JEDEC STANDARD

Graphics Double Data Rate 7 SGRAM Standard (GDDR7)

JESD239.01

(Editorial Revision of JESD239, February 2024)

ATTENTION USERS: Be advised that the formulating subcommittee is working on an update that may require host design changes. Members of the subcommittee may contact the TG for more information..

April 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

3103 10th Street North
Suite 240S
Arlington, VA 22201

PRICE: Contact JEDEC

Printed in the U.S.A.

DO NOT VIOLATE

THE

LAW!

For information, contact:

JEDEC Solid State Technology Association

3103 10th Street North

Suite 240S

Arlington, VA 22201

https://www.jedec.org/contact

ATTENTION USERS

Be advised that the formulating subcommittee is working on an update that may require host design changes. Members of the subcommittee may contact the TG for more information.

Graphics Double Data Rate 7 SGRAM Standard (GDDR7)

Contents

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

List of Figures

Pages

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

List of Tables

Pages

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

Contents (cont'd)

Graphics Double Data Rate 7 SGRAM Standard (GDDR7)

From JEDEC Board Ballot JCB-24-02, formulated under the cognizance of the JC-42.1 sub-committee on Graphics RAMs (GDDRx), item 1860.99.

1 Scope

This standard defines the Graphics Double Data Rate 7 (GDDR7) Synchronous Graphics Random Access Memory (SGRAM) specification, including features, functionality, package, and pin assignments.

The purpose of this standard is to define the minimum set of requirements for 16 Gb through 64 Gb x8 quad channel GDDR7 SGRAM devices. System designs based on the required aspects of this standard will be supported by all GDDR7 SGRAM vendors providing compatible devices. Some aspects of the GDDR7 standard such as AC timings and capacitance values were not standardized. Some features are optional and therefore may vary among vendors. In all cases, vendor data sheets should be consulted for specifics. This document was created based on some aspects of the GDDR6 Standard (JESD250) and other JEDEC device standards.

2 Overview, Definitions, and Organization

2.1 Features

- 4 separate independent channels with point-to-point interface for data, Command Address (CA), and clocks
- Common differential clock inputs WCK_t/WCK_c per channel for the data bus (DQ, DQE) and the CA bus
- Double Data Rate (DDR) Data bus (with regards to the WCK)
- High-speed data signaling using PAM3; low speed data signaling option for NRZ
- Single ended or differential Read clock (RCK) per channel
- Single Data Rate (SDR) CA bus (with regards to the WCK); NRZ input with single and multicycle commands
- ERR signal (PAM3 coded) for communicating WRCRC and CAPAR errors
- CA Parity (CAPAR) and CA Parity with command blocking (CAPARBLK)
- 16 internal banks per channel
- 32n prefetch architecture: 256 bit per array read or write access per channel
- Burst length: 16 Symbols (PAM3) or 32 (NRZ)
- Programmable READ and WRITE latency
- Command Address bus inversion (CABI)
- Command Address bus training: command address input monitoring by DQ signals
- Data read and write training via READ FIFO or LFSR
- READ FIFO pattern preload by LDFF command
- PRBS 11 or 15 with programmable seed for LFSR Training
- LFSR burst error counter for Write training
- Lane masking, inversion, and pattern shift for LFSR training
- Eye masking for LSFR or FIFO mode training
- ERR signal training
- CA Oscillator (CAOSC)
- Read/Write data transmission integrity secured by cyclic redundancy check
- READ/WRITE CRC on/off mode
- Programmable DQE latency and WRCRC2ERR latency
- On-chip temperature sensor with read-out
- Auto precharge option for each burst access
- Auto refresh (16k cycles) with all-bank and per-bank options
- Self Refresh, Sleep, Self Refresh Sleep and Hibernate Self Refresh Sleep modes
- Temperature Controlled Self Refresh rate
- Frequency change sequence and Dynamic Voltage Sequence (DVS) support including FDMR
- CSP command for Sleep to synchronous operation
- CSP feedback for confirmation that device received CSP command
- Info Read for device identification and status information
- On-die ECC with error severity reporting during Reads
- Refresh Management (RFM), Adaptive RFM and Directed RFM
- Post package repair support (hPPR)
- On-die termination (ODT)
- Impedance calibration with external reference resistor (120 Ohm)
- Programmable termination and driver strength offsets for DQ/DQE/RCK
- Internal VREF for data inputs and CA inputs with programmable levels
- 4-channel/2-channel mode configuration set at power-up with CA0_A and CA0_C
- 1.2 V +/- 0.036 V supply for device operation (VDD)
- 1.2 V +/- 0.036 V supply for I/O interface (VDDQ)
- 1.8 V + 0.108 V / - 0.054 V supply for VPP
- 266 ball BGA package with 0.73 x 0.75 mm pitch

2.2 Functional Description

The GDDR7 SGRAM is a high-speed dynamic random-access memory designed for applications requiring high bandwidth. The device's architecture consists of four byte wide fully independent channels. GDDR7 uses a 256b per read and write array access architecture and a DDR PAM3 interface to achieve high-speed operation. In PAM3 mode the byte consists of ten DQ and one DQE signal. See the *PAM3* section for details on high speed I/O signaling. GDDR7 also includes a NRZ I/O signaling mode for low power operation that has a byte definition that includes eight DQ and one DQE.

GDDR7 devices contain the following number of bits:

16 Gb has 17,179,869,184 bits

24 Gb has 25,769,803,776 bits

32 Gb has 34,359,738,368 bits

48 Gb has 51,539,607,552 bits

64 Gb has 68,719,476,736 bits

The GDDR7 SGRAM's high-speed interface is optimized for point-to-point connections to a host controller. On-die termination (ODT) is provided for all high-speed interface signals to eliminate the need for termination resistors in the system.

GDDR7 operates from a differential clock WCK_t and WCK_c. WCK is common to both Command Address (CA) and DQ buses.

Command and Address (CA) are registered at every rising edge of WCK. GDDR7 uses a separate 4n-cycle (nCK4) Row Command bus and 4n-cycle (nCK4) Column Command bus. Row Command Address is 11-bit that contains command, bank address, row address, op code, and etc. via CA[2:0]. Column Command Address is 7-bit that contains command, bank address, column address, and etc. via CA[4:3]. There are both single nCK4 cycle and multi nCK4 cycle commands. See the *COMMAND TRUTH TABLE* section for details.

The row and bank address to be accessed is registered coincident with the Activate command. The address bits registered coincident with the Read or Write command are used to select the bank and the starting column location for the burst access. Read and write accesses to GDDR7 are burst oriented; accesses start at a selected location and consists of a total of sixteen symbols in PAM3 mode and thirty-two data words in NRZ mode. Accesses begin with the registration of an Activate command, which is then followed by a Read or Write command.

GDDR7 include several features to improve the integrity of data including OD-ECC, CRC, hPPR, CA Parity, and CAPAR with Command Blocking. See the *DATA INTEGRITY* section for more details.

Prior to normal operation, the GDDR7 SGRAM is required to be initialized. The *INITIALIZATION* section provides detailed information covering device initialization.

This standard includes all features and functionality required for GDDR7 SGRAM devices both in PAM3 mode and NRZ mode. In many cases the GDDR7 specification describes the behavior of a single channel.

2.3 Definition Of Signal State Terminology

GDDR7 SGRAM will be operated in both ODT Enable (terminated) and ODT Disable (unterminated) modes. For highest data rates it is recommended to operate in the ODT Enable mode. ODT Disable mode is designed to reduce power and may operate at reduced data rates. There exist situations where ODT Enable mode cannot be guaranteed for a short period of time, i.e., during power up.

