

JEDEC STANDARD

SPI Safety Extensions (CRC) for Non Volatile SPI Flash Memories (QPI and xSPI)

JESD255

February 2024

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION



NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or refer to www.jedec.org under Standards and Documents for alternative contact information.

Published by

©JEDEC Solid State Technology Association 2024
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

JEDEC retains the copyright on this material. By downloading this file the individual agrees not to charge for or resell the resulting material.

PRICE: Contact JEDEC

Printed in the U.S.A.
All rights reserved

DO NOT VIOLATE
THE
LAW!

This document is copyrighted by JEDEC and may not be
reproduced without permission.

Organizations may obtain permission to reproduce a limited number of copies
through entering into a license agreement. For information, contact:

JEDEC Solid State Technology Association
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107
<https://www.jedec.org/contact>

This page intentionally left blank

SPI Safety Extensions (CRC) for Non Volatile SPI Flash Memories (QPI and xSPI)

Contents

	Pages
Foreword.....	iii
Introduction.....	iii
1 Scope.....	1
2 Normative Reference.....	1
3 Terms and Definitions.....	2
4 CRC Key Features	3
4.1 General Features.....	3
4.2 CRC Modes.....	3
4.2.1 CRC-8	4
4.2.2 CRC-16	4
4.3 Checksum Calculations	4
4.4 Checksum Fields	5
4.5 Transaction Formats by Protocol Mode	5
4.5.1 Profile 1.0 DOPI (8D-8D-8D) 16-Bit Aligned Bus Transaction Format Prototypes with CRC-16....	6
4.5.2 Profile 1.0 SOPI (8S-8S-8S) 8-Bit Aligned Bus Transaction Format Prototypes with CRC-8	7
4.5.3 Profile 1.0 DTR QPI (4S-4D-4D & 4D-4D-4D) and STR QPI (4S-4S-4S) 8-Bit Aligned Bus Transaction Format Prototypes with CRC-8.....	8
4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats	9
4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats	14
5 CRC Status and Configuration Registers	20
6 Annex A — (Informative) Differences between Document Revisions	20
7 Annex B — (Informative) AUTOSAR CRC Routines.....	21
7.1 For CRC-8, with 8 Bits of Data:	21
7.2 CRC-16, with 16-Bit of Data:	21
7.3 Numerical Examples for CRC-8 and CRC-16 Calculation:.....	21

Contents (cont'd)

List of Figures

Figure 1 — Profile 1.0 8D-0D-0D Format 1.A: (e.g., Write Enable, Chip Erase)	9
Figure 2 — Profile 1.0 8D-8D-8D Format 1.B: (e.g., Fast Read)	9
Figure 3 — Profile 1.0 8D-8D-0D Format 1.C: (e.g., Sector Erase, Block Erase).....	10
Figure 4 — Profile 1.0 8D-8D-8D Format 1.D: (e.g., Page Program).....	10
Figure 5 — Profile 1.0 8D-0D-8D Format 1.E: (e.g., Write Extended Address Reg)	10
Figure 6 — Profile 1.0 8D-8D-8D Format 1.F: (e.g., Write 1-Byte Register).....	11
Figure 7 — Profile 1.0 8D-0D-8D Format 1.G: (e.g., Write >1-Byte Register).....	11
Figure 8 — Profile 1.0 8D-8D-8D Format 1.H: (e.g., Write >1-Byte Register).....	11
Figure 9 — Profile 1.0 8D-0/8D-8D Format 1.I: (e.g., Read 1-Byte Register)	12
Figure 10 — Profile 1.0 8D-8D-8S Format 1.J: (e.g., Read 1-Byte Register).....	12
Figure 11 — Profile 1.0 8D-0/8D-8D Format 1.K: (e.g., Read >1-Byte Register)	13
Figure 12 — Profile 1.0 8D-8D-8S Format 1.L: (e.g., Read >1-Byte Register).....	13
Figure 13 — Profile 1.0 8D-0D-8D Format 1.M: (e.g., Read Extended Status Reg w/o Address)	14
Figure 14 — Profile 1.0 4S-4D-0D Format 3.A: (e.g., Write Enable).....	14
Figure 15 — Profile 1.0 4S-0D-4D Format 3.B: (e.g., Read Status Register).....	15
Figure 16 — Profile 1.0 4S-4D-4D Format 3.C: Read SFDP.....	15
Figure 17 — Profile 1.0 4S-4D-4D Format 3.D: Read n*(CRC line length) Bytes	15
Figure 18 — Profile 1.0 4S-4D-4D Format 3.E: (e.g., Fast Read Quad I/O)	16
Figure 19 — Profile 1.0 4S-4D-0D Format 3.F: Write 1 Byte Data.....	16
Figure 20 — Profile 1.0 4S-4D-0D Format 3.G: (e.g., Sector Erase).....	16
Figure 21 — Profile 1.0 4S-4D-4D Format 3.H: 1-Byte Address, CRC, Write 1 Byte Data	17
Figure 22 — Profile 1.0 4S-4D-4D Format 3.I: (e.g., Page Program).....	17
Figure 23 — Profile 1.0 4S-4D-4D Format 3.J: Write 1 Byte Data	17
Figure 24 — Profile 1.0 4S-4D-4D Format 3.K: Write >1 Byte Data.....	18
Figure 25 — Profile 1.0 4S-4D-4D Format 3.L: Read 1 Byte Data	18
Figure 26 — Profile 1.0 4S-4D-4D Format 3.M: (e.g., Read ID (9F)).....	19
Figure 27 — Profile 1.0 4S-4D-4D Format 3.N: (e.g., Read Extended Status Reg)	19

List of Tables

Table 1 — Profile 1.0 DOPI (8D-8D-8D) Format Prototypes.....	6
Table 2 — Profile 1.0 SOPI (8S-8S-8S) Format Prototypes	7
Table 3 — Profile 1.0 DTR QPI (4S-4D-4D, 4D-4D-4D) and STR QPI (4S-4S-4S) Format Prototypes....	8

Foreword

This document was prepared by the JC42.4_3 Serial Flash task group authorized by the JC-42.4 NonVolatile Memory subcommittee.

This standard is intended for use by SoC, ASIC, ASSP, and FPGA developers or vendors interested in incorporating an initiator (sometimes called ‘controller’) interface having a low signal count and high data transfer bandwidth with access to multiple sources of target devices compliant with the interface. It is also, intended for use by peripheral developers or vendors interested in providing target devices compliant with the standard, including non-volatile memories, volatile memories, graphics peripherals, networking peripherals, FPGAs, sensors, etc.

Introduction

This standard defines optional CRC functions for Quad and Expanded Serial Peripheral Interface (QPI and xSPI) nonvolatile memory devices.

The standard provides descriptions of CRC modes, configuring and enabling them. The CRC modes covered include CRC-8 and CRC-16 and their polynomials.

This page intentionally left blank

SPI Safety Extensions (CRC) for Non Volatile SPI Flash Memories (QPI and xSPI)

From JEDEC Board Ballot JCB-24-01, formulated under the cognizance of the JC-42.4 subcommittee on Non Volatile Memory, item number 1775.76.

1 Scope

The JESD255 document defines CRC modes supported with 8-bit aligned and 16-bit aligned data transactions. It is limited to logical bus transactions and does not cover the electrical properties of the IO bus.

2 Normative Reference

The Normative References listed in JEDEC Standard JESD251 and the following additional normative documents contain provisions that through reference in this text, constitute provisions of this standard.

