSCCR_L2_EL0			RW		RW	RW	RW	RW	RW

### 1.2.3.1. IMP\_SCCR\_CTRL\_EL1

Purpose Access control register for sector cache

Usage constraints IMP\_SCCR\_CTRL\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	RO/ Config- RW	RW	RW	RW	RW

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

63	62	61	32
el1ae	el0ae	RES0	

31	0
RES0	

Bits	Name	Value	Function
[63]	el1ae	RW	<p>1: Read/write from Non-Secure EL1 is enabled to IMP_SCCR_CTRL_EL1, IMP_SCCR_ASSIGN_EL1, IMP_SCCR_L1_EL0, IMP_SCCR_SET0_L2_EL1, IMP_SCCR_SET1_L2_EL1, and IMP_SCCR_VSCCR_L2_EL0.</p> <p>0: Read/write from Non-Secure EL1 to IMP_SCCR_ASSIGN_EL1, IMP_SCCR_L1_EL0, IMP_SCCR_SET0_L2_EL1, IMP_SCCR_SET1_L2_EL1, and IMP_SCCR_VSCCR_L2_EL0 is trapped to EL2 with EC=0x18. Moreover, Write from Non-Secure EL1 to IMP_SCCR_CTRL_EL1 is trapped to EL2 with EC=0x18.</p> <p>This bit is writable only from Secure EL1 and EL2/EL3. When the writing from Non-Secure EL1 at el1ae=1, the writing is ignored.</p>
[62]	el0ae	RW	<p>1: When el1ae=1, Read/Write from EL0 is enabled to IMP_SCCR_L1_EL0 and IMP_SCCR_VSCCR_L2_EL0. When el1ae=0, Access from EL0 to IMP_SCCR_L1_EL0 and IMP_SCCR_VSCCR_L2_EL0 is trapped to EL1 with EC=0x18.</p> <p>0: The access to IMP_SCCR_L1_EL0 and IMP_SCCR_VSCCR_L2_EL0 by EL0 is trapped to EL1 with EC=0x18.</p>
[61:0]	-	0x0	Reserved, RES0

Accessing MRS <Xt>, S3\_0\_C11\_C8\_0  
MSR S3\_0\_C11\_C8\_0, <Xt>

op0	op1	CRn	CRm	op2
11	000	1011	1000	000

### 1.2.3.2. IMP\_SCCR\_ASSIGN\_EL1

Purpose Sector cache allocation and operation control register

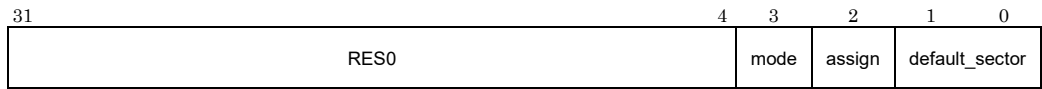
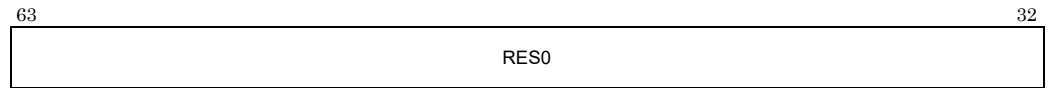
Usage constraints IMP\_SCCR\_ASSIGN\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	Config- RW	RW	RW	RW	RW

The access by EL1 is controlled by the IMP\_SCCR\_CTRL\_EL1.el1ae bit.

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63:4]	-	0x0	Reserved, RES0
[3]	mode	RW	0: Sector ID that the cache line maintains gets updated, when Sector ID that the accessed cache line maintains and access Sector ID are different. 1: Sector ID that the cache line maintains is kept, even if Sector ID that the accessed cache line maintains and access Sector ID are different.
[2]	assign	RW	An accessible register is selected by IMP_SCCR_VSCCR_L2_EL0. 0: IMP_SCCR_VSCCR_L2_EL0 becomes alias of IMP_SCCR_SET0_L2_EL1. 1: IMP_SCCR_VSCCR_L2_EL0 becomes alias of IMP_SCCR_SET1_L2_EL1.
[1:0]	default_sector	RW	Sector ID might not be specified by the instruction by some reasons. In this case, sector value written in the default_sector is used as Sector ID.

Accessing MRS <Xt>, S3\_0\_C11\_C8\_1  
MSR S3\_0\_C11\_C8\_1, <Xt>

op0	op1	CRn	CRm	op2
11	000	1011	1000	001

### 1.2.3.3. IMP\_SCCR\_L1\_EL0

Purpose L1 sector cache capacity setting register

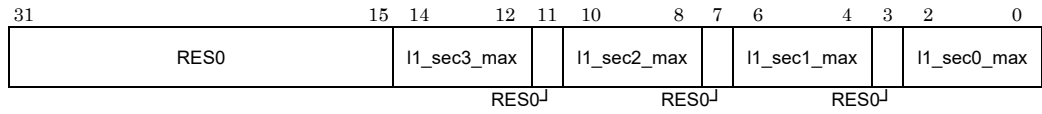
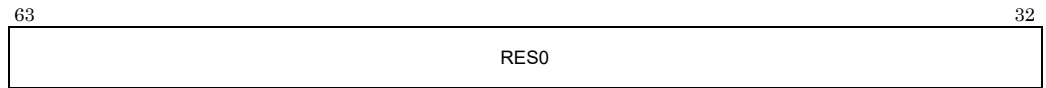
Usage constraints IMP\_SCCR\_L1\_EL0 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
Config-RW	Config-RW	RW	RW	RW	RW

The access by EL1 is controlled by the IMP\_SCCR\_CTRL\_EL1.el1ae bit. Moreover, the access by EL0 is controlled by the IMP\_SCCR\_CTRL\_EL1.el0ae bit.

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63:15]	-	0x0	Reserved, RES0
[14:12]	l1_sec3_max	RW	The number of L1 Sector ID=3 of maximum sectors is set.
[11]	-	0x0	Reserved, RES0
[10:8]	l1_sec2_max	RW	The number of L1 Sector ID=2 of maximum sectors is set.
[7]	-	0x0	Reserved, RES0
[6:4]	l1_sec1_max	RW	The number of L1 Sector ID=1 of maximum sectors is set.
[3]	-	0x0	Reserved, RES0
[2:0]	l1_sec0_max	RW	The number of L1 Sector ID=0 of maximum sectors is set.