Following are four terminologies defined for the state of a device (GDDR7 SGRAM or controller) signal during operation. The state of the bus will be determined by the combination of the device signal connected to the bus in the system. For example, in GDDR7 it is possible for the SGRAM signal to be tristate while the controller signal is HIGH or ODT. In both cases the bus would be HIGH if the ODT is enabled. For details on the device's signals and their function see the *SIGNALS* section.

Device signal level:

- HIGH: A device signal is driving the Logic "HIGH" state in NRZ mode and "+1" in PAM3 mode. See the *PAM3* section for more details on PAM3 levels.
- LOW: A device signal is driving the Logic "LOW" state in NRZ mode and "-1" in PAM3 mode.
- High-Z: A device signal is tristate.
- ODT: A device signal terminates with ODT setting, which could be terminating or tristate depending on Mode Register setting.

2.4 Clocking

The GDDR7 SGRAM supports two operating modes for WCK frequency which differ in the DQ[9:0]/DQE to WCK clock frequency ratio. The GDDR7 SGRAM supports NRZ mode and PAM3 mode operating modes for WCK frequency which differ in the DQ[9:0]/DQE to WCK clock frequency ratio.

FIGURE 1 illustrates the difference between a NRZ mode and a PAM3 mode.

Each channel of a GDDR7 SGRAM includes a single differential clock pair WCK_t/WCK_c for latching command/address (CA) inputs, for latching write data and for driving read data. Command and Address (CA) are registered at every rising WCK edge.

A rising WCK edge is defined as the crossing of the positive edge of WCK_t and the negative edge of WCK_c. A falling WCK edge is defined as the crossing of the negative edge of WCK_t and the positive edge of WCK_c.

CK4 (int.) is quarter rate of WCK frequency for internal clocking and for AC timings. For commands that span multiple CK4 cycles the last CK4 cycle of a command is the reference point for all related AC timings and latencies. Timing diagrams are drawn accordingly.

Table 1 — Example of WCK and Interface Signal Frequency Relationship

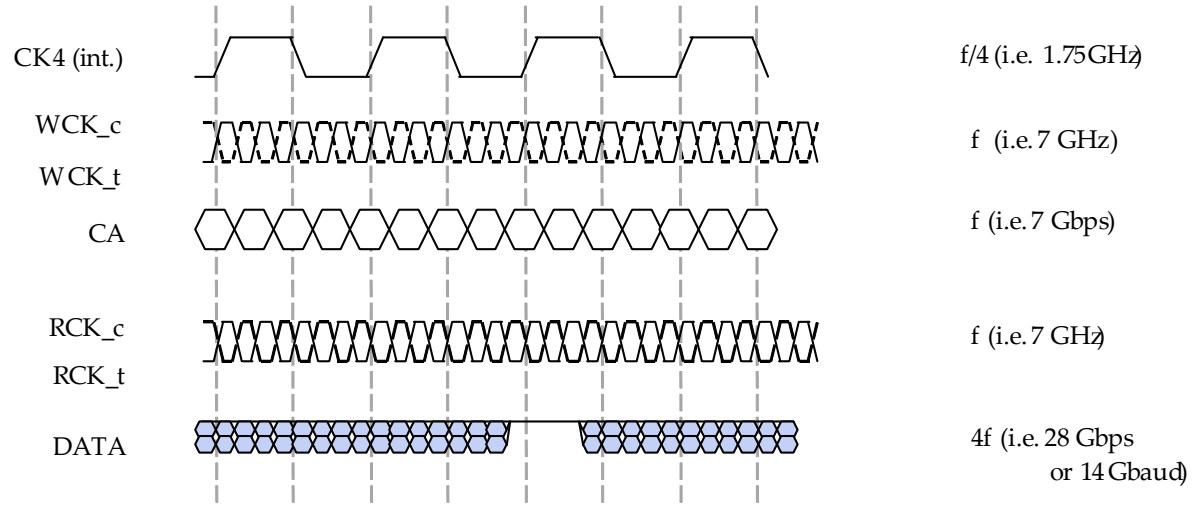
Signal	PAM3 mode	NRZ mode	UNIT	Notes
CK4 (int)	1.75	0.5-	GHz	1
WCK_t, WCK_c RCK_t, RCK_c	7	2	GHz	
CA[4:0]	7	2	Gbps/pin	
DQ[7:0], DQE	-	4	Gbps/pin	
	14	-	Gbaud/pin	2
DQ[9:8]	14	-	Gbaud/pin	2

NOTE 1 CK4 (int.) is an internal clock that is referenced e.g., by cycle based AC Timings and latencies.

NOTE 2 The 14 GBaud in PAM3 mode correspond to a data rate of 28 Gbps over 8 DQs or a per-channel memory bandwidth of 28 GB/s.

2.4 Clocking (cont'd)

PAM3 mode



NRZ mode

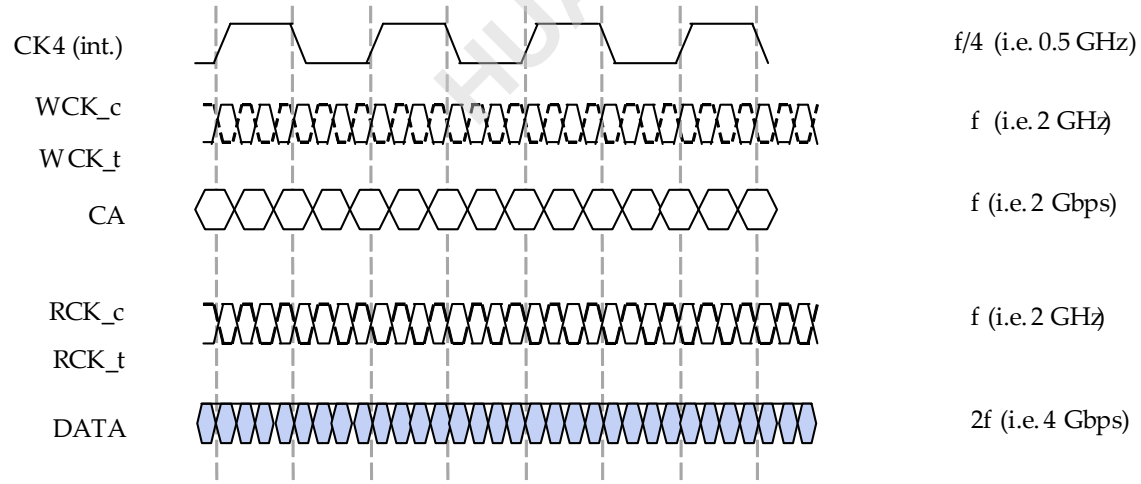


Figure 1 — GDDR7 Clocking and Interface Relationship

2.4 Clocking (cont'd)

The GDDR7 SGRAM supports two operating modes for WCK frequency which differ in the DQ[9:0]/DQE to WCK clock frequency ratio. The GDDR7 SGRAM supports NRZ mode and PAM3 mode operating modes for WCK frequency which differ in the DQ[9:0]/DQE to WCK clock frequency ratio.

FIGURE 1 illustrates the difference between a NRZ mode and a PAM3 mode.

Each channel of a GDDR7 SGRAM includes a single differential clock pair WCK_t/WCK_c for latching command/address (CA) inputs, for latching write data and for driving read data. Command and Address (CA) are registered at every rising WCK edge.