- ISO 26262-1:2018 Standard, road vehicles functional safety.
- JEDEC Standard, JESD251, Expanded Serial Peripheral Interface (xSPI) for NonVolatile Memory Devices
- JEDEC Standard, JESD251-1, Addendum No. 1 to JESD251 - OPTIONAL x4 QUAD I/O with Data Strobe
- Koopman: <https://users.ece.cmu.edu/~koopman/crc/>
- AUTOSAR: Specification of CRC Polynomials https://www.autosar.org/fileadmin/standards/R21-11/CP/AUTOSAR_SWS_CRCLibrary.pdf

3 Terms and Definitions

For purposes of this publication, the following abbreviations for common terms apply:

- **ASIL-D:** Automotive Safety Integrity Level. The standard defines functional safety as “the absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical or electronic systems.” ASILs establish safety requirements — based on the probability and acceptability of harm — for automotive components to be compliant with ISO 26262-1:2018.
- **AUTOSAR:** (**A**UTomotive **O**pen **S**ystem **A**Rchitecture) is a worldwide development partnership of vehicle manufacturers, suppliers, service providers, and companies from the automotive electronics, semiconductor, and software industry.
- **Checksum Field:** 1 or 2 CRC bytes
- **CRC:** Cyclic Redundancy Check
- **CRC#:** CRC complement
- **CRC-8:** 8-bit cyclic redundancy check
- **CRC-16:** 16-bit cyclic redundancy check
- **DDR:** Double Data Rate where data is received or transmitted on both clock edges (rising and falling).
- **DTR:** Double Transfer Rate is synonymous with DDR and can be used interchangeably.
- **DTR QPI (4D-4D-4D, 4S-4D-4D):** Instruction format for Double Transfer Rate QPI
- **Data chunk size:** number of bytes in a data payload
- **DOPI (8D-8D-8D):** Double Transfer Rate Octal Peripheral Interface
- **Dummy Cycles:** Clock cycles after target address is captured and before data is output. CRC clocks or mode bits are included in the count.
- **Mode Bits:** Used to enable continuous read (command bypass mode). Not supported when CRC mode is enabled.
- **OPI:** Octal Peripheral Interface
- **POR:** Power-on Reset
- **QPI:** Quad Peripheral Interface
- **SOPI (8S-8S-8S):** Single Transfer Rate Octal Peripheral Interface
- **SDR:** Single Data Rate where data is received or transmitted on only one clock edge (typically rising edge)
- **STR:** Single Transfer Rate is synonymous with SDR and can be used interchangeably.
- **STR OPI (8S-8S-8S):** Instruction format for Single Transfer Rate OPI
- **STR QPI (4S-4S-4S):** Instruction format for Single Transfer Rate QPI

4 CRC Key Features

The following is a summary of the CRC features:

4.1 General Features

- Provides CRC protection to all IO bus transactions, including commands, command modifiers, addresses, and data.
- A CRC is generated for command/command modifiers/address and a separate CRC for each data chunk.
- CRC enable/disable under user control
- Extended data transfers are to be partitioned into segments with each segment having a CRC value
 - Data chunk sizes: 16B, 32B, 64B, and 128B; is user selectable
 - Data transfers need to be aligned to the configured segment length
 - A transaction can only end after a full segment length for CRC protection
 - Other transactions that are less than the segment length (e.g., register reads/writes), and protected only by Data/Data#, can terminate anytime.
- CRC mode (CRC-8 or CRC-16) depends on the bus mode (SOPI or QPI = CRC-8 and DOPI=CRC-16).
- CRC can be enabled or disabled by default after POR
- When a read request fails the CRC check (on the memory):
 - The memory will intentionally “corrupt” the CRC (or Data/Data# if a return CRC is not used) on the data returned to the host
 - A CRC failure will be noted in the Status Register
 - Interrupt output indication will be generated

4.2 CRC Modes

Two CRC modes are defined: CRC-8 (for 8-bit aligned bus modes) and CRC-16 (for 16-bit aligned bus modes). The CRC mode is derived from the SPI bus mode.

4.2.1 CRC-8

- Provides an 8-bit CRC checksum on the data stream
- 2-bit error detection
- Data chunk size up to 16 Bytes
- Suitable for ASIL-D applications that use the following instruction formats:
 - STR OPI (8S-8S-8S)
 - DTR QPI (4D-4D-4D)
 - DTR QPI (4S-4D-4D)
 - STR QPI (4S-4S-4S)
 - Any other applicable 8-bit aligned transaction format

4.2.2 CRC-16

- Provides 16-bit CRC checksums for the major elements of the data stream
- 3-bit error detection
- Data chunk size is typically set to 16 Bytes but may be set as high as 128 Bytes
 - Suitable for ASIL-D applications that use the DTR OPI (8D-8D-8D) format
 - Any other applicable 16-bit aligned transaction format

4.3 Checksum Calculations

- AUTOSAR polynomials are used for CRC-8 and CRC-16 Checksum calculations (see Annex B for details)
- DTR OPI (8D-8D-8D) is a 16-bit aligned interface therefore a 16-bit Checksum field (CRC-16) is used
- Other bus modes use CRC-8
- The checksum is reset to FFh/FFFFh on CS# assertion (on new instruction) and after each checksum field is transmitted over the SPI bus
- Checksum fields themselves are excluded from the calculation
- Command Extensions (to provide 16-bit bus alignment) and Data extensions (i.e., a register byte that is repeated twice or more) are included in the checksum calculation
- Dummy cycles are excluded from the calculation.

4.4 Checksum Fields

- DTR OPI Formats (8D-8D-8D)
 - Use CRC-16.
 - Two checksum bytes are used to transmit the 16-bit CRC (MSB first).
 - Instruction OpCode is extended to 16-bit.
 - OpCode Extension byte (2nd byte) may be identical to the OpCode (“Command Repeat Mode”), its inverse (“Command Inverse Mode”), or “Don’t Care” and is always included in the CRC-16 calculation.
 - The OpCode Extension (EXT) is included in the checksum calculation.
 - Data Extension: For 1B register read/write, where the 1B data payload/response, the data is extended to 16-bit to maintain bus alignment.
 - The Data Extension byte (2nd byte) is identical to the Data byte (“Data Repeat Mode”) or its inverse (“Data Inverse Mode”).

4.5 Transaction Formats by Protocol Mode

- The following protocol mode formats will be described in Tables 1, 2, and 3:
 - Profile 1.0 DOPI (8D-8D-8D, CRC-16)
 - Example timing diagrams for xSPI DOPI 8D-8D-8D mode are shown in Section 4.5.4 and were based on the rules shown in Table 1 — Profile 1.0 DOPI (8D-8D-8D) Format Prototypes.
 - Profile 1.0 SOPI (8S-8S-8S, CRC-8)
 - Timing diagrams for xSPI SOPI 8S-8S-8S are not shown but they can be derived based on the rules shown in Table 2 — Profile 1.0 SOPI (8S-8S-8S) Format Prototypes
 - Profile 1.0 STR QPI (4S-4S-4S, CRC-8)
 - Timing diagrams for QPI (4S-4S-4S) are not shown but they can be derived based on the rules shown in Table 3 — Profile 1.0 DTR QPI (4S-4D-4D, 4D-4D-4D) and STR QPI (4S-4S-4S) Format Prototypes.
 - Profile 1.0 DTR QPI (4S-4D-4D, CRC-8)
 - Example timing diagrams for QPI (4S-4D-4D) mode are shown in Section 4.5.5 and were based on the rules shown in Table 3 — Profile 1.0 DTR QPI (4S-4D-4D, 4D-4D-4D) and STR QPI (4S-4S-4S) Format Prototypes.
 - Profile 1.0 DTR QPI (4D-4D-4D, CRC-8)
 - Timing diagrams for QPI (4D-4D-4D) are not shown but they can be derived based on the rules shown in Table 3 — Profile 1.0 DTR QPI (4S-4D-4D, 4D-4D-4D) and STR QPI (4S-4S-4S) Format Prototypes.