Accessing MRS <Xt>, S3\_3\_C11\_C8\_2  
MSR S3\_3\_C11\_C8\_2, <Xt>

op0	op1	CRn	CRm	op2
11	011	1011	1000	010

#### 1.2.3.4. IMP\_SCCR\_SET0\_L2\_EL1

Purpose L2 sector cache maximum capacity setting register

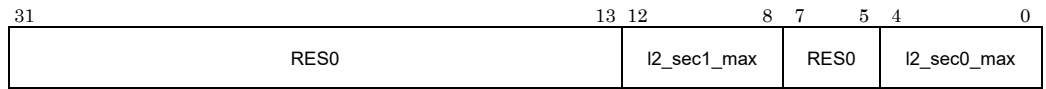
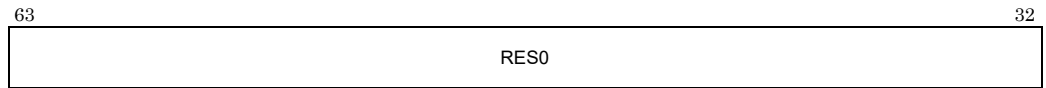
Usage constraints IMP\_SCCR\_SET0\_L2\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	Config- RW	RW	RW	RW	RW

The access by EL1 is controlled by the IMP\_SCCR\_CTRL\_EL1.el1ae bit.

Configuration This register is 64 bits wide. It is shared between two or more PE in CMG and it is to be noted that changing IMP\_SCCR\_SET0\_L2\_EL1 in one PE influences other PE.

Attributes The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63:13]	-	0x0	Reserved, RES0
[12:8]	l2_sec1_max	RW	The number of L2 Sector ID=1 of maximum sectors is set.
[7:5]	-	0x0	Reserved, RES0
[4:0]	l2_sec0_max	RW	The number of L2 Sector ID=0 of maximum sectors is set.

Accessing      MRS <Xt>, S3\_0\_C15\_C8\_2  
 MSR S3\_0\_C15\_C8\_2, <Xt>

op0	op1	CRn	CRm	op2
11	000	1111	1000	010

### 1.2.3.5. IMP\_SCCR\_SET1\_L2\_EL1

Purpose              L2 sector cache maximum capacity setting register

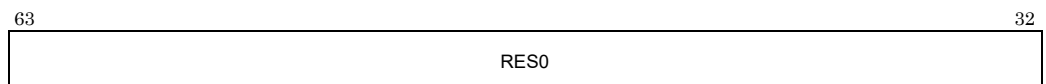
Usage constraints   IMP\_SCCR\_SET1\_L2\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	Config- RW	RW	RW	RW	RW

The access by EL1 is controlled by the IMP\_SCCR\_CTRL\_EL1.el1ae bit.

Configuration      This register is 64 bits wide. It is shared between two or more PE in CMG and it is to be noted that changing IMP\_SCCR\_SET1\_L2\_EL1 in one PE influences other PE.

Attributes           The allocation of the register and implementation in the A64FX core are shown.



31	13	12	8	7	5	4	0
RES0		l2_sec1_max		RES0		l2_sec0_max	

Bits	Name	Value	Function
[63:13]	-	0x0	Reserved, RES0
[12:8]	l2_sec1_max	RW	The number of L2 Sector ID=3 of maximum sectors is set.
[7:5]	-	0x0	Reserved, RES0
[4:0]	l2_sec0_max	RW	The number of L2 Sector ID=2 of maximum sectors is set.

Accessing MRS <Xt>, S3\_0\_C15\_C8\_3  
MSR S3\_0\_C15\_C8\_3, <Xt>

op0	op1	CRn	CRm	op2
11	000	1111	1000	011

#### 1.2.3.6. IMP\_SCCR\_VSCCR\_L2\_EL0

Purpose L2 sector cache capacity setting register

Usage constraints IMP\_SCCR\_L2\_VSCCR\_EL0 is accessible in following Exception Level.

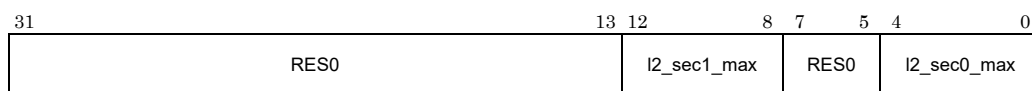
EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
Config-RW	Config-RW	RW	RW	RW	RW

The access by EL1 is controlled by the IMP\_SCCR\_CTRL\_EL1.el1ae bit. Moreover, the access by EL0 is controlled by the IMP\_SCCR\_CTRL\_EL1.el0ae bit.

Configuration This register is 64 bits wide. This register is a window register to enable the access to IMP\_SCCR\_L2\_SET{0|1}\_EL1 selected by IMP\_SCCR\_ASSIGN\_EL1.assign. When this register is updated, IMP\_SCCR\_L2\_SET{0|1}\_EL1 selected by IMP\_SCCR\_ASSIGN\_EL1.assign is updated. This register update influences between two or more PE in CMG as well as IMP\_SCCR\_SET{0|1}\_L2\_EL1.

Attributes The allocation of the register and implementation in the A64FX core are shown.

63	32
RES0	



Bits	Name	Value	Function
[63:13]	-	0x0	Reserved, RES0
[12:8]	l2_sec1_max	RW	The number of L2 Sector ID=1/3 of maximum sectors is set.
[7:5]	-	0x0	Reserved, RES0
[4:0]	l2_sec0_max	RW	The number of L2 Sector ID=0/2 of maximum sectors is set.

Accessing      MRS <Xt>, S3\_3\_C15\_C8\_2  
 MSR S3\_3\_C15\_C8\_2, <Xt>

op0	op1	CRn	CRm	op2
11	011	1111	1000	010

---

## 1.3. Hardware prefetch assistance

The hardware prefetch assistance is a mechanism that software controls the hardware prefetch mechanism installed in the A64FX core depending on the characteristic of the application. The goal is to hide the latency of the memory access by controlling an appropriate hardware prefetch through the HPC tag address override, setting the system registers corresponding to the characteristic of the application and telling the characteristic of each load/store to a hardware side.

### 1.3.1. Overview

The hardware prefetch engine is implemented on the A64FX processor. There are two kinds of operational modes of implemented hardware prefetches.

- Hardware prefetch based on automatic detection of continuous memory access
- Hardware prefetch based on information on a certain register

The hardware prefetch operates for the load instruction and the store instruction. It is to be noted that the hardware prefetch does not work for the prefetch instruction. Moreover, even for a load instruction and a store instruction, the hardware prefetch might not be generated depending on the hardware prefetch mode. Refer to A64FX Microarchitecture Manual for detailed hardware prefetch generation conditions.