A rising WCK edge is defined as the crossing of the positive edge of WCK_t and the negative edge of WCK_c. A falling WCK edge is defined as the crossing of the negative edge of WCK_t and the positive edge of WCK_c.

CK4 (int.) is quarter rate of WCK frequency for internal clocking and for AC timings.

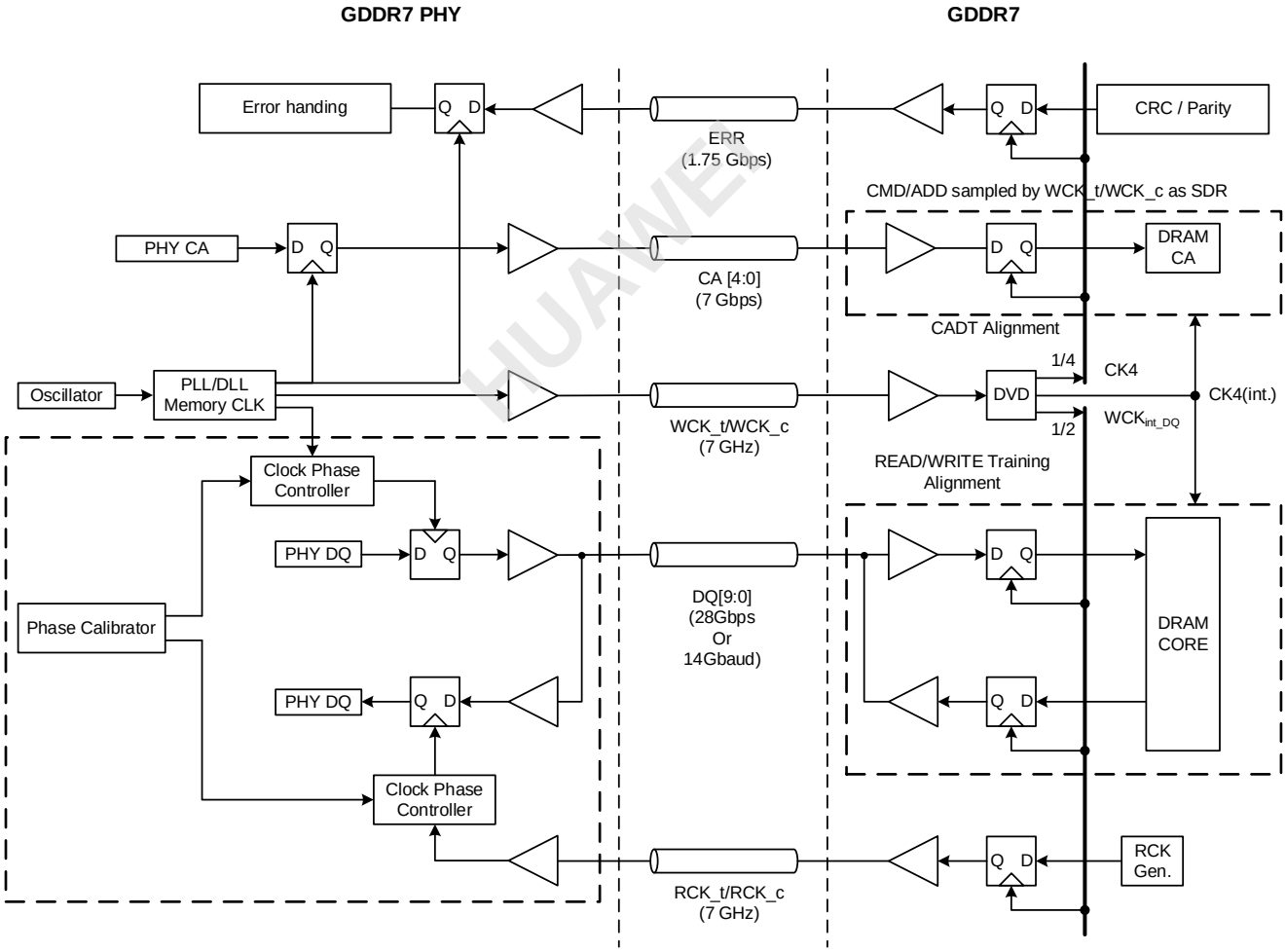


Figure 2 — Block Diagram of an Example Clock System

2.5 Addressing

GDDR7 SGRAMs use a single data rate command address scheme on the device. The command and addresses is packetized on 5 CA signals (CA[4:0]) over either single CK4 cycle or multi CK4 cycles depending on the command (see *COMMAND TRUTH TABLE*).

To enable higher performance, GDDR7 SGRAMs provide semi-independent row and column command interfaces for each channel. These interfaces increase command bandwidth by allowing read and write commands to be issued simultaneously with other commands like Activate and Precharge commands. See *ROW AND COLUMN COMMANDS*.

GDDR7 addressing is defined for a single channel with devices being configured to either 2 or 4 channels/device.

Table 2 — Addressing Scheme

Memory Density	16 Gb		24 Gb		32 Gb		48 Gb		64 Gb	
	17,179,869,184		25,769,803,776		34,359,738,368		51,539,607,552		68,719,476,736	
Device Organization	**4 CH Mode**	**2 CH Mode**	**4 CH Mode**	**2 CH Mode**	**4 CH Mode**	**2 CH Mode**	**4 CH Mode**	**2 CH Mode**	**4 CH Mode**	**2 CH Mode**
Number channels	4	2	4	2	4	2	4	2	4	2
Channel Density	4 Gb	8 Gb	6 Gb	12 Gb	8 Gb	16 Gb	12 Gb	24 Gb	16 Gb	32 Gb
Array Pre-Fetch (bits, per channel)	256		256		256		256		256	
Bank address (per channel)	BA[3:0]		BA[3:0]		BA[3:0]		BA[3:0]		BA[3:0]	
Number Banks (per channel)	16		16		16		16		16	
Row address (per channel)	R[13:0]	R[14:0]	R[14:0]	R[15:0]	R[14:0]	R[15:0]	TBD	TBD	TBD	TBD
Number of Rows (per channel)	16,384	32,768	24,576	49,152	32,768	65,536	TBD	TBD	TBD	TBD
Column address (per channel)	C[5:0]	C[5:0]	C[5:0]	C[5:0]	C[5:0]	C[5:0]	TBD	TBD	TBD	TBD
Page Size (per channel)	2K	2K	2K	2K	2K	2K	TBD	TBD	TBD	TBD

NOTES:

1. The column address notation for GDDR7 does not include addressing within the prefetch of 256 bits as the burst order is always fixed for READ and WRITE commands.
2. Page Size = 2^COLBITS * (Prefetch_Size/8) where COLBITS is the number of column address bits.
3. Row address range with R[15:14] = 11 for 2 CH mode and R[14:13] = 11 for 4 CH mode is not present for 24 Gb density.

2.6 Command Address Bus Inversion (CABI)

Command Address Bus Inversion (CABI) reduces the power requirements on Command Address (CA) bus, as the number of CA lines driving a LOW level can be limited to 10-bit in 20-bit of 4-cycle Row and Column CA per channel.

The Command Address Bus Inversion function is associated with the electrical signaling on the CA lines between a controller and the device, regardless of whether the information conveyed on the CA lines is a row or column address, a command, a mode register op-code, par, or any other pattern.