4.5.1 Profile 1.0 DOPI (8D-8D-8D) 16-Bit Aligned Bus Transaction Format Prototypes with CRC-16

Table 1 — Profile 1.0 DOPI (8D-8D-8D) Format Prototypes

8D-8D-8D Write Clock Cycle		1	2	3	4	5	6	7	8	9	10	11	12	These Clock Cycles/Edges apply to the write phase only. 1st Data Out may occur after a rising or falling edge of the clock													
8D-8D-8D Write Clock Edge		↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓		↑	↓											
8D-8D-8D Read Data Strobe (DS)		X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	H	L	H	L	H	L	H	L	H	L	H	L	1st DS output H may occur after a rising or falling edge of clock. There is no guaranteed phase relationship		
Format ^{1,2}	Phases	Example Instruction																					Comments				
1.A	C	Chip Erase	CMD	EXT	CRC1	CRC2																Command (CMD/EXT) is protected by CRC-16 CRC1=MSB checksum; CRC2=LSB checksum					
1.B	C+A+d+RDnB	Fast Read Read SFDP	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2	← dummy ³ →				D0	D1	Dn-2	Dn-1	CRC1	CRC2	Dn	Dn+1	D2n-2	D2n-1	CRC1	CRC2	Each data segment (nB) protected by CRC-16 Output 'n' = CRC line length
1.C	C+A	Sector Erase	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2													CRC-16 to protect CMD+Modifier				
1.D	C+A+WRnB	Page Program	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2	D0	D1	--	--	Dn-2	Dn-1	CRC1	CRC2	Dn	Dn+1	Dn+2	---	D2n-2	D2n-1	CRC1	CRC2	Each data segment (nB) is protected by CRC-16 'n' = CRC line length. Data segments < 'nB' must be padded with FFh (or 00h) by the host to fill the CRC line
1.E	C+WR1B	Write 1B Register	CMD	EXT	CRC1	CRC2	D	D#															D/D# as data				
1.F	C+A+WR1B	Register	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2	D	D#															
1.G	C+WR>1B	Write >1B Register	CMD	EXT	CRC1	CRC2	D0	D1	---	Dm	FFh	FFh	FFh	FFh	---	FFh	CRC1	CRC2						Pad data with FFh (or 00h) to fill CRC-line length			
1.H	C+A+WR>1B	Register	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2	D0	D1	---	Dm	FFh	FFh	FFh	FFh	---	FFh	CRC1	CRC2					
1.I	C+A+d+RD1B	Read 1B Register	CMD	EXT	[A3] ⁴	[A2] ⁴	[A1] ⁴	[A0] ⁴	CRC1	CRC2	← dummy ³ →				D	D#	D	D#	D	D#	D	D#	---	---	---	---	D/D# as data
1.J		Register	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2	← dummy ³ →				D	D	D#	D#	D	D	D#	D#	---	---	---	---	DD/D#D# as data. This format is vendor specific.
1.K	C+[A] ⁴ +d+RD>1B	Read >1B Register	CMD	EXT	[A3] ⁴	[A2] ⁴	[A1] ⁴	[A0] ⁴	CRC1	CRC2	← dummy ³ →				D0	D1	--	Dm	FFh	FFh	FFh	FFh	---	FFh	CRC1	CRC2	Device pads response with FFh (or 00h) to fill CRC-line length
1.L	C+A+d+RD>1B	Read >1B Register	CMD	EXT	A3	A2	A1	A0	CRC1	CRC2	← dummy ³ →				D0	D0	--	--	Dm	Dm	FFh	FFh	---	FFh	CRC1	CRC2	Device pads response with FFh (or 00h) to fill CRC-line length. This format is vendor specific.
1.M	C+d+RD4B	Read 4B Register	CMD	EXT	CRC1	CRC2	← dummy ³ →				D0	D1	D2	D3	CRC1	CRC2	D4	D5	D6	D7	CRC1	CRC2			Each 4B of response is protected by CRC-16		

Phases:

C = Command

A = Address

d = dummy cycles

RD1B = Read 1 Byte

RD3B = Read 3 Bytes

RD4B = Read 4 Bytes

RDnB = Read 'n' Bytes

RD>1B = Read >1 Bytes

WR1B = Write 1 Bytes

WR>1B = Write >1 Bytes

WRnB = Write 'n' Bytes

Notes:

1. Formats 1.A to 1.D derived from JESD251 (xSPI).

2. Formats 1.E to 1.M are not defined in JESD251, but are additions to JESD251 made by this addendum.

3. dummy cycles are not shown to scale and their number varies depending on the instruction.

4. Address may be 0 or 4 Bytes

5. DS will be driven to 'L' before the first 'H' but the timing cannot be determined here. Refer to JESD251 xSPI spec for timing details.

Color Key:

Command+Modifier+Address Field
CRC-16 Checksum Input
CRC-16 Checksum Output
dummy cycles
Data Field
FFh (or 00h) fill data is included in CRC

NOTE: This standard requires only a subset of the above-mentioned formats. The optional formats are described to ensure that if other commands are implemented in the 8D-8D-8D mode, they will follow a pre-defined format.

4.5.2 Profile 1.0 SOPI (8S-8S-8S) 8-Bit Aligned Bus Transaction Format Prototypes with CRC-8

Table 2 — Profile 1.0 SOPI (8S-8S-8S) Format Prototypes

8S-8S-8S Write Clock Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	These Clock Cycles/Edges apply to the write phase only. During the read phase, Data Out always follows the falling edge of the clock
8S-8S-8S Write Clock Edge	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	
8S-8S-8S Read Data Strobe	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	X ⁵	H	L	H	L	H	L	H	L	H	L	H	DS is edge aligned with Data Out and always follows the falling edge of the clock

Format ^{1,2}	Phases	Example Instruction																					Comments		
2.A	C	Chip Erase	CMD	EXT ⁴	CRC																Command (CMD/EXT) is protected by CRC-8				
2.B	C+A+d+RDnB	Fast Read	CMD	EXT ⁴	[A3] ³	A2	A1	A0	CRC	dummy ²					D0	D1	Dn-2	Dn-1	CRC	Dn	Dn+1	D2n-2	D2n-1	CRC	Each data segment (nB) protected by CRC-8 Output 'n' = CRC line length
2.C	C+A	Sector Erase	CMD	EXT ⁴	A3	A2	A1	A0	CRC																CRC-8 to protect CMD+Modifier
2.D	C+A+WRnB	Page Program	CMD	EXT ⁴	A3	A2	A1	A0	CRC	D0	D1	--	--	Dn-2	Dn-1	CRC	Dn	Dn+1	Dn+2	---	D2n-2	D2n-1	CRC		Each data segment (nB) is protected by CRC-16 'n' = CRC line length. Data segments < 'nB' must be padded with FFh (or 00h) by the host to fill the CRC line

2.E	C+WR1B	Write 1B Register	CMD	EXT ⁴	CRC	D	D#																D/D# as data	
2.F	C+A+WR1B		CMD	EXT ⁴	A3	A2	A1	A0	CRC	D	D#													

2.G	C+WR>1B	Write >1B Register	CMD	EXT ⁴	CRC	D0	D1	---	Dm	FFh	FFh	FFh	FFh	---	FFh	CRC										Device pads data with FFh (or 00h) to fill CRC-line length
2.H	C+A+WR>1B		CMD	EXT ⁴	A3	A2	A1	A0	CRC	D0	D1	---	Dm	FFh	FFh	FFh	FFh	---	FFh	CRC						

2.I	C+d+RD1B	Read 1B Register	CMD	EXT ⁴	CRC	dummy ²										D	D#	D	D#	D	D#	---	---	D/D# as repeating response data
2.J	C+A+d+RD1B		CMD	EXT ⁴	A3	A2	A1	A0	CRC	dummy ²					D	D#	D	D#	D	D#	---	---		

2.K	C+d+RD>1B	Read >1B (Device ID)	CMD	EXT ⁴	CRC	dummy ²										D0	D1	D2	---	FFh	FFh	FFh	FFh	---	FFh	CRC	Device pads response with FFh (or 00h) to fill CRC-line
2.L	C+A+d+RD>1B		CMD	EXT ⁴	A3	A2	A1	A0	CRC	dummy ²					D0	D1	D2	---	FFh	FFh	FFh	FFh	---	FFh	CRC		

2.M	C+d+RD4B	Read 4B Register	CMD	EXT ⁴	CRC	dummy ²										D0	D1	D2	D3	CRC	D4	D5	D6	D7	CRC	Each 4B of response is protected by CRC-8 Output
-----	----------	------------------	-----	------------------	-----	--------------------	--	--	--	--	--	--	--	--	--	----	----	----	----	-----	----	----	----	----	-----	--

Phases:

C = Command

A = Address

d = dummy cycles

RD1B = Read 1 Byte

RD4B = Read 4 Bytes

RDnB = Read 'n' Bytes

RD>1B = Read >1 Byte

WR1B = Write 1 Bytes

WR>1B = Write >1 Byte

WRnB = Write 'n' Bytes

Notes:

1. Formats 2.A to 2.M are not defined in JESD251, but are additions to JESD251 made by this addendum.

2. Dummy cycles are not shown to scale and their number varies depending on the instruction.

3. Only used for 4B address mode. Not sent by host if 3B mode is active

4. Command Extension (EXT) is optional. Without EXT, the format is CMD/CRC or CMD/ADDR/CRC

5. DS will be driven to 'L' before the first 'H' but the timing cannot be determined here. Refer to JESD251 xSPI spec for timing details.