Among these two kinds of hardware prefetch modes, Default is hardware prefetch based on automatic detection of continuous memory access, while it is possible to specify hardware prefetch mode control at each instruction with HPC tag address override function.

The hardware prefetch mechanism does not work for the prefetch instruction, and the prefetch is generated in the mode specified by the HPC tag address override function or the register. There are two modes for the prefetch instruction.

- Strong prefetch
- Weak prefetch

Strong prefetch always performs memory access if there is no corresponding data in cache. Weak prefetch also might perform memory access if there is no corresponding data in cache, but it might not perform memory access depending on internal hardware condition.

For instance, when the TLB miss is detected in Weak prefetch, Hardware Table walker does not start and the Prefetch operation is cancelled. In Strong prefetch, Hardware Table walker starts, and Prefetch operates as long as Fault etc. are not detected, when the TLB miss is detected.

When both Weak Prefetch and Strong Prefetch detect Fault on the way, Trap is not generated. (When Fault is detected, the corresponding Prefetch operation is cancelled.)

The operation control in the prefetch instruction is different depending on the operational mode of the specified hardware prefetch. The details are in 1.3.2. Hardware prefetch operation mode.

### 1.3.2. Hardware prefetch operation mode

It explains two kinds of operational modes of the hardware prefetch.

#### 1.3.2.1. Stream detect mode

In stream detect mode, the hardware prefetch works based on the automatic detection of hardware of a continuous memory access.

The hardware prefetch always operates as this mode when the HPC tag address override function is invalid.

---

**Compatibility Note** This hardware prefetch function is equivalent with the ones of SPARC64VIIIfx, SPARC64IXfx, and SPARC64XIfx.

---

This mode operates based on the setting of IMP\_PF\_STREAM\_DETECT\_CTRL\_EL0. Details are 1.3.4.2 IMP\_PF\_STREAM\_DETECT\_CTRL\_EL0.

When pf\_func=0x0 ~ 0x7 is specified when the HPC tag address override function is valid, it operates as stream detect mode. At this time, pf\_func[2:0] is interpreted in the meaning of Table 1-8.

Table 1-8 Allocation of pf\_func[2:0] at Stream detect mode

pf_func	name	description
bit[2]	DIS_L1HWPF	0: The L1 hardware prefetch is enabled. 1: The L1 hardware prefetch is disabled However, when IMP_PF_STREAM_DETECT_CTRL_EL0.L1PF_DIS = 1, the L1 hardware prefetch is disabled regardless of this bit. Also, this bit is ignored in the Prefetch instruction.
bit[1]	DIS_L2HWPF	0: The L2 hardware prefetch is enabled. 1: The L2 hardware prefetch is disabled However, when IMP_PF_STREAM_DETECT_CTRL_EL0.L2PF_DIS = 1, the L2 hardware prefetch is disabled regardless of this bit. Also, this bit is ignored in the Prefetch instruction.
bit[0]	SWPF_STRONG	0: The Prefetch instruction is generated with Strong prefetch hint. 1: The Prefetch instruction is generated with Weak prefetch hint. This bit is ignored except for the Prefetch instruction.

### 1.3.2.2. Prefetch injection mode

Prefetch injection mode is a mode, in which the hardware prefetch is controlled by two sets of eight prefetch injection control registers.

The setting is selected from the eight registers by using the HPC tag address override function. The two setting sets are IMP\_PF\_INJECTION\_CTRL[0-7]\_EL0, IMP\_PF\_INJECTION\_DISTANCE[0-7]\_EL0 respectively.

The stride prefetch can be generated by specifying the attribute, and the effective range is from 4B to 16MB. The Prefetch injection mode has two operational modes.

- PFQ\_ALLOCATE mode: Only when load/store causes the L1D cache miss and the specified condition is met, the hardware prefetch is generated.
- PFQ\_UNALLOCATE mode: The hardware prefetch is unconditionally generated when there is load/store access.

In each set, it is possible to specify the mode.

The Prefetch instruction operates in the mode specified by IMP\_PF\_INJECTION\_CTRL[0-7]\_EL0.SWW field when this mode is specified.

When IMP\_PF\_INJECTION\_CTRL[0-7]\_EL0.V=0 and corresponding prefetch injection mode is specified, L1 hardware prefetch and L2 hardware prefetch are not generated and prefetch instruction operates as Strong Prefetch.

---

## PFQ\_ALLOCATE mode

PFQ\_ALLOCATE mode is valid when the Prefetch Injection mode is IMP\_PF\_INJECTION\_CTRL[0-7]\_ELO.A=1(PFQ\_ALLOCATE) when the load/store is executed. When the address that the load/store instruction accessed meets the specific condition, hardware generates the hardware prefetch in the PFQ\_ALLOCATE mode.

## PFQ\_UNALLOCATE mode

PFQ\_UNALLOCATE mode is valid when the Prefetch Injection mode is IMP\_PF\_INJECTION\_CTRL[0-7]\_ELO.A=0(PFQ\_UNALLOCATE) when the load/store is executed. In the PFQ\_UNALLOCATE mode, hardware generates the prefetch of L1 and L2 to the address where the load/store instruction accessed plus the value in L1PF\_DISTANCE of IMP\_PF\_INJECTION\_DISTANCE[0-7]\_ELO and L2PF\_DISTANCE. Unlike the PFQ\_ALLOCATE mode, the prefetch is generated regardless of the presence of the L1D cache miss.

### 1.3.3. Allocation of HPC tag address override function

The hardware prefetch assistance function changes its behavior, depending on bit[63:60] (pf\_func) in the address where the load instruction and the store instruction were operated when the HPC tag address override function is valid. Table 1-9 shows the value of pf\_func specified by memory access instruction and corresponding hardware prefetch assistance function.