Once enabled by the corresponding CABI Mode Register bit, the GDDR7 SGRAM will invert the pattern received on the CA inputs in case CABI was sampled LOW, or leave the pattern non-inverted in case CABI was sampled HIGH, as shown in *FIGURE 3*. If CABI is disabled (MR0 OP0=1), CABI (CA1 of 0th clock cycle) bit is "Valid" (V) but it is recommended to be set to HIGH.

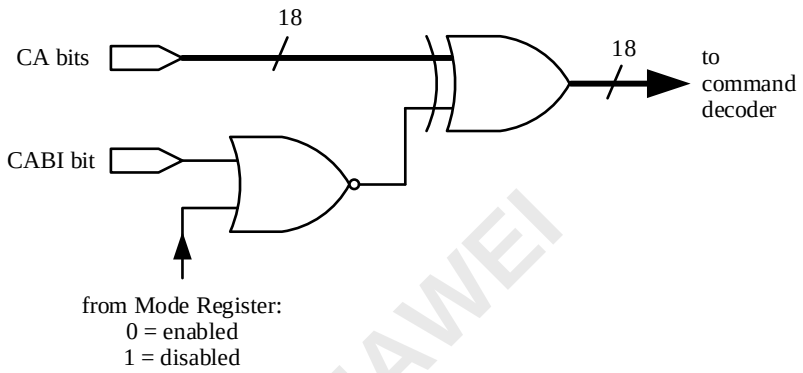


Figure 3 — Example of Command Address Bus Inversion Logic

2.7 Command Address Parity (CAPAR) Definition

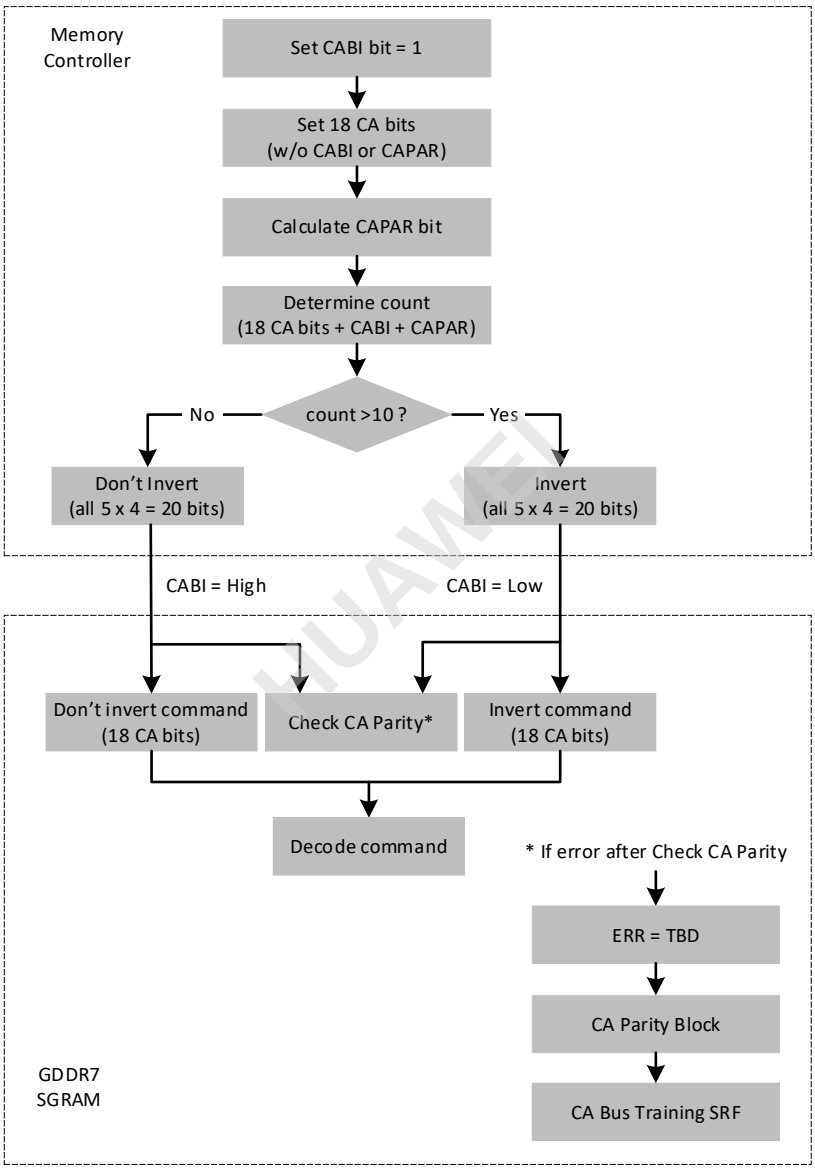
The convention of parity is even parity, i.e., valid parity is defined as an even number of ones across the inputs used for parity computation combined with the parity signal. In other words, the parity bit is chosen so that the total number of 1's in the 20-bit, including the parity bit is even. If CAPAR is disabled (MR15 OP0=0), CAPAR bit (CA4 of 3rd clock cycle) is "Valid" (V) but it is recommended to be set to HIGH. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.

Table 3 — Example of CAPAR (Even Parity)

'1' count	Command Address (19-bit)	CAPAR
even	000 0000 0000 1111 1111	LOW
odd	000 0000 0001 1111 1111	HIGH

2.8 CABI and CAPAR Order

The flow diagram in *FIGURE 4* illustrates the CABI and CAPAR operation. The controller decides whether to invert or not invert the data conveyed and then make even parity with CAPAR on the CA lines. The GDDR7 SGRAM must perform the reverse operation based on even parity with CAPAR and the level of CABI. CA input timing parameters are only valid with CABI being enabled and a maximum of 9 CA inputs driven LOW in 4-cycle CA BUS.



NOTE 1 18-bit is 4-cycle of CA bus excluding CABI (0^{th} clock cycle CA1) and CAPAR (3^{rd} clock cycle CA4)
NOTE 2 CABI (MR0 OP0), CAPAR (MR015, OP0) and CAPARBLK (MR015 OP1) are enabled for illustration
NOTE 3 If CABI and CAPAR are disabled, CABI bit & PAR bit are "Valid" (V) but they are recommended to be set to HIGH.

Figure 4 — Command Address Bus Inversion (CABI) and Command Address Parity (CAPAR) Flow Diagram

2.9 PAM3

2.9.1 PAM3 Signaling and Encoding Conventions

The GDDR7 SGRAMs can be configured to use a PAM3 I/O system for the data interface (DQ[9:0], DQE). PAM3 works with point-to-point signaling between driver and receiver. PAM3 is enabled by setting MR0 OP8 to 1.

The PAM3 signaling transmits a ternary digit (or trit) per U.I. using three distinct signal levels in a balanced signed-digit representation (+1, 0 , -1). Each PAM3 symbol is internally represented by two bits, although only three combinations are valid as shown in *TABLE 4*.

Table 4 — PAM3 Signal Levels

Binary [MSB:LSB]	PAM3 Symbol (Ternary Digit)	Nominal Voltage Level [%V_{DDQ}][1]
1 1	+1	100.0%
1 0	Invalid	-
0 1	0	75%
0 0	-1	50%
NOTE 1 Refer to the *OPERATING CONDITIONS* Section for absolute VDDQ voltage levels.		