Color Key:

Command+Modifier+Address Field

CRC-8 Checksum Input

CRC-8 Checksum Output

dummy cycles

Data Field

FFh (or 00h) fill data

Optional EXT Field

4.5.3 Profile 1.0 DTR QPI (4S-4D-4D & 4D-4D-4D) and STR QPI (4S-4S-4S) 8-Bit Aligned Bus Transaction Format Prototypes with CRC-8

Table 3 — Profile 1.0 DTR QPI (4S-4D-4D, 4D-4D-4D) and STR QPI (4S-4S-4S) Format Prototypes

STR QPI (4S-4S-4S) Clock Cycle

STR QPI (4S-4S-4S) D_{in} Clock Edge

DTR QPI (4S-4D-4D) Clock Cycle

DTR QPI (4S-4D-4D) Write Clock Edge

DTR QPI (4D-4D-4D) Clock Cycle

DTR QPI (4D-4D-4D) Write Clock Edge

1234567891011121314151617181920212223242526272829303132333435363738³

↑↑

4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats

- In the following timing diagrams, “CRC1” denotes the MSB Checksum-byte and “CRC2” denotes the LSB Checksum-byte.
- For DOPI, CRC-16 is used so “CRC1” is the 8 MS bits and “CRC2” is the 8 LS bits.
- In the following diagrams, “D” denotes the data byte and “D#” denotes the extension byte, which can be identical or inverse value of the preceding data byte.
- Note that DS (Data Strobe) is mandatory per the xSPI spec, but it is not shown in the following timing diagrams for simplification and to focus on CRC placement and usage.
- Profile 1.0 8D-0D-0D Format 1.A: CMD, EXT, CRC1, CRC2:

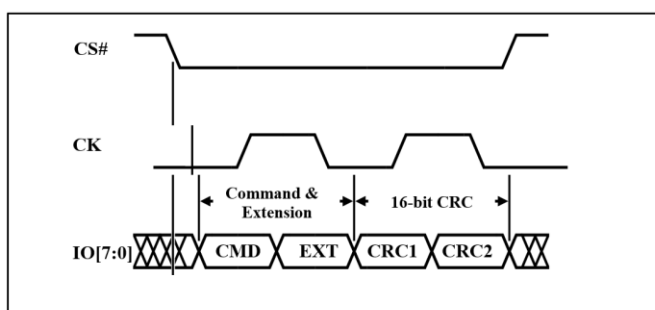


Figure 1 — Profile 1.0 8D-0D-0D Format 1.A: (e.g., Write Enable, Chip Erase)

- Profile 1.0 8D-8D-8D Format 1.B: CMD, EXT, 4-byte Address, CRC1, CRC2, ‘x’ Dummy Cycles, Read Data Chunk, CRC1, CRC2,Read Data Chunk, CRC1, CRC2:

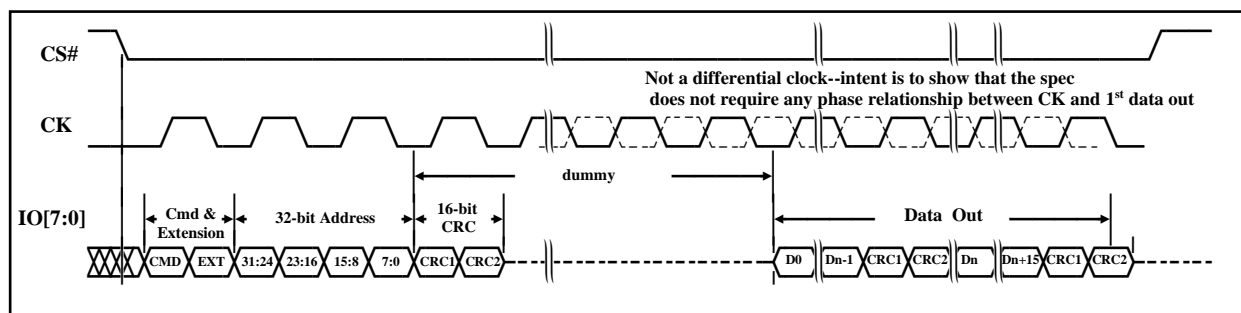


Figure 2 — Profile 1.0 8D-8D-8D Format 1.B: (e.g., Fast Read)

NOTE: Timing shown for data chunk size = 16 Bytes, but data chunk sizes of 32B, 64B, and 128B are also supported.

4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats (cont'd)

- Profile 1.0 8D-8D-0D Format 1.C: CMD, EXT, 4-Byte Address, CRC1, CRC2:

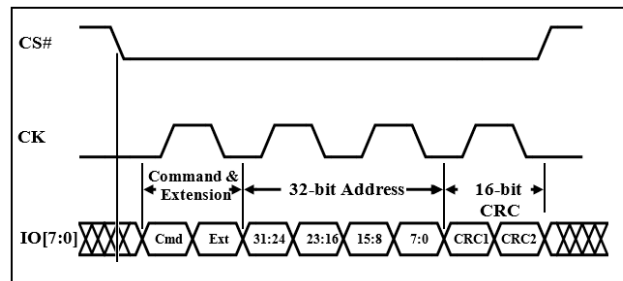


Figure 3 — Profile 1.0 8D-8D-0D Format 1.C: (e.g., Sector Erase, Block Erase)

- Profile 1.0 8D-8D-8D Format 1.D: CMD, EXT, 4-byte Address, Write Data Chunk, CRC1, CRC2, Write Data Chunk, CRC1, CRC2....Write Data Chunk, CRC1, CRC2:

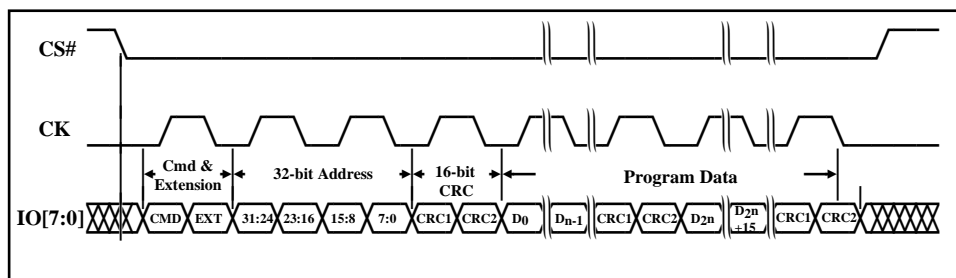


Figure 4 — Profile 1.0 8D-8D-8D Format 1.D: (e.g., Page Program)

NOTE: Timing shown for data chunk size = 16 Bytes, but data chunk sizes of 32B, 64B, and 128B are also supported.

- Profile 1.0 8D-0D-8D Format 1.E: CMD, EXT, Write Data Byte, Write Data# Byte, CRC1, CRC2:

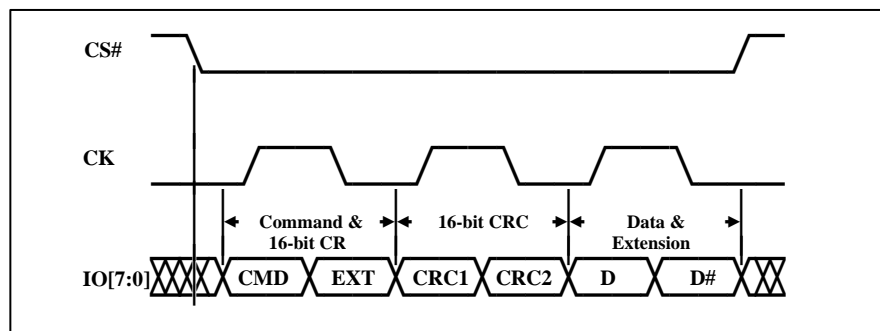


Figure 5 — Profile 1.0 8D-0D-8D Format 1.E: (e.g., Write Extended Address Reg)

4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats (cont'd)

- Profile 1.0 8D-8D-8D Format 1.F: CMD, EXT, 32-bit Address, CRC1, CRC2, Write Data/Data# Bytes:

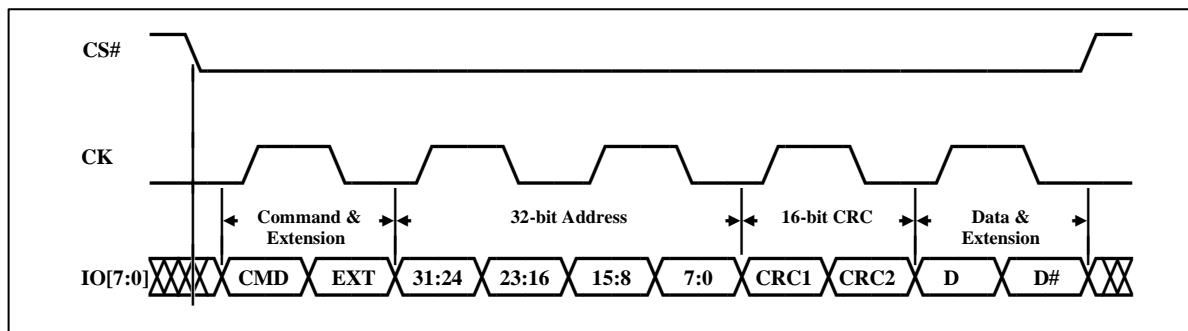


Figure 6 — Profile 1.0 8D-8D-8D Format 1.F: (e.g., Write 1-Byte Register)