---

Table 1-9 Value of pf\_func when Load/Store/Prefetch is executed and operation of hardware prefetch

pf_func	mode	description
0x0	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is valid.</li> <li>• L2HWPF is valid.</li> <li>• Software Prefetch: Strong</li> </ul>	The continuous hardware prefetch is valid.
0x1	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is valid.</li> <li>• L2HWPF is valid.</li> <li>• Software Prefetch: Weak</li> </ul>	The continuous hardware prefetch is valid.
0x2	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is valid.</li> <li>• L2HWPF is invalid</li> <li>• Software Prefetch: Strong</li> </ul>	The continuous hardware prefetch is valid.
0x3	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is valid.</li> <li>• L2HWPF is invalid</li> <li>• Software Prefetch: Weak</li> </ul>	The continuous hardware prefetch is valid.
0x4	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is invalid.</li> <li>• L2HWPF is valid.</li> <li>• Software Prefetch: Strong</li> </ul>	The continuous hardware prefetch is valid.
0x5	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is invalid.</li> <li>• L2HWPF is valid.</li> <li>• Software Prefetch: Weak</li> </ul>	The continuous hardware prefetch is valid.
0x6	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is invalid.</li> <li>• L2HWPF is invalid</li> <li>• Software Prefetch: Strong</li> </ul>	The continuous hardware prefetch is invalid.
0x7	Stream detect mode <ul style="list-style-type: none"> <li>• L1HWPF is invalid.</li> <li>• L2HWPF is invalid.</li> <li>• Software Prefetch: Weak</li> </ul>	The continuous hardware prefetch is invalid
0x8	Prefetch injection 0	The 0th SystemRegister settings are used in the Prefetch Injection mode.
0x9	Prefetch injection 1	The first SystemRegister setting is used in the Prefetch Injection mode.
0xa	Prefetch injection 2	The second SystemRegister settings are used in the Prefetch Injection mode.
0xb	Prefetch injection 3	The third SystemRegister settings are used in the Prefetch Injection mode.
0xc	Prefetch injection 4	The fourth SystemRegister settings are used in the Prefetch Injection mode.
0xd	Prefetch injection 5	The fifth SystemRegister settings are used in the Prefetch Injection mode.
0xe	Prefetch injection 6	The sixth SystemRegister settings are used in the Prefetch Injection mode.
0xf	Prefetch injection 7	The seventh SystemRegister settings are used in the Prefetch Injection mode.

### 1.3.4. System Register Description

All registers of the hardware prefetch assistance are defined in IMPLEMENTATION DEFINE D region (S3\_<op1>\_<Cn>\_<Cm>\_<op2>).

Table 1-10 shows the list of all setting registers concerning the hardware prefetch assistance. All registers are defined by the 64 bits wide.

Table 1-10 Hardware prefetch assistance register list

op0	op1	CRn	CRm	op2	Register Name	Shared Domain
11	000	1011	0100	000	IMP_PF_CTRL_EL1	PE
11	011	1011	0100	000	IMP_PF_STREAM_DETECT_CTRL_EL0	PE
11	011	1011	0110	000	IMP_PF_INJECTION_CTRL0_EL0	PE
11	011	1011	0110	001	IMP_PF_INJECTION_CTRL1_EL0	PE
11	011	1011	0110	010	IMP_PF_INJECTION_CTRL2_EL0	PE
11	011	1011	0110	011	IMP_PF_INJECTION_CTRL3_EL0	PE
11	011	1011	0110	100	IMP_PF_INJECTION_CTRL4_EL0	PE
11	011	1011	0110	101	IMP_PF_INJECTION_CTRL5_EL0	PE
11	011	1011	0110	110	IMP_PF_INJECTION_CTRL6_EL0	PE
11	011	1011	0110	111	IMP_PF_INJECTION_CTRL7_EL0	PE
11	011	1011	0111	000	IMP_PF_INJECTION_DISTANCE0_EL0	PE
11	011	1011	0111	001	IMP_PF_INJECTION_DISTANCE1_EL0	PE
11	011	1011	0111	010	IMP_PF_INJECTION_DISTANCE2_EL0	PE
11	011	1011	0111	011	IMP_PF_INJECTION_DISTANCE3_EL0	PE
11	011	1011	0111	100	IMP_PF_INJECTION_DISTANCE4_EL0	PE
11	011	1011	0111	101	IMP_PF_INJECTION_DISTANCE5_EL0	PE
11	011	1011	0111	110	IMP_PF_INJECTION_DISTANCE6_EL0	PE
11	011	1011	0111	111	IMP_PF_INJECTION_DISTANCE7_EL0	PE

The access by Non-Secure EL1 and EL0 to a setting register of the hardware prefetch assistance is controlled from system register IMP\_PF\_CTRL\_EL1.

Table 1-11

Hardware prefetch assistance register access right

Register Name	el1ae=0			el1ae=1 and el0ae=0			el1ae=1 and el0ae=1		
	EL0	EL1 (NS)	EL1(S)	EL0	EL1 (NS)	EL1(S)	EL0	EL1 (NS)	EL1(S)
IMP_PF_CTRL_EL1		RO	RW		RW	RW		RW	RW
IMP_PF_STREAM_DETECT_CTRL_EL0			RW		RW	RW	RW	RW	RW
IMP_PF_INJECTION_CTRL[0-7]_EL0			RW		RW	RW	RW	RW	RW
IMP_PF_INJECTION_DISTANCE[0-7]_EL0			RW		RW	RW	RW	RW	RW

### 1.3.4.1. IMP\_PF\_CTRL\_EL1

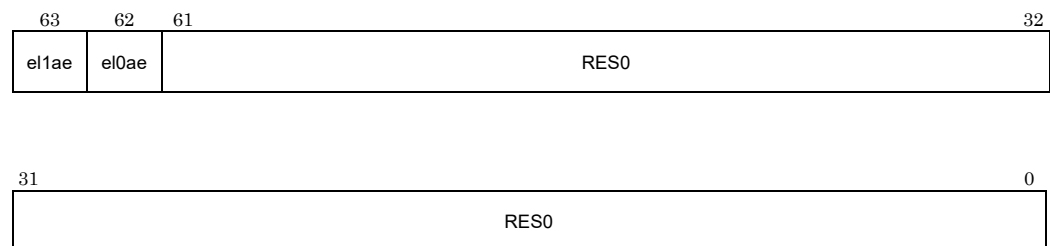
Purpose Prefetch assistance control register

Usage constraints IMP\_PF\_CTRL\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	RO/ Config- RW	RW	RW	RW	RW