2.9.2 PAM3 Burst Encoding

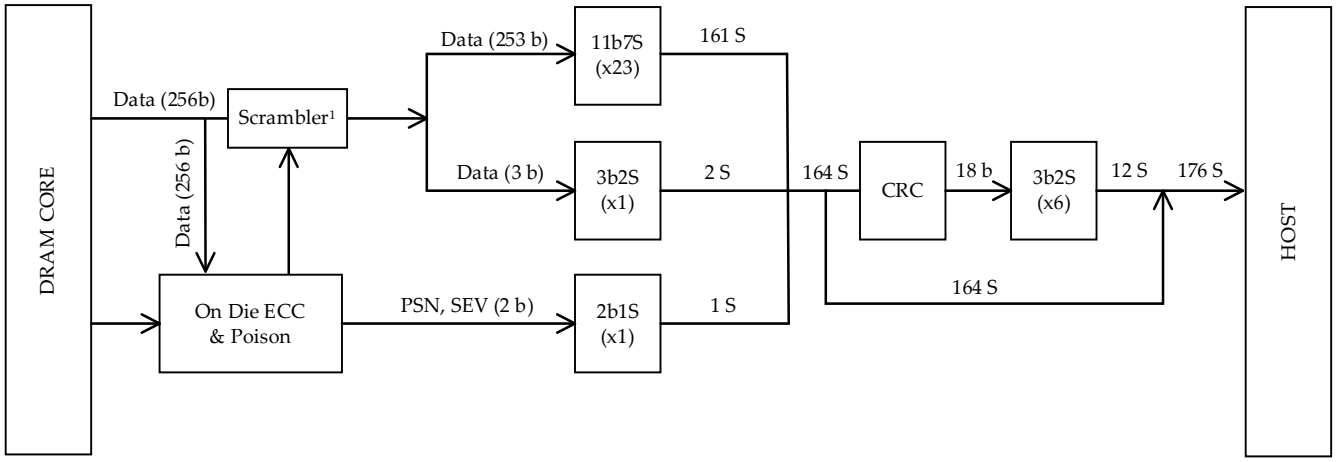
In PAM3 mode GDDR7 SGRAMs transfer a total of 176 symbols per burst access over 11 data lines (BL 16 x 11 DQs = 176 Symbols). Write data is PAM3 decoded before written to the memory array, read data is PAM3 encoded before it is transmitted. The maximum burst data payload is 276 bits, as in the case of read direction when CRC and poison are enabled (256 data bits + 18 CRC bits + 1 Severity bit + 1 Poison bit).

To avoid CRC error multiplication, reduce static termination power and to maximize the relationship between encoding efficiency and implementation simplicity, the GDDR7 makes use of different types of PAM3 encoders/decoders, as shown in *FIGURE 5* and *FIGURE 6*.

- **11b7S**: eleven-bits-to-seven-symbols encoder/decoder for efficient data coding. The encoder/decoder truth table is shown in *TABLE 7* and a recommendation for efficient implementation of the encoder/decoder is provided and explained in *SECTION 2.9.3*.
- **3b2S**: three bits-to-two-symbols encoder/decoder for DC balanced encoding, as defined in *TABLE 5*. Instances of this encoder are used to encode the CRC (in PAM3 and NRZ modes), and in PAM3 mode for LFSR training and the 3 bits of data remainder on DQ4 (see *FIGURE 7*).
- **2b1S**: two-bits-to-1-symbol encoder/decoder for Severity and Poison, as in *TABLE 6*. Note that SEV/PSN are mutually exclusive, Severity prevails over Poison.

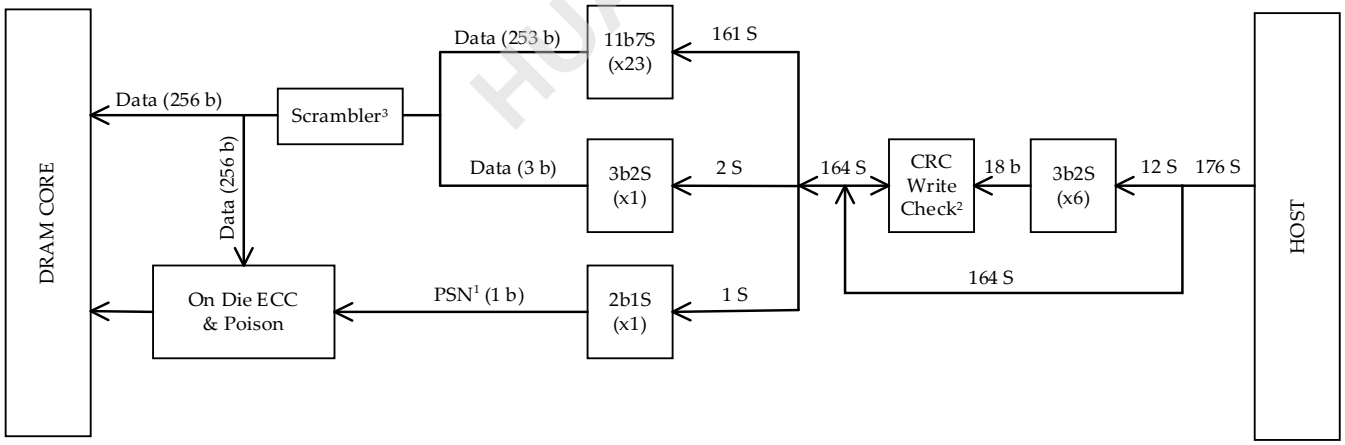
The total burst transfer payload per channel is encoded using 23 x 11b7S and 1 x 3b2S for the data, 6 x 3b2S for the CRC and 1 x 2b1S for the SEV/PSN, it adds up to the 176 PAM3 symbols that can be allocated for a 16 burst over 11 data lines. The logical to physical lane mapping, burst order and bit-to-symbol encoder grouping is shown in *FIGURE 7*.

2.9 PAM3 (cont'd)



NOTE 1: The scrambler block performs a bitwise XOR operation on the array read data before PAM3 encoding using the scrambler code as programmed in MR20. Please refer to the *STATIC DATA SCRAMBLER* section for details.

Figure 5 —Burst Data Payload Encoding Diagram (Read Operation)



NOTE 1 Severity flag is not transmitted during write operation.

NOTE 2 During Writes, the DRAM computes the CRC on the received data and the result is compared with the 18 bits CRC received from the host, a mismatch is reported back to the host on the ERR pin as CRC Write Error.

NOTE 3 The scrambler block performs a bitwise XOR operation on the data after PAM3 decoding before writing it to the array using the scrambler code as programmed in MR20. Please refer to the *STATIC DATA SCRAMBLER* section for details.

Figure 6 —Burst Data Payload Decoding Diagram (Write Operation)

2.9 PAM3 (cont'd)

Table 5 — 3b2S Encoder for CRC, Data Remainder, and LFSR Training

3b2S PAM3 Encoder/Decoder Truth Table								
3 Bits			Internal Binary Representation				2 Trits	
MSB		LSB	S1 MSB	S1 LSB	S0 MSB	S0 LSB		
b2	b1	b0	b3	b2	b1	b0	S1	S0
0	0	0	0	1	0	0	0	-1
0	0	1	0	0	1	1	-1	+1
0	1	0	1	1	0	0	+1	-1
0	1	1	0	1	1	1	0	+1
Invalid			0	1	0	1	0	0
1	0	0	0	0	0	0	-1	-1
1	0	1	0	0	0	1	-1	0
1	1	0	1	1	0	1	+1	0
1	1	1	1	1	1	1	+1	+1

Table 6 — Severity and Poison 2b1S Encoder/Decoder

2b1S PAM3 Encoder/Decoder for SEV/PSN				
2 Bits		Binary		1 Trit
SEV	PSN	b1	b0	S0
1	X	0	0	-1
0	1	0	1	0
0	0	1	1	+1

NOTE 1 During read operation Severity has precedence over Poison, therefore Poison flag is "don't care" when Severity flag is set. During write operation SEV flag is always 0, hence only +1 and 0 input levels are allowed.