- Profile 1.0 8D-0D-8D Format 1.G: CMD, EXT, Write >1 Data Bytes, Write FFh (or 00h) Pad Bytes, CRC1, CRC2:

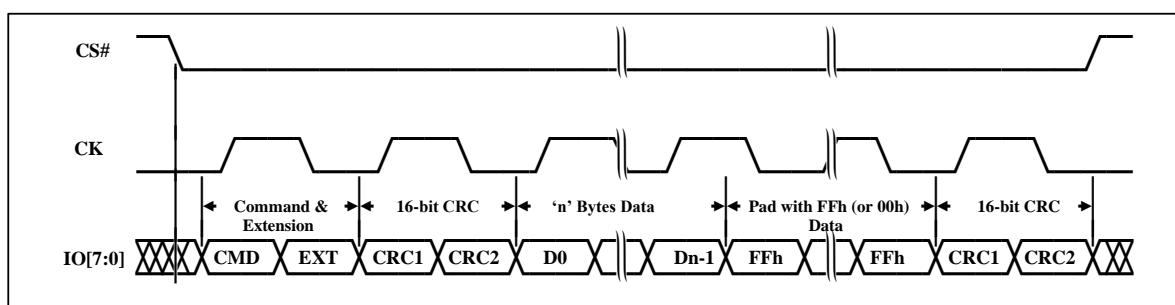


Figure 7 — Profile 1.0 8D-0D-8D Format 1.G: (e.g., Write >1-Byte Register)

- Profile 1.0 8D-8D-8D Format 1.H: CMD, EXT, 32-bit Address, Write >1 Data Bytes, Write FFh (or 00h) Pad Bytes, CRC1, CRC2:

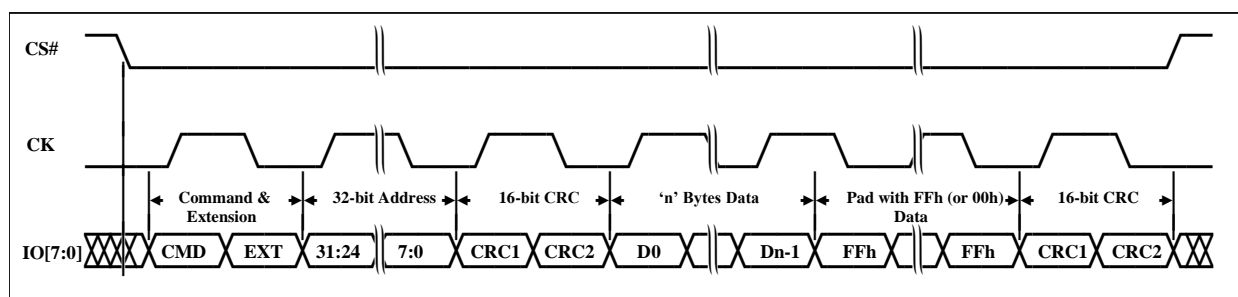


Figure 8 — Profile 1.0 8D-8D-8D Format 1.H: (e.g., Write >1-Byte Register)

4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats (cont'd)

- Profile 1.0 8D-0/8D-8D Format 1.I: CMD, EXT, 0- or 32-bit Address, CRC1, CRC2, 'x' Dummy Cycles, Read Data/Data# Bytes:

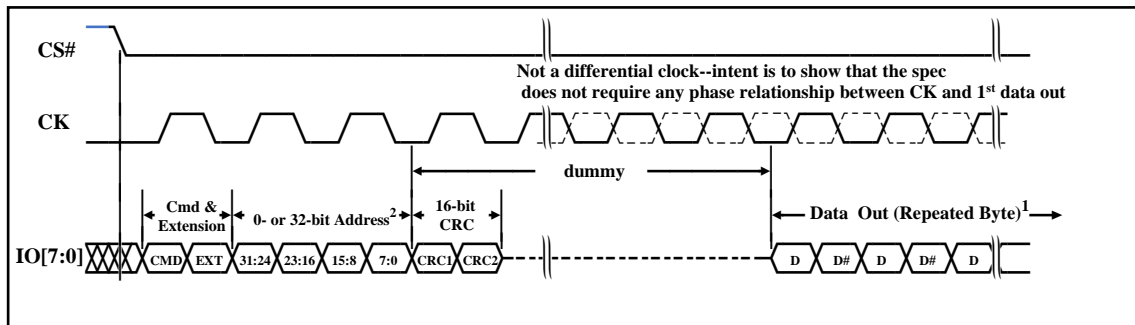


Figure 9 — Profile 1.0 8D-0/8D-8D Format 1.I: (e.g., Read 1-Byte Register)

- NOTES:
- Response Data ("D") is not covered by a CRC Checksum. Data Extension "Inverse" mode should be used to protect data
 - Address may be 0- or 32-bits

- Profile 1.0 8D-8D-8D Format 1.J: CMD, EXT, 32-bit Address, CRC1, CRC2, 'x' Dummy Cycles, Read Data/Data# Bytes:

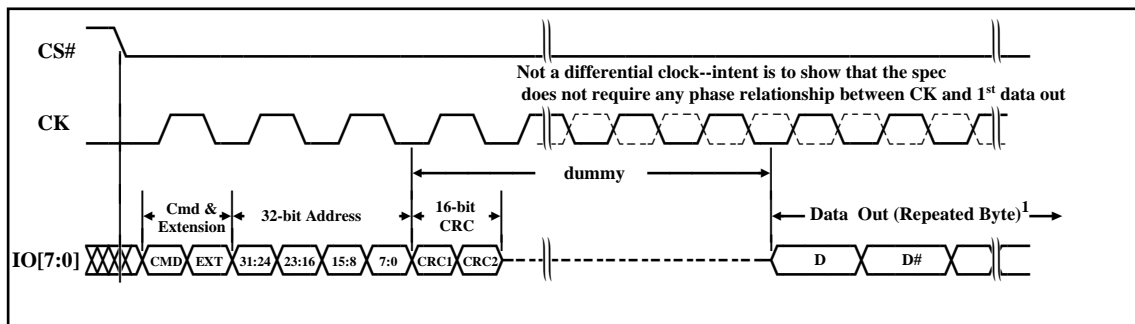


Figure 10 — Profile 1.0 8D-8D-8S Format 1.J: (e.g., Read 1-Byte Register)

- NOTES:
- Response Data ("D") is not covered by a CRC Checksum. Data Extension "Inverse" mode ("D#") should be used to protect data.
 - This format is vendor specific.

4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats (cont'd)

- Profile 1.0 8D-0/8D-8D Format 1.K: CMD, EXT, 0- or 32-bit Address, CRC1, CRC2, 'x' Dummy Cycles, Read >1 Bytes, FFh pad to fill the CRC line, CRC1, CRC2:

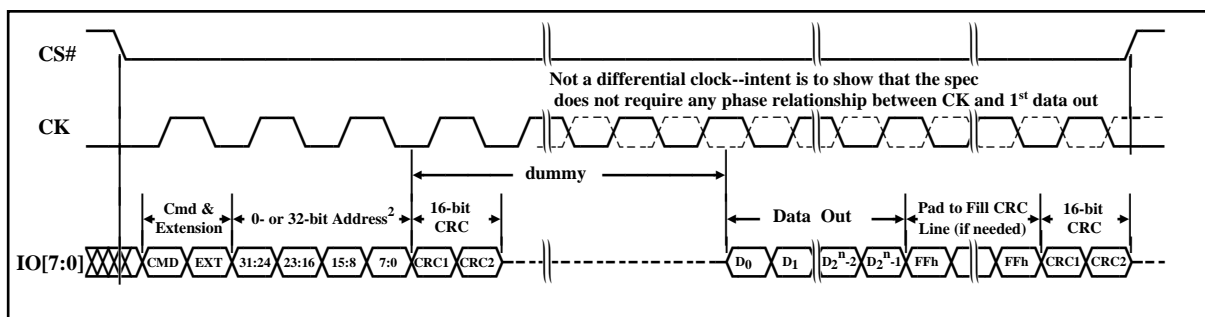


Figure 11 — Profile 1.0 8D-0/8D-8D Format 1.K: (e.g., Read >1-Byte Register)

- Notes:
1. The fixed-size response is padded if needed with FFh (or 00h) pad to fill the CRC line.
 2. If 0-bit Address is used, dummy cycles begin immediately after CMD/EXT phase