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63]	el1ae	RW	<p>1: Read/write from Non-Secure EL1 is enabled to IMP_PF_STREAM_DETECT_CTRL_EL0, IMP_PF_INJECTION_CTRL[0-7]_EL0, IMP_PF_INJECTION_DISTANCE[0-7]_EL0</p> <p>0: Read/write from Non-Secure EL1 to IMP_PF_STREAM_DETECT_CTRL_EL0, IMP_PF_INJECTION_CTRL[0-7]_EL0, and IMP_PF_INJECTION_DISTANCE[0-7]_EL0 is trapped to EL2 with EC=0x18. Moreover, Write from Non-Secure EL1 to IMP_PF_CTRL_EL1 is trapped EL2 with EC=0x18. This bit is writable only from Secure EL1 and EL2/EL3. When the writing from Non-Secure EL1 at el1ae=1, the writing is ignored.</p>
[62]	el0ae	RW	<p>1: When el1ae=1, Read/Write from EL0 is enabled to IMP_PF_STREAM_DETECT_CTRL_EL0, IMP_PF_INJECTION_CTRL[0-7]_EL0 and IMP_PF_INJECTION_DISTANCE[0-7]_EL0. When el1ae=0, Access from EL0 to IMP_PF_STREAM_DETECT_CTRL_EL0, IMP_PF_INJECTION_CTRL[0-7]_EL0 and IMP_PF_INJECTION_DISTANCE[0-7]_EL0 is trapped to EL1 with EC=0x18.</p> <p>0: The access to IMP_PF_STREAM_DETECT_CTRL_EL0, IMP_PF_INJECTION_CTRL[0-7]_EL0 and IMP_PF_INJECTION_DISTANCE[0-7]_EL0 by EL0 is trapped to EL1 with EC=0x18.</p>
[61:0]	-	0x0	Reserved, RES0

Accessing MRS <Xt>, S3\_0\_C11\_C4\_0  
MSR S3\_0\_C11\_C4\_0, <Xt>

op0	op1	CRn	CRm	op2
11	000	1011	0100	000

#### 1.3.4.2. IMP\_PF\_STREAM\_DETECT\_CTRL\_EL0

Purpose Setting register for hardware prefetch control for PF\_STREAM\_DETECT

Usage constraints IMP\_PF\_STREAM\_DETECT\_CTRL\_EL0 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
Config-RW	Config-RW	RW	RW	RW	RW

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.



op0	op1	CRn	CRm	op2
11	011	1011	0100	000

### 1.3.4.3. IMP\_PF\_INJECTION\_CTRL[0-7]\_EL0

**Purpose** Setting register for hardware prefetch control for PF\_INJECTION

**Usage constraints** IMP\_PF\_INJECTION\_CTRL[0-7]\_EL0 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
Config-RW	Config-RW	RW	RW	RW	RW

**Configuration** This register is 64 bits wide.

**Attributes** The allocation of the register and implementation in the A64FX core are shown.

63	62	61	60	59	58	57	32
V	L1W	L2W	A	T	SWW	RES0	

31	25	24	2	1	0
RES0			PRQ_OFFSET[24:2]		RES0

Bits	Name	Value	Function
[63]	V	RW	1: PF injection is valid. 0: PF injection is disabled, the hardware prefetch to L1 and L2 is disabled, and the operation of the Prefetch instruction is Strong Prefetch.
[62]	L1W	RW	The hardware prefetch to L1 cache is Weak Prefetch.
[61]	L2W	RW	Weak Prefetch does the hardware prefetch to L2 cache.
[60]	A	RW	PFQ_ALLOCATION is valid.
[59]	T	RW	Type of Prefetch is specified. 0: PLD (prefetch for load) 1: PST (prefetch for store)
[58]	SWW	RW	The operation of the Prefetch instruction is set. 0: Strong Prefetch. 1: Weak Prefetch.
[57:25]	-	0x0	Reserved, RES0
[24:2]	PFQ_OFFSET	RW	PFQ registration address OFFSET is set within the range of + (16MB - 4B) ~ -16MB.
[1:0]	-	0x0	Reserved, RES0

Accessing MRS <Xt>, S3\_3\_C11\_C6\_[0-7]  
MSR S3\_3\_C11\_C6\_[0-7], <Xt>

op0	op1	CRn	CRm	op2
11	011	1011	0110	000 - 111

#### 1.3.4.4. IMP\_PF\_INJECTION\_DISTANCE[0-7]\_EL0

Purpose L1cache hardware prefetch Distance setting register for PF\_INJECTION

Usage constraints IMP\_PF\_INJECTION\_DISTANCE[0-7]\_EL0 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
Config- RW	Config- RW	RW	RW	RW	RW

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

63	57	56	34	33	32
RES0	L1PF_DISTANCE [24:2]				RES0

31	25	24	2	1	0
RES0	L2PF_DISTANCE[24:2]				RES0

Bits	Name	Value	Function
[63:57]	-	0x0	Reserved, RES0
[56:34]	L1PF_DISTANCE	RW	The Prefetch distance of L1cache is set within the range of + (16MB - 4B) ~ -16MB. When 0 is set, Hardware Prefetch to L1cache is disabled.
[33:25]	-	0x0	Reserved, RES0
[24:2]	L2PF_DISTANCE	RW	The Prefetch distance of L2cache is set within the range of + (16MB - 4B) ~ -16MB. When 0 is set, Hardware Prefetch to L2cache is disabled.
[1:0]	-	0x0	Reserved, RES0,

Accessing MRS <Xt>, S3\_3\_C11\_C7\_[0-7]  
MSR S3\_3\_C11\_C7\_[0-7], <Xt>

---

op0	op1	CRn	CRm	op2
11	011	1011	0111	000 - 111

---

## 1.4. Hardware barrier

Hardware barrier provides the synchronization function between different PEs by hardware. By using this function, it can accelerate synchronization performance to reduce the overhead of synchronization in a thread parallel program. The A64FX processor provides hardware barriers between PEs within the same CMG. Synchronization between CMGs can be realized by using software barrier synchronization.

### 1.4.1. Overview

On the A64FX processor, six barrier blades (BB) are mounted per CMG as resources for synchronization. Each BB indicates the physical position of the PE participating at initialization sequence, and each PE accesses the BB set through IMP\_BARRIER\_BS\_SYNC\_W[0-3]\_EL0/IMP\_BARRIER\_LBSY\_SYNC\_W[0-3]\_EL0 which are registers called window registers that are configured by IMP\_BARRIER\_ASSIGN\_SYNC\_W [0-3]\_EL1.

Since each PE has four sets of windows, four types of BB can be referred to simultaneously. When the window is accessed by MSR (Write), the BST bit is updated, and when the window is accessed by MRS (Read), the LBSY bit is read out.