NOTE 2 MR0 OP9 (Severity) shall be turned on when MR0 OP10 (Poison) is enabled. When Poison is enabled, it is enabled for both RD and WR operations. When Poison and/or Severity are disabled, their default value will be set to '0'.

2.9 PAM3 (cont'd)

Table 7 — 11b7S Encoder/Decoder Table

11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x000	0x447	0x100	0x1047	0x200	0xc47	0x300	0x3047	0x400	0x1447	0x500	0x3447	0x600	0x1c47	0x700	0x3c47
0x001	0x444	0x101	0x1044	0x201	0xc44	0x301	0x3044	0x401	0x1444	0x501	0x3444	0x601	0x1c44	0x701	0x3c44
0x002	0x445	0x102	0x1045	0x202	0xc45	0x302	0x3045	0x402	0x1445	0x502	0x3445	0x602	0x1c45	0x702	0x3c45
0x003	0x441	0x103	0x1041	0x203	0xc41	0x303	0x3041	0x403	0x1441	0x503	0x3441	0x603	0x1c41	0x703	0x3c41
0x004	0x453	0x104	0x1053	0x204	0xc53	0x304	0x3053	0x404	0x1453	0x504	0x3453	0x604	0x1c53	0x704	0x3c53
0x005	0x450	0x105	0x1050	0x205	0xc50	0x305	0x3050	0x405	0x1450	0x505	0x3450	0x605	0x1c50	0x705	0x3c50
0x006	0x451	0x106	0x1051	0x206	0xc51	0x306	0x3051	0x406	0x1451	0x506	0x3451	0x606	0x1c51	0x706	0x3c51
0x007	0x405	0x107	0x1005	0x207	0xc05	0x307	0x3005	0x407	0x1405	0x507	0x3405	0x607	0x1c05	0x707	0x3c05
0x008	0x44f	0x108	0x104f	0x208	0xc4f	0x308	0x304f	0x408	0x144f	0x508	0x344f	0x608	0x1c4f	0x708	0x3c4f
0x009	0x44c	0x109	0x104c	0x209	0xc4c	0x309	0x304c	0x409	0x144c	0x509	0x344c	0x609	0x1c4c	0x709	0x3c4c
0x00a	0x44d	0x10a	0x104d	0x20a	0xc4d	0x30a	0x304d	0x40a	0x144d	0x50a	0x344d	0x60a	0x1c4d	0x70a	0x3c4d
0x00b	0x407	0x10b	0x1007	0x20b	0xc07	0x30b	0x3007	0x40b	0x1407	0x50b	0x3407	0x60b	0x1c07	0x70b	0x3c07
0x00c	0x473	0x10c	0x1073	0x20c	0xc73	0x30c	0x3073	0x40c	0x1473	0x50c	0x3473	0x60c	0x1c73	0x70c	0x3c73
0x00d	0x470	0x10d	0x1070	0x20d	0xc70	0x30d	0x3070	0x40d	0x1470	0x50d	0x3470	0x60d	0x1c70	0x70d	0x3c70
0x00e	0x471	0x10e	0x1071	0x20e	0xc71	0x30e	0x3071	0x40e	0x1471	0x50e	0x3471	0x60e	0x1c71	0x70e	0x3c71
0x00f	0x404	0x10f	0x1004	0x20f	0xc04	0x30f	0x3004	0x40f	0x1404	0x50f	0x3404	0x60f	0x1c04	0x70f	0x3c04
0x010	0x457	0x110	0x1057	0x210	0xc57	0x310	0x3057	0x410	0x1457	0x510	0x3457	0x610	0x1c57	0x710	0x3c57
0x011	0x454	0x111	0x1054	0x211	0xc54	0x311	0x3054	0x411	0x1454	0x511	0x3454	0x611	0x1c54	0x711	0x3c54
0x012	0x455	0x112	0x1055	0x212	0xc55	0x312	0x3055	0x412	0x1455	0x512	0x3455	0x612	0x1c55	0x712	0x3c55
0x013	0x443	0x113	0x1043	0x213	0xc43	0x313	0x3043	0x413	0x1443	0x513	0x3443	0x613	0x1c43	0x713	0x3c43
0x014	0x477	0x114	0x1077	0x214	0xc77	0x314	0x3077	0x414	0x1477	0x514	0x3477	0x614	0x1c77	0x714	0x3c77
0x015	0x474	0x115	0x1074	0x215	0xc74	0x315	0x3074	0x415	0x1474	0x515	0x3474	0x615	0x1c74	0x715	0x3c74
0x016	0x475	0x116	0x1075	0x216	0xc75	0x316	0x3075	0x416	0x1475	0x516	0x3475	0x616	0x1c75	0x716	0x3c75
0x017	0x411	0x117	0x1011	0x217	0xc11	0x317	0x3011	0x417	0x1411	0x517	0x3411	0x617	0x1c11	0x717	0x3c11
0x018	0x45f	0x118	0x105f	0x218	0xc5f	0x318	0x305f	0x418	0x145f	0x518	0x345f	0x618	0x1c5f	0x718	0x3c5f
0x019	0x45c	0x119	0x105c	0x219	0xc5c	0x319	0x305c	0x419	0x145c	0x519	0x345c	0x619	0x1c5c	0x719	0x3c5c
0x01a	0x45d	0x11a	0x105d	0x21a	0xc5d	0x31a	0x305d	0x41a	0x145d	0x51a	0x345d	0x61a	0x1c5d	0x71a	0x3c5d
0x01b	0x413	0x11b	0x1013	0x21b	0xc13	0x31b	0x3013	0x41b	0x1413	0x51b	0x3413	0x61b	0x1c13	0x71b	0x3c13
0x01c	0x47f	0x11c	0x107f	0x21c	0xc7f	0x31c	0x307f	0x41c	0x147f	0x51c	0x347f	0x61c	0x1c7f	0x71c	0x3c7f
0x01d	0x47c	0x11d	0x107c	0x21d	0xc7c	0x31d	0x307c	0x41d	0x147c	0x51d	0x347c	0x61d	0x1c7c	0x71d	0x3c7c
0x01e	0x47d	0x11e	0x107d	0x21e	0xc7d	0x31e	0x307d	0x41e	0x147d	0x51e	0x347d	0x61e	0x1c7d	0x71e	0x3c7d
0x01f	0x410	0x11f	0x1010	0x21f	0xc10	0x31f	0x3010	0x41f	0x1410	0x51f	0x3410	0x61f	0x1c10	0x71f	0x3c10
0x020	0x507	0x120	0x1107	0x220	0xd07	0x320	0x3107	0x420	0x1507	0x520	0x3507	0x620	0x1d07	0x720	0x3d07
0x021	0x504	0x121	0x1104	0x221	0xd04	0x321	0x3104	0x421	0x1504	0x521	0x3504	0x621	0x1d04	0x721	0x3d04
0x022	0x505	0x122	0x1105	0x222	0xd05	0x322	0x3105	0x422	0x1505	0x522	0x3505	0x622	0x1d05	0x722	0x3d05
0x023	0x501	0x123	0x1101	0x223	0xd01	0x323	0x3101	0x423	0x1501	0x523	0x3501	0x623	0x1d01	0x723	0x3d01
0x024	0x513	0x124	0x1113	0x224	0xd13	0x324	0x3113	0x424	0x1513	0x524	0x3513	0x624	0x1d13	0x724	0x3d13