- Profile 1.0 8D-8D-8D Format 1.L: CMD, EXT, 32-bit Address, CRC1, CRC2, 'x' Dummy Cycles, Read >1 Bytes, Pad FFh, CRC1, CRC2:

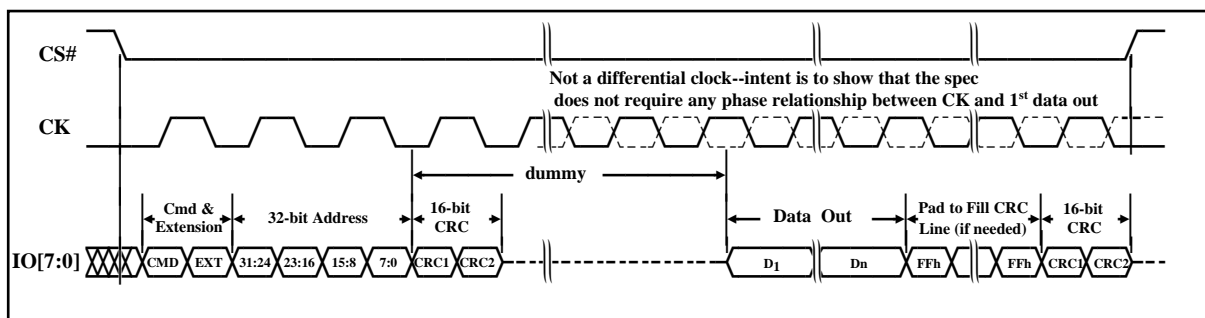


Figure 12 — Profile 1.0 8D-8D-8S Format 1.L: (e.g., Read >1-Byte Register)

Notes:

1. The fixed-size response is padded if needed with FF (or 00h) pad to fill the CRC line.
2. Data is output in 8S mode, but FFh pad data and CRC are output in 8D mode.
3. This format is vendor specific.

4.5.4 Profile 1.0 8D-8D-8D CRC Transaction Formats (cont'd)

- Profile 1.0 8D-0D-8D Format 1.M: CMD, EXT, CRC1, CRC2, 'x' Dummy Cycles, Read DWORD (4 Bytes), CRC1, CRC2, Read DWORD (4 Bytes), CRC1, CRC2,

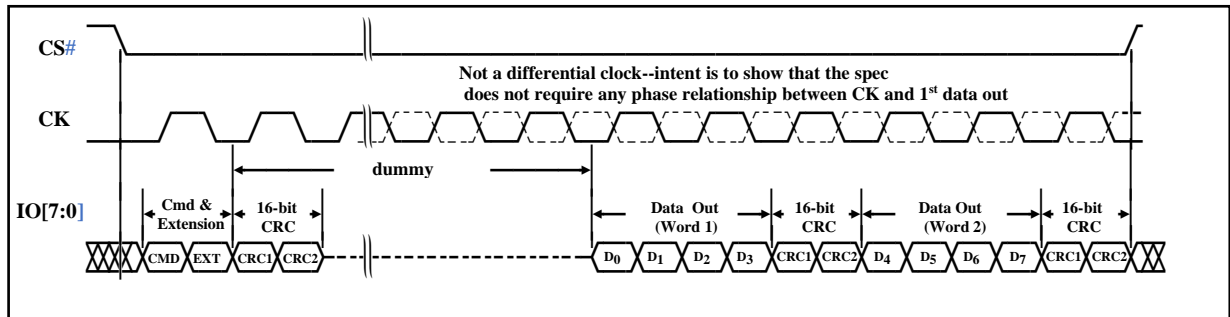


Figure 13 — Profile 1.0 8D-0D-8D Format 1.M: (e.g., Read Extended Status Reg without Address)

NOTE: This format contains response words of 4 bytes each, where each word is followed by a checksum field. Any number of words can be read and the instruction can be terminated after any checksum field. This format is equivalent to format 1.F with a data size of 4 bytes and CRC line length also set to 4 bytes.

4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats

- For all QPI modes, CRC-8 is used.
- In the following diagrams, “D” denotes the data byte and “D#” denotes the optional data extension byte, which can be inverse value of the preceding data byte or omitted altogether.
- Profile 1.0 4S-4D-0D Format 3.A: STR CMD, DTR CRC

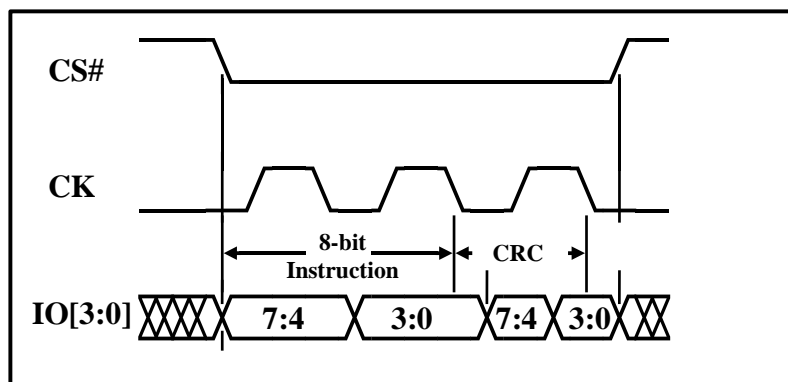


Figure 14 — Profile 1.0 4S-4D-0D Format 3.A: (e.g., Write Enable)

4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats (cont'd)

- Profile 1.0 4S-0D-4D Format 3.B: STR CMD, CRC, 'x' Dummy, DTR Read Data/Data#

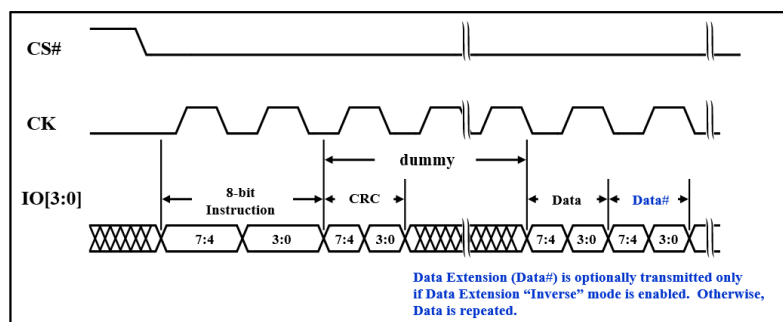


Figure 15 — Profile 1.0 4S-0D-4D Format 3.B: (e.g., Read Status Register)

* = For some devices, 'x' is zero Dummy (the minimum dummy cycle count must allow for the CRC)

- Profile 1.0 4S-4D-4D Format 3.C: STR: CMD, DTR: 3-Byte Address, CRC, 'x' Dummy Cycles, Read n*(CRC line length) Data, CRC

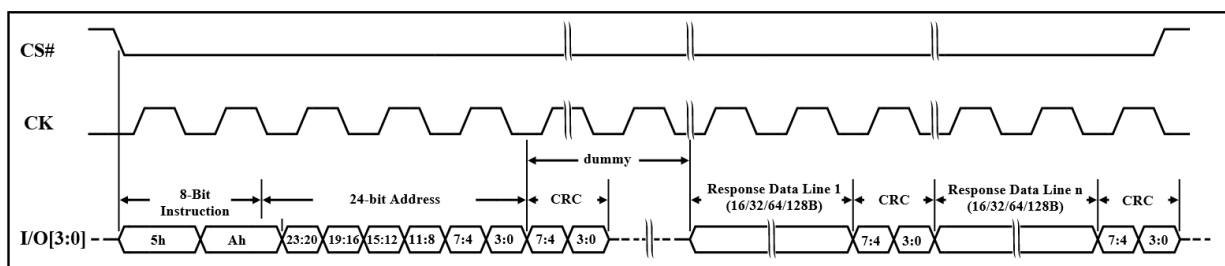


Figure 16 — Profile 1.0 4S-4D-4D Format 3.C: Read SFDP

- Profile 1.0 4S-4D-4D Format 3.D: STR: CMD, DTR: 1-Byte Address, CRC, 'x' Dummy Cycles, Read n*(CRC line length) Bytes