The thread whose processing advances to the synchronization point inverts the LBSY bit read out from the window and writes it in the window. As a result, the BST bit of BB is updated to a value different from that of LBSY. When all PEs (Threads) participating in synchronization update the BST bit, the BB inverts the LBSY bit and notifies the PE of the Event. The program monitors the LBSY bit and determines that synchronization is complete when the written BST value and the LBSY value are identical. Refer to the usage example (p.37) for details.

### 1.4.2. Compatibility Note

The basic concept of the hardware barrier is like SPARC64 XIfx, but the WFE instruction is used instead of the sleep instruction to wait for barrier synchronization. When the barrier is established, an Event is notified to the PE participating in the barrier synchronization, and the PE returns from an Event waiting state.

### 1.4.3. System Register Description

All the hardware barriers are defined in IMPLEMENTATION DEFINE D region (S3\_<op1>\_<Cn>\_<Cm>\_<op2>). The explanation in this paragraph

Table 1-13 shows the list of all setting registers concerning the hardware barrier. All registers are 64 bits wide.

Table 713 Hardware barrier register list

op0	op1	CRn	CRm	op2	Register Name	Shared Domain
11	000	1011	1100	000	IMP_BARRIER_CTRL_EL1	PE
11	000	1011	1100	100	IMP_BARRIER_BST_BIT_EL1	PE
11	000	1111	1101	000	IMP_BARRIER_INIT_SYNC_BB0_EL1	CMG
11	000	1111	1101	001	IMP_BARRIER_INIT_SYNC_BB1_EL1	CMG
11	000	1111	1101	010	IMP_BARRIER_INIT_SYNC_BB2_EL1	CMG
11	000	1111	1101	011	IMP_BARRIER_INIT_SYNC_BB3_EL1	CMG
11	000	1111	1101	100	IMP_BARRIER_INIT_SYNC_BB4_EL1	CMG
11	000	1111	1101	101	IMP_BARRIER_INIT_SYNC_BB5_EL1	CMG
11	000	1111	1111	000	IMP_BARRIER_ASSIGN_SYNC_W0_EL1	PE
11	000	1111	1111	001	IMP_BARRIER_ASSIGN_SYNC_W1_EL1	PE
11	000	1111	1111	010	IMP_BARRIER_ASSIGN_SYNC_W2_EL1	PE
11	000	1111	1111	011	IMP_BARRIER_ASSIGN_SYNC_W3_EL1	PE
11	011	1111	1111	000	IMP_BARRIER_BST_SYNC_W0_EL0(W) IMP_BARRIER_LBSY_SYNC_W0_EL0(R)	PE(CMG) <sup>ii</sup>
11	011	1111	1111	001	IMP_BARRIER_BST_SYNC_W1_EL0(W) IMP_BARRIER_LBSY_SYNC_W1_EL0(R)	PE(CMG) <sup>ii</sup>
11	011	1111	1111	010	IMP_BARRIER_BST_SYNC_W2_EL0(W) IMP_BARRIER_LBSY_SYNC_W2_EL0(R)	PE(CMG) <sup>ii</sup>
11	011	1111	1111	011	IMP_BARRIER_BST_SYNC_W3_EL0(W) IMP_BARRIER_LBSY_SYNC_W3_EL0(R)	PE(CMG) <sup>ii</sup>

The access by Non-Secure EL1 and EL0 to the hardware barrier register is controlled from system register IMP\_BARRIER\_CTRL\_EL1. The access right and value of each register is as follows.

<sup>ii</sup> Those registers are the alias of IMP\_BARRIER\_INIT\_SYNC\_BB[0-5]\_EL1 that is specified by IMP\_BARRIER\_ASSIGN\_SYNC\_W[0-3]\_EL1.

Table 1-14 Hardware barrier register access right

Register Name	el1ae=0			el1ae=1 and el0ae=0			el1ae=1 and el0ae=1		
	EL0	EL1 (NS)	EL1(S)	EL0	EL1 (NS)	EL1(S)	EL0	EL1 (NS)	EL1(S)
IMP_BARRIER_CTRL_EL1		RO	RW		RW	RW		RW	RW
IMP_BARRIER_BST_BIT_EL1			RO		RO	RO		RO	RO
IMP_BARRIER_INIT_SYNC_BB*_EL1			RW		RW	RW		RW	RW
IMP_BARRIER_ASSIGN_SYNC_W*_EL1			RW		RW	RW		RW	RW
IMP_BARRIER_BST_SYNC_W*_EL0 / IMP_BARRIER_LBSY_SYNC_W*_EL0		RW	RW		RW	RW	RW	RW	RW

#### 1.4.3.1. IMP\_BARRIER\_CTRL\_EL1

Purpose Hardware barrier control register

Usage constraints IMP\_BARRIER\_CTRL\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	RO/ Config- RW	RW	RW	RW	RW

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

63	62	61	32
el1ae	el0ae	RES0	

31	0
RES0	

Bits	Name	Value	Function
[63]	el1ae	RW	<p>1: Read/write from Non-Secure EL1 is enabled to IMP_BARRIER_CTRL_EL1, IMP_BARRIER_INIT_SYNC_BB*_EL0, and IMP_BARRIER_ASSIGN_SYNC_W*_EL0. Moreover, Read from Non-Secure EL1 is enabled to IMP_BARRIER_BST_BIT_EL1.</p> <p>0: Write from Non-secure EL1 to IMP_BARRIER_CTRL_EL1 is trapped to EL2 with EC=0x18. Read/write from Non-Secure EL1 to IMP_BARRIER_INIT_SYNC_BB*_EL0, and IMP_BARRIER_ASSIGN_SYNC_W*_EL0 is trapped to EL2 with EC=0x18. Read from Non-Secure EL1 to IMP_BARRIER_BST_BIT_EL1 is trapped to EL2 with EC=0x18.</p> <p>This bit is writable only from Secure EL1 and EL2/EL3. When the writing from Non-Secure EL1 at el1ae=1, the writing is ignored.</p>
[62]	el0ae	RW	<p>1: When el1ae=1, Read/Write from EL0 is enabled to IMP_BARRIER_INIT_SYNC_BB*_EL0, and IMP_BARRIER_ASSIGN_SYNC_W*_EL0. When el1ae=0, access from EL0 to IMP_BARRIER_INIT_SYNC_BB*_EL0, and IMP_BARRIER_ASSIGN_SYNC_W*_EL0 is trapped to EL1 with EC=0x18.</p> <p>0: Read/write from EL0 to IMP_BARRIER_INIT_SYNC_BB*_EL0, and IMP_BARRIER_ASSIGN_SYNC_W*_EL0 is trapped to EL1 with EC=0x18.</p>
[61:0]	-	0x0	Reserved, RES0

Accessing MRS <Xt>, S3\_0\_C11\_C12\_0  
MSR S3\_0\_C11\_C12\_0, <Xt>

op0	op1	CRn	CRm	op2
11	000	1011	1100	000

#### 1.4.3.2. IMP\_BARRIER\_INIT\_SYNC\_BB[0-5]\_EL1

Purpose Hardware barrier initialization register

Usage constraints IMP\_BARRIER\_INIT\_SYNC\_BB 0-5 \_ EL1 is accessible in following Exception Level.