Table 7 — 11b7S Encoder/Decoder Table (cont'd)

11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x025	0x510	0x125	0x1110	0x225	0xd10	0x325	0x3110	0x425	0x1510	0x525	0x3510	0x625	0x1d10	0x725	0x3d10
0x026	0x511	0x126	0x1111	0x226	0xd11	0x326	0x3111	0x426	0x1511	0x526	0x3511	0x626	0x1d11	0x726	0x3d11
0x027	0x40d	0x127	0x100d	0x227	0xc0d	0x327	0x300d	0x427	0x140d	0x527	0x340d	0x627	0x1c0d	0x727	0x3c0d
0x028	0x50f	0x128	0x110f	0x228	0xd0f	0x328	0x310f	0x428	0x150f	0x528	0x350f	0x628	0x1d0f	0x728	0x3d0f
0x029	0x50c	0x129	0x110c	0x229	0xd0c	0x329	0x310c	0x429	0x150c	0x529	0x350c	0x629	0x1d0c	0x729	0x3d0c
0x02a	0x50d	0x12a	0x110d	0x22a	0xd0d	0x32a	0x310d	0x42a	0x150d	0x52a	0x350d	0x62a	0x1d0d	0x72a	0x3d0d
0x02b	0x40f	0x12b	0x100f	0x22b	0xc0f	0x32b	0x300f	0x42b	0x140f	0x52b	0x340f	0x62b	0x1c0f	0x72b	0x3c0f
0x02c	0x533	0x12c	0x1133	0x22c	0xd33	0x32c	0x3133	0x42c	0x1533	0x52c	0x3533	0x62c	0x1d33	0x72c	0x3d33
0x02d	0x530	0x12d	0x1130	0x22d	0xd30	0x32d	0x3130	0x42d	0x1530	0x52d	0x3530	0x62d	0x1d30	0x72d	0x3d30
0x02e	0x531	0x12e	0x1131	0x22e	0xd31	0x32e	0x3131	0x42e	0x1531	0x52e	0x3531	0x62e	0x1d31	0x72e	0x3d31
0x02f	0x40c	0x12f	0x100c	0x22f	0xc0c	0x32f	0x300c	0x42f	0x140c	0x52f	0x340c	0x62f	0x1c0c	0x72f	0x3c0c
0x030	0x517	0x130	0x1117	0x230	0xd17	0x330	0x3117	0x430	0x1517	0x530	0x3517	0x630	0x1d17	0x730	0x3d17
0x031	0x514	0x131	0x1114	0x231	0xd14	0x331	0x3114	0x431	0x1514	0x531	0x3514	0x631	0x1d14	0x731	0x3d14
0x032	0x515	0x132	0x1115	0x232	0xd15	0x332	0x3115	0x432	0x1515	0x532	0x3515	0x632	0x1d15	0x732	0x3d15
0x033	0x503	0x133	0x1103	0x233	0xd03	0x333	0x3103	0x433	0x1503	0x533	0x3503	0x633	0x1d03	0x733	0x3d03
0x034	0x537	0x134	0x1137	0x234	0xd37	0x334	0x3137	0x434	0x1537	0x534	0x3537	0x634	0x1d37	0x734	0x3d37
0x035	0x534	0x135	0x1134	0x235	0xd34	0x335	0x3134	0x435	0x1534	0x535	0x3534	0x635	0x1d34	0x735	0x3d34
0x036	0x535	0x136	0x1135	0x236	0xd35	0x336	0x3135	0x436	0x1535	0x536	0x3535	0x636	0x1d35	0x736	0x3d35
0x037	0x431	0x137	0x1031	0x237	0xc31	0x337	0x3031	0x437	0x1431	0x537	0x3431	0x637	0x1c31	0x737	0x3c31
0x038	0x51f	0x138	0x111f	0x238	0xd1f	0x338	0x311f	0x438	0x151f	0x538	0x351f	0x638	0x1d1f	0x738	0x3d1f
0x039	0x51c	0x139	0x111c	0x239	0xd1c	0x339	0x311c	0x439	0x151c	0x539	0x351c	0x639	0x1d1c	0x739	0x3d1c
0x03a	0x51d	0x13a	0x111d	0x23a	0xd1d	0x33a	0x311d	0x43a	0x151d	0x53a	0x351d	0x63a	0x1d1d	0x73a	0x3d1d
0x03b	0x433	0x13b	0x1033	0x23b	0xc33	0x33b	0x3033	0x43b	0x1433	0x53b	0x3433	0x63b	0x1c33	0x73b	0x3c33
0x03c	0x53f	0x13c	0x113f	0x23c	0xd3f	0x33c	0x313f	0x43c	0x153f	0x53c	0x353f	0x63c	0x1d3f	0x73c	0x3d3f
0x03d	0x53c	0x13d	0x113c	0x23d	0xd3c	0x33d	0x313c	0x43d	0x153c	0x53d	0x353c	0x63d	0x1d3c	0x73d	0x3d3c
0x03e	0x53d	0x13e	0x113d	0x23e	0xd3d	0x33e	0x313d	0x43e	0x153d	0x53e	0x353d	0x63e	0x1d3d	0x73e	0x3d3d
0x03f	0x430	0x13f	0x1030	0x23f	0xc30	0x33f	0x3030	0x43f	0x1430	0x53f	0x3430	0x63f	0x1c30	0x73f	0x3c30
0x040	0x4c7	0x140	0x10c7	0x240	0xcc7	0x340	0x30c7	0x440	0x14c7	0x540	0x34c7	0x640	0x1cc7	0x740	0x3cc7
0x041	0x4c4	0x141	0x10c4	0x241	0xcc4	0x341	0x30c4	0x441	0x14c4	0x541	0x34c4	0x641	0x1cc4	0x741	0x3cc4
0x042	0x4c5	0x142	0x10c5	0x242	0xcc5	0x342	0x30c5	0x442	0x14c5	0x542	0x34c5	0x642	0x1cc5	0x742	0x3cc5
0x043	0x4c1	0x143	0x10c1	0x243	0xcc1	0x343	0x30c1	0x443	0x14c1	0x543	0x34c1	0x643	0x1cc1	0x743	0x3cc1
0x044	0x4d3	0x144	0x10d3	0x244	0xcd3	0x344	0x30d3	0x444	0x14d3	0x544	0x34d3	0x644	0x1cd3	0x744	0x3cd3
0x045	0x4d0	0x145	0x10d0	0x245	0xcd0	0x345	0x30d0	0x445	0x14d0	0x545	0x34d0	0x645	0x1cd0	0x745	0x3cd0
0x046	0x4d1	0x146	0x10d1	0x246	0xcd1	0x346	0x30d1	0x446	0x14d1	0x546	0x34d1	0x646	0x1cd1	0x746	0x3cd1
0x047	0x415	0x147	0x1015	0x247	0xc15	0x347	0x3015	0x447	0x1415	0x547	0x3415	0x647	0x1c15	0x747	0x3c15
0x048	0x4cf	0x148	0x10cf	0x248	0xccf	0x348	0x30cf	0x448	0x14cf	0x548	0x34cf	0x648	0x1ccf	0x748	0x3ccf
0x049	0x4cc	0x149	0x10cc	0x249	0xccc	0x349	0x30cc	0x449	0x14cc	0x549	0x34cc	0x649	0x1ccc	0x749	0x3ccc
0x04a	0x4cd	0x14a	0x10cd	0x24a	0xccd	0x34a	0x30cd	0x44a	0x14cd	0x54a	0x34cd	0x64a	0x1ccd	0x74a	0x3ccd