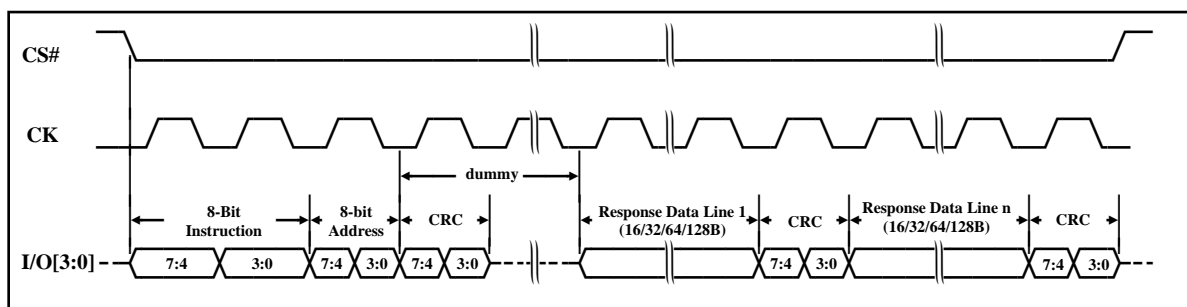


Figure 17 — Profile 1.0 4S-4D-4D Format 3.D: Read n*(CRC line length) Bytes

4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats (cont'd)

- Profile 1.0 4S-4D-4D Format 3.E: STR: CMD, DTR: 4-Byte Address, CRC, 'x' Dummy Cycles, Fast Read $n \times (\text{CRC line length})$ Bytes

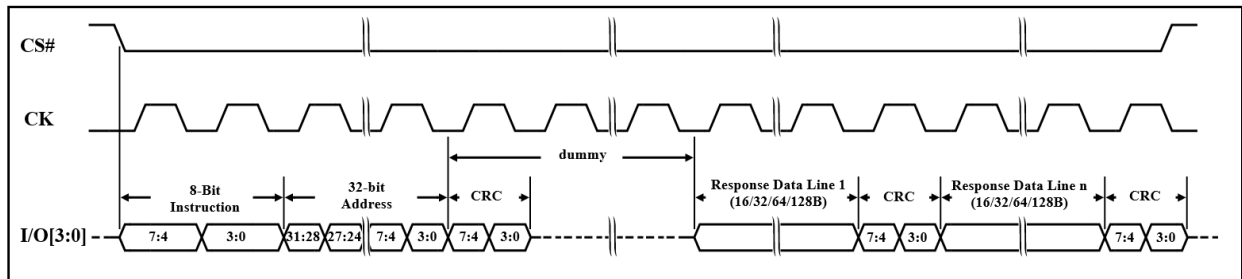


Figure 18 — Profile 1.0 4S-4D-4D Format 3.E: (e.g., Fast Read Quad I/O)

NOTE: No restrictions on starting address.

- Profile 1.0 4S-4D-4D Format 3.F: STR: CMD, CRC, DTR: Write Data Byte, Write Data# Byte

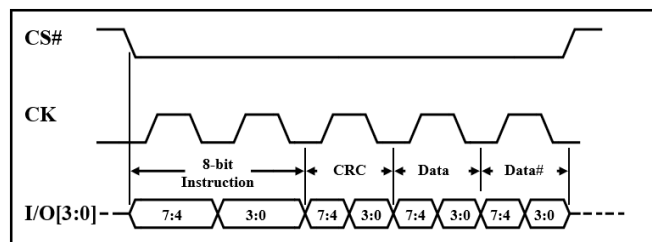


Figure 19 — Profile 1.0 4S-4D-0D Format 3.F: Write 1 Byte Data

- Profile 1.0 4S-4D-0D Format 3.G: STR: CMD, DTR: 4-Byte Address, CRC

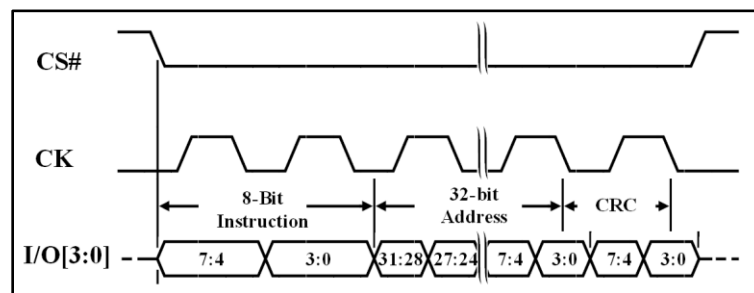


Figure 20 — Profile 1.0 4S-4D-0D Format 3.G: (e.g., Sector Erase)

4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats (cont'd)

- Profile 1.0 4S-4D-4D Format 3.H: STR: CMD, DTR: 1-Byte Address, CRC, Write 1 Byte Data

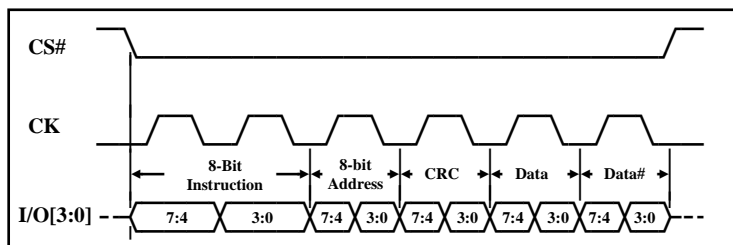


Figure 21 — Profile 1.0 4S-4D-4D Format 3.H: 1-Byte Address, CRC, Write 1 Byte Data

- Profile 1.0 4S-4D-4D Format 3.I: STR: CMD, DTR: 4-Byte Address, CRC, Write n^* (CRC line length) Bytes

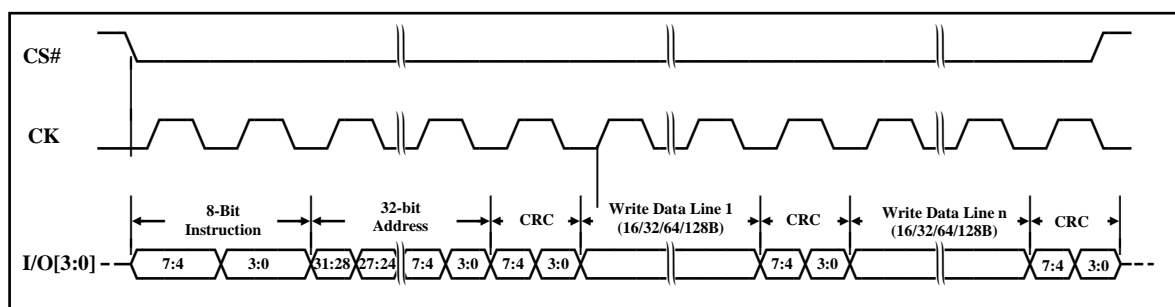


Figure 22 — Profile 1.0 4S-4D-4D Format 3.I: (e.g., Page Program)

- Profile 1.0 4S-4D-4D Format 3.J: STR: CMD, DTR: 24- or 32-bit Address, CRC, Write 1 Byte Data

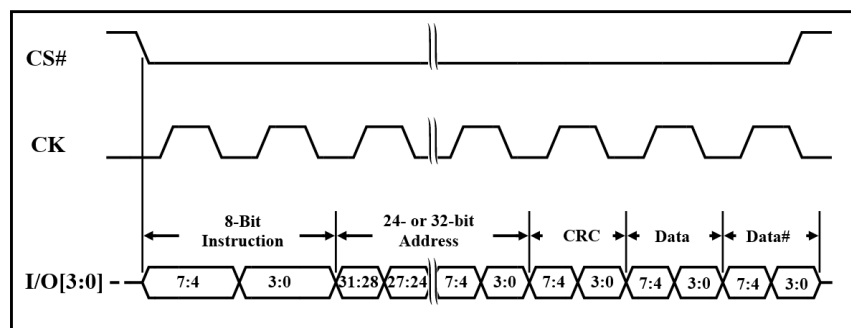


Figure 23 — Profile 1.0 4S-4D-4D Format 3.J: Write 1 Byte Data

4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats (cont'd)

- Profile 1.0 4S-4D-4D Format 3.K: STR: CMD, DTR: 0-, 24- or 32-bit Address, CRC, Write >1 Byte Data, Pad FFh (or 00h) to fill CRC line, CRC

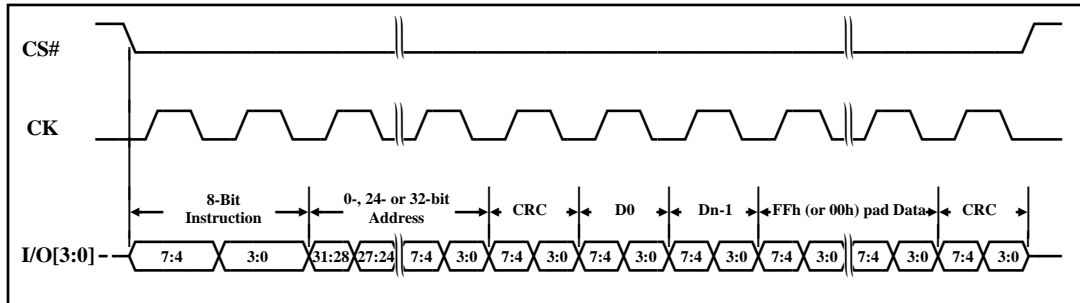


Figure 24 — Profile 1.0 4S-4D-4D Format 3.K: Write >1 Byte Data

- Profile 1.0 4S-4D-4D Format 3.L: STR: CMD, DTR: 24- or 32-bit Address, CRC, Read 1 Byte Data

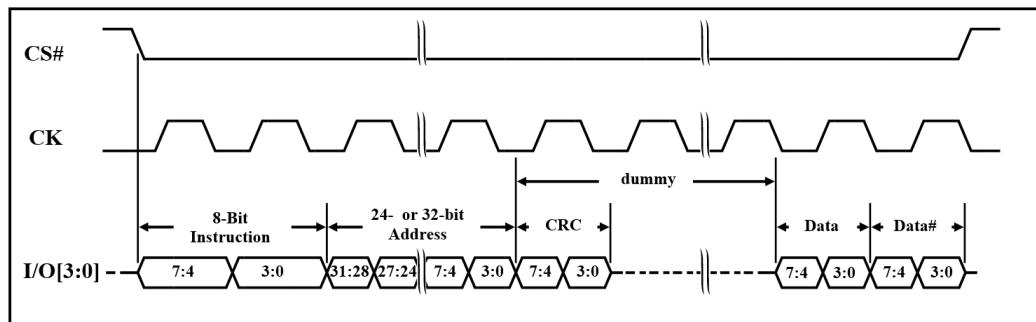


Figure 25 — Profile 1.0 4S-4D-4D Format 3.L: Read 1 Byte Data

4.5.5 Profile 1.0 4S-4D-4D CRC Transaction Formats (cont'd)

- Profile 1.0 4S-4D-4D Format 3.M: STR: CMD, DTR: 0-, 24-, or 32-bit Address, Read 'n' Bytes Data, pad FFh (or 00h) to fill CRC line, CRC

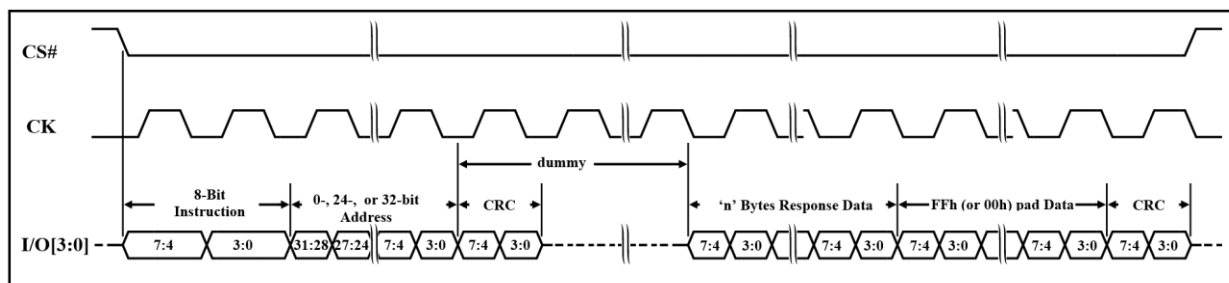


Figure 26 — Profile 1.0 4S-4D-4D Format 3.M: (e.g., Read ID (9F))

- Profile 1.0 4S-4D-4D Format 3.N: STR: CMD, DTR: Read 4 Bytes Data, CRC

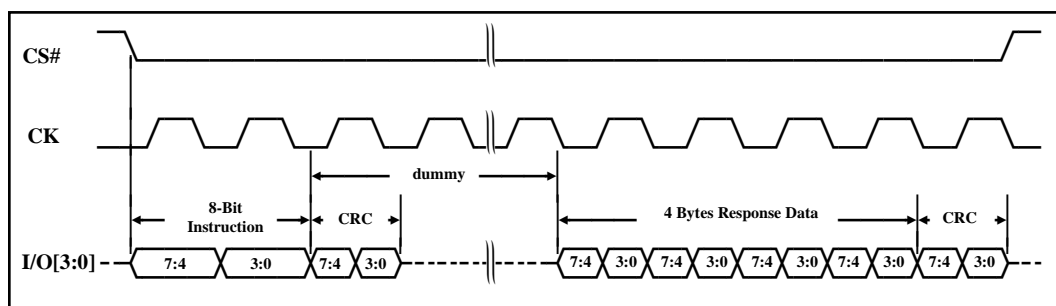


Figure 27 — Profile 1.0 4S-4D-4D Format 3.N: (e.g., Read Extended Status Reg)

5 CRC Status and Configuration Registers

5.1 Status and Configuration Registers

- The following features must be supported:
 - CRC Enable/Disable
 - CRC Line Length selection (16, 32, 64, or 128 bytes)
 - Command/-Command Extension (In 8D-8D-8D mode, the two first rising/falling clock edges capture the command op-code and its extension. The command extension may be CMD (repeated), CMD# (inverted), or Don't Care). Whichever extension is used, it is included in the CRC.
 - Data/Data Extension (some registers only output a single byte, but in 8D-8D-8D mode, a fill byte is required for the clock falling edge. The data extension fill byte must be the inverted data (e.g., DATA/DATA#)).
 - CRC Integrity Error Flag (indicates a CRC error occurred)
 - CRC_INT (set when CRC Integrity Error Flag is set)
 - CRC_IMASK (disables external INT# output if a CRC error occurred)
- The Status and Configuration Registers are defined by the JEDEC CRC Parameter Table in JESD216G (or later revisions).

6 Annex A — (Informative) Differences between Document Revisions

A.0 Initial release

7 Annex B — (Informative) AUTOSAR CRC Routines

7.1 For CRC-8, with 8 Bits of Data:

Hamming Distance (HD) of 3 for 16B data, thus all 2-bit errors are detected ([Koopman])
 // CRC polynomial coefficients: $x^8 + x^4 + x^3 + x^2 + 1$
 // 0x1D (hex)
 // CRC width: 8 bits
 // CRC shift direction: left (big-endian)
 // Input word width: 8 bits
 // Initial Value: FFh
 // Result Value: inverted

7.2 CRC-16, with 16-Bit of Data:

Hamming Distance (HD) of 4 for 16B-128B data (and up to 31kb), thus all 3b errors are detected ([Koopman]).
 // CRC polynomial coefficients: $x^{16} + x^{12} + x^5 + 1$
 // 0x1021 (hex)
 // CRC width: 16 bits
 // CRC shift direction: left (big-endian)
 // Input word width: 16 bits
 // Initial Value: FFFFh

7.3 Numerical Examples for CRC-8 and CRC-16 Calculation:

SOPI/QPI FAST-READ (EBh) with CRC-8 and Dummy Cycles example for 0x1D polynomial:

0xEB 0x00 0x12 0x34 0x56 0xC3 [Dummy Cycles] 0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87 0x88
 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F 0xA2

Where:

- 0xEB is the instruction OpCode
- 0x00123456 is the instruction address (big Endian)
- 0xC3 is the command CRC-8
- 0x80 0x81 ... 0x8F is the 16B response data (little Endian)
- 0xA2 is the response CRC-8

DOPI FAST-READ with CRC-16 and Dummy Cycles example for 0x1021 polynomial:

0xEB 0x15 0x00 0x12 0x34 0x56 0xBF 0xE6 [Dummy Cycles] 0x80 0x81 0x82 0x83 0x84 0x85 0x86
 0x87 0x88 0x89 0x8A 0x8B 0x8C 0x8D 0x8E 0x8F 0x31 0xFD

Where:

- 0xEB 0x15 is the instruction OpCode and extension (inverse mode)
- 0x00123456 is the instruction address (big Endian)
- 0xBFE6 is the command CRC-16 (MS Byte first)
- 0x80 0x81 ... 0x8F is the 16B response data (little Endian)
- 0x31FD is the response CRC-16 (MS Byte first)

This page intentionally left blank



Standard Improvement Form**JEDEC Standard No. JESD255**

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form and return to:

JEDEC
Attn: Publications Department
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

3. Other suggestions for document improvement:

Submitted by

Name: _____

Phone: _____

Company: _____

E-mail: _____

Address: _____

City/State/Zip: _____

Date: _____