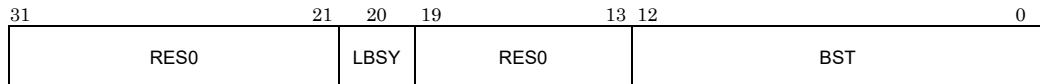
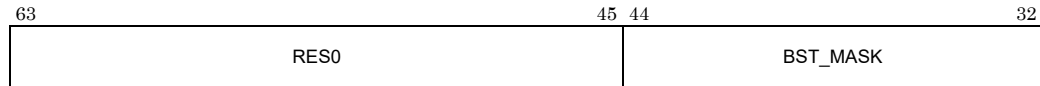
EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	Config- RW	RW	RW	RW	RW

Configuration This register is 64 bits wide. It is shared between two or more PE in CMG and it is to be noted that this register for influences other PE in

CMG by changing IMP\_BARRIER\_INIT\_SYNC\_BB 0-5 \_ EL1 in one PE.

Attributes

The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63:45]	-	0x0	Reserved, RES0
[44:32]	BST_MASK		The mask of BST is specified and read. The correspondence of each bit and PE is associated by the value acquired in IMP_BARRIER_BST_BIT_EL1. Details are Table 1-15 reference. When BST_MASK is all 0, it is not checked whether the barrier synchronization gets completed and the written value in LBSY is kept as it is.
[31:21]	-	0x0	Reserved, RES0
[20]	LBSY		LBSY. It is possible to read it with IMP_BARRIER_LBSY_SYNC_W*_EL0. When IMP_BARRIER_INIT_SYNC_BB 0-5 _ EL1 is written, it is checked whether the barrier synchronization gets completed with the value of BST and BST_MASK, and the LBSY value is updated.
[19:13]	-	0x0	Reserved, RES0
[12:0]	BST		The value of BST is specified and read. The correspondence of each bit and PE is associated by the value acquired in IMP_BARRIER_BST_BIT_EL1. Details are Table 1-15 reference.

The value check of BST and BST\_MASK and the update of LBSY are done as follows.

- If ((bst and bst\_mask) = 0), 0 is set in lbsy.
- If ((bst and bst\_mask) = bst\_mask), 1 is set in lbsy.

Table 1-15 Table for assistant core, computing core, BST, and BST\_MASK

<b>IMP_BARRIER_BST_BIT_</b> <b>EL1</b> <b>(BST_BIT)</b>	<b>BST</b>	<b>BST_MASK</b>
12 (Computing core 12 or Assistant core)	bit[12] (BST[12])	bit[44] (BST_MASK[12])
11 (computing core 11)	bit[11] (BST[11])	bit[43] (BST_MASK[11])
10 (computing core 10)	bit[10] (BST[10])	bit[42] (BST_MASK[10])
9 (computing core 9)	bit[9] (BST[9])	bit[41] (BST_MASK[9])
8 (computing core 8)	bit[8] (BST[8])	bit[40] (BST_MASK[8])
7 (computing core 7)	bit[7] (BST[7])	bit[39] (BST_MASK[7])
6 (computing core 6)	bit[6] (BST[6])	bit[38] (BST_MASK[6])
5 (computing core 5)	bit[5] (BST[5])	bit[37] (BST_MASK[5])
4 (computing core 4)	bit[4] (BST[4])	bit[36] (BST_MASK[4])
3 (computing core 3)	bit[3] (BST[3])	bit[35] (BST_MASK[3])
2 (computing core 2)	bit[2] (BST[2])	bit[34] (BST_MASK[2])
1 (computing core 1)	bit[1] (BST[1])	bit[33] (BST_MASK[1])
0 (computing core 0)	bit[0] (BST[0])	bit[32] (BST_MASK[0])

Accessing      MRS <Xt>, S3\_0\_C15\_C13\_0  
                   MSR S3\_0\_C15\_C13\_0, <Xt>

                  MRS <Xt>, S3\_0\_C15\_C13\_1  
                   MSR S3\_0\_C15\_C13\_1, <Xt>

                  MRS <Xt>, S3\_0\_C15\_C13\_2  
                   MSR S3\_0\_C15\_C13\_2, <Xt>

                  MRS <Xt>, S3\_0\_C15\_C13\_3  
                   MSR S3\_0\_C15\_C13\_3, <Xt>

                  MRS <Xt>, S3\_0\_C15\_C13\_4  
                   MSR S3\_0\_C15\_C13\_4, <Xt>

                  MRS <Xt>, S3\_0\_C15\_C13\_5  
                   MSR S3\_0\_C15\_C13\_5, <Xt>

<b>op0</b>	<b>op1</b>	<b>CRn</b>	<b>CRm</b>	<b>op2</b>
11	000	1111	1101	000-101

### 1.4.3.3. IMP\_BARRIER\_ASSIGN\_SYNC\_W[0-3]\_EL1

Purpose              Access control register for hardware barrier

Usage constraints   IMP\_BARRIER\_ASSIGN\_SYNC\_W[0-3]\_EL1 is accessible in  
                           following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	Config- RW	RW	RW	RW	RW

Configuration This register is 64 bits wide.

Attributes The allocation of the register and implementation in the A64FX core are shown.

63	62	32
VALID	RES0	

31	3	2	0
RES0			BB_NUM

Bits	Name	Value	Function
[63]	VALID		The allocation of the window is valid. When it accesses the window where VALID is 0 through IMP_BARRIER_BST_SYNC_W[0:3]_EL0(Write) and IMP_BARRIER_LBSY_SYNC_W[0:3]_EL0(Read), the result becomes UNPREDICTABLE.
[62:3]	-	0x0	Reserved, RES0
[2:0]	BB_NUM		BB that the window ALLOCATEs is selected (When 0x6 and 0x7 are selected, writing is ignored. However, it is not ignored and it is written for writing VALID=0).