Table 7 — 11b7S Encoder/Decoder Table (cont'd)

11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x04b	0x417	0x14b	0x1017	0x24b	0xc17	0x34b	0x3017	0x44b	0x1417	0x54b	0x3417	0x64b	0x1c17	0x74b	0x3c17
0x04c	0x4f3	0x14c	0x10f3	0x24c	0xcf3	0x34c	0x30f3	0x44c	0x14f3	0x54c	0x34f3	0x64c	0x1cf3	0x74c	0x3cf3
0x04d	0x4f0	0x14d	0x10f0	0x24d	0xcf0	0x34d	0x30f0	0x44d	0x14f0	0x54d	0x34f0	0x64d	0x1cf0	0x74d	0x3cf0
0x04e	0x4f1	0x14e	0x10f1	0x24e	0xcf1	0x34e	0x30f1	0x44e	0x14f1	0x54e	0x34f1	0x64e	0x1cf1	0x74e	0x3cf1
0x04f	0x414	0x14f	0x1014	0x24f	0xc14	0x34f	0x3014	0x44f	0x1414	0x54f	0x3414	0x64f	0x1c14	0x74f	0x3c14
0x050	0x4d7	0x150	0x10d7	0x250	0xcd7	0x350	0x30d7	0x450	0x14d7	0x550	0x34d7	0x650	0x1cd7	0x750	0x3cd7
0x051	0x4d4	0x151	0x10d4	0x251	0xcd4	0x351	0x30d4	0x451	0x14d4	0x551	0x34d4	0x651	0x1cd4	0x751	0x3cd4
0x052	0x4d5	0x152	0x10d5	0x252	0xcd5	0x352	0x30d5	0x452	0x14d5	0x552	0x34d5	0x652	0x1cd5	0x752	0x3cd5
0x053	0x4c3	0x153	0x10c3	0x253	0xcc3	0x353	0x30c3	0x453	0x14c3	0x553	0x34c3	0x653	0x1cc3	0x753	0x3cc3
0x054	0x4f7	0x154	0x10f7	0x254	0xcf7	0x354	0x30f7	0x454	0x14f7	0x554	0x34f7	0x654	0x1cf7	0x754	0x3cf7
0x055	0x4f4	0x155	0x10f4	0x255	0xcf4	0x355	0x30f4	0x455	0x14f4	0x555	0x34f4	0x655	0x1cf4	0x755	0x3cf4
0x056	0x4f5	0x156	0x10f5	0x256	0xcf5	0x356	0x30f5	0x456	0x14f5	0x556	0x34f5	0x656	0x1cf5	0x756	0x3cf5
0x057	0x435	0x157	0x1035	0x257	0xc35	0x357	0x3035	0x457	0x1435	0x557	0x3435	0x657	0x1c35	0x757	0x3c35
0x058	0x4df	0x158	0x10df	0x258	0xcdf	0x358	0x30df	0x458	0x14df	0x558	0x34df	0x658	0x1cdf	0x758	0x3cdf
0x059	0x4dc	0x159	0x10dc	0x259	0xcdc	0x359	0x30dc	0x459	0x14dc	0x559	0x34dc	0x659	0x1cdc	0x759	0x3cdc
0x05a	0x4dd	0x15a	0x10dd	0x25a	0xcdd	0x35a	0x30dd	0x45a	0x14dd	0x55a	0x34dd	0x65a	0x1cdd	0x75a	0x3cdd
0x05b	0x437	0x15b	0x1037	0x25b	0xc37	0x35b	0x3037	0x45b	0x1437	0x55b	0x3437	0x65b	0x1c37	0x75b	0x3c37
0x05c	0x4ff	0x15c	0x10ff	0x25c	0xcff	0x35c	0x30ff	0x45c	0x14ff	0x55c	0x34ff	0x65c	0x1cff	0x75c	0x3cff
0x05d	0x4fc	0x15d	0x10fc	0x25d	0xcfc	0x35d	0x30fc	0x45d	0x14fc	0x55d	0x34fc	0x65d	0x1cfc	0x75d	0x3cfc
0x05e	0x4fd	0x15e	0x10fd	0x25e	0xcfd	0x35e	0x30fd	0x45e	0x14fd	0x55e	0x34fd	0x65e	0x1cfd	0x75e	0x3cfd
0x05f	0x434	0x15f	0x1034	0x25f	0xc34	0x35f	0x3034	0x45f	0x1434	0x55f	0x3434	0x65f	0x1c34	0x75f	0x3c34
0x060	0x707	0x160	0x1307	0x260	0xf07	0x360	0x3307	0x460	0x1707	0x560	0x3707	0x660	0x1f07	0x760	0x3f07
0x061	0x704	0x161	0x1304	0x261	0xf04	0x361	0x3304	0x461	0x1704	0x561	0x3704	0x661	0x1f04	0x761	0x3f04
0x062	0x705	0x162	0x1305	0x262	0xf05	0x362	0x3305	0x462	0x1705	0x562	0x3705	0x662	0x1f05	0x762	0x3f05
0x063	0x701	0x163	0x1301	0x263	0xf01	0x363	0x3301	0x463	0x1701	0x563	0x3701	0x663	0x1f01	0x763	0x3f01
0x064	0x713	0x164	0x1313	0x264	0xf13	0x364	0x3313	0x464	0x1713	0x564	0x3713	0x664	0x1f13	0x764	0x3f13
0x065	0x710	0x165	0x1310	0x265	0xf10	0x365	0x3310	0x465	0x1710	0x565	0x3710	0x665	0x1f10	0x765	0x3f10
0x066	0x711	0x166	0x1311	0x266	0xf11	0x366	0x3311	0x466	0x1711	0x566	0x3711	0x666	0x1f11	0x766	0x3f11
0x067	0x41d	0x167	0x101d	0x267	0xc1d	0x367	0x301d	0x467	0x141d	0x567	0x341d	0x667	0x1c1d	0x767	0x3c1d
0x068	0x70f	0x168	0x130f	0x268	0xf0f	0x368	0x330f	0x468	0x170f	0x568	0x370f	0x668	0x1f0f	0x768	0x3f0f
0x069	0x70c	0x169	0x130c	0x269	0xf0c	0x369	0x330c	0x469	0x170c	0x569	0x370c	0x669	0x1f0c	0x769	0x3f0c
0x06a	0x70d	0x16a	0x130d	0x26a	0xf0d	0x36a	0x330d	0x46a	0x170d	0x56a	0x370d	0x66a	0x1f0d	0x76a	0x3f0d
0x06b	0x41f	0x16b	0x101f	0x26b	0xc1f	0x36b	0x301f	0x46b	0x141f	0x56b	0x341f	0x66b	0x1c1f	0x76b	0x3c1f
0x06c	0x733	0x16c	0x1333	0x26c	0xf33	0x36c	0x3333	0x46c	0x1733	0x56c	0x3733	0x66c	0x1f33	0x76c	0x3f33
0x06d	0x730	0x16d	0x1330	0x26d	0xf30	0x36d	0x3330	0x46d	0x1730	0x56d	0x3730	0x66d	0x1f30	0x76d	0x3f30
0x06e	0x731	0x16e	0x1331	0x26e	0xf