Accessing

MRS <Xt>, S3\_0\_C15\_C15\_0  
MSR S3\_0\_C15\_C15\_0, <Xt>

MRS <Xt>, S3\_0\_C15\_C15\_1  
MSR S3\_0\_C15\_C15\_1, <Xt>

MRS <Xt>, S3\_0\_C15\_C15\_2  
MSR S3\_0\_C15\_C15\_2, <Xt>

MRS <Xt>, S3\_0\_C15\_C15\_3  
MSR S3\_0\_C15\_C15\_3, <Xt>

op0	op1	CRn	CRm	op2
11	000	1111	1111	000-011

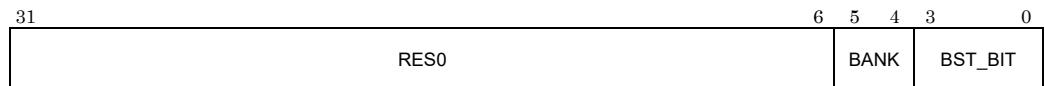
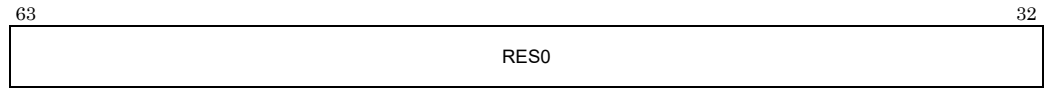
#### 1.4.3.4. IMP\_BARRIER\_BST\_BIT\_EL1

Purpose Physical, positional display register for hardware barrier

Usage constraints IMP\_BARRIER\_BST\_BIT\_EL1 is accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
	Config-RO	RO	RO	RO	RO

Configuration	This register is 64 bits wide.
Attributes	The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63:6]	-	0x0	Reserved, RES0
[5:4]	BANK		Physics CMG number is displayed.
[3:0]	BST_BIT		The physical core number in CMG is displayed. Refer to details to Table 1-15.

Accessing	MRS <Xt>, S3_0_C11_C12_4
-----------	--------------------------

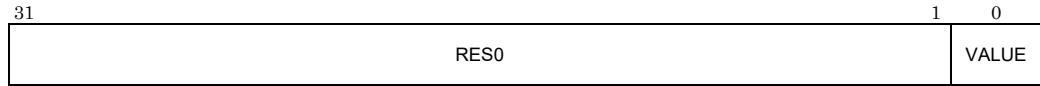
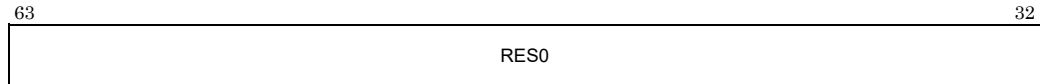
op0	op1	CRn	CRm	op2
11	000	1011	1100	100

#### 1.4.3.5. IMP\_BARRIER\_BST\_SYNC\_W[0-3]\_EL0(Write) IMP\_BARRIER\_LBSY\_SYNC\_W[0-3]\_EL0(Read)

Purpose	Window register for hardware
Usage constraints	IMP_BARRIER_BST_SYNC_W[0-3]_EL0 and IMP_BARRIER_LBSY_SYNC_W[0-3]_EL0 are accessible in following Exception Level.

EL0	EL1(NS)	EL1(S)	EL2	EL3(SCR.NS=1)	EL3(SCR.NS=0)
Config-RW	RW	RW	RW	RW	RW

Configuration	This register is 64 bits wide. This register is shared between two or more PEs in a CMG. Writing IMP_BARRIER_BST_SYNC_W[0-3]_EL0 affects corresponding BB(IMP_BARRIER_INIT_SYNC_BB[0-5]_EL1) assigned by IMP_BARRIER_ASSIGN_SYNC_W[0-3]_EL1 As a result, it is to be noted that it influences other PE.
Attributes	The allocation of the register and implementation in the A64FX core are shown.



Bits	Name	Value	Function
[63:1]	-	0x0	Reserved, RES0
[0]	VALUE		Write: The bit for the physical core number of the BST field corresponding to BB set by IMP_BARRIER_ASSIGN_SYNC_W[0-3]_EL1 is updated. Read: The value of LBSY corresponding to BB set by IMP_BARRIER_ASSIGN_SYNC_W[0-3]_EL1 is read.  Attention: When IMP_BARRIER_ASSIGN_SYNC_W[0-3]_EL1.VALID is 0, the result of Write/Read is UNPREDICTABLE.

Accessing

MRS <Xt>, S3\_3\_C15\_C15\_0 /\* IMP\_BARRIER\_LBSY\_SYNC\_W0\_EL0 \*/  
 MSR S3\_3\_C15\_C15\_0, <Xt> /\* IMP\_BARRIER\_BST\_SYNC\_W0\_EL0 \*/

MRS <Xt>, S3\_3\_C15\_C15\_1 /\* IMP\_BARRIER\_LBSY\_SYNC\_W1\_EL0 \*/  
 MSR S3\_3\_C15\_C15\_1, <Xt> /\* IMP\_BARRIER\_BST\_SYNC\_W1\_EL0 \*/

MRS <Xt>, S3\_3\_C15\_C15\_2 /\* IMP\_BARRIER\_LBSY\_SYNC\_W2\_EL0 \*/  
 MSR S3\_3\_C15\_C15\_2, <Xt> /\* IMP\_BARRIER\_BST\_SYNC\_W2\_EL0 \*/

MRS <Xt>, S3\_3\_C15\_C15\_3 /\* IMP\_BARRIER\_LBSY\_SYNC\_W3\_EL0 \*/  
 MSR S3\_3\_C15\_C15\_3, <Xt> /\* IMP\_BARRIER\_BST\_SYNC\_W3\_EL0 \*/

op0	op1	CRn	CRm	op2
11	011	1111	1111	000-011

### Usage example

```

mrs X1, S3_3_C15_C15_0    // Read LBSY
mvn X1, X1                // Invert read LBSY
and X1, X1, #1            // Mask out RES0 bits
msr S3_3_C15_C15_0, X1    // Write BST
                           // When update BST bit, Don't require
                           // ISB instruction
sevl                      // Local Event register set

loop:

wfe                        // Wait
mrs X2, S3_3_C15_C15_0    // Read LBSY
and X2, X2, #1            // Mask out RES0 bits
cmp X1, X2                // Compare the readout and written value
b.eq post-sync            // When LBSY==BST, Synchronization is
                           // completed
b    loop                 // Check LBSY again

```

---

```
post-sync:
:
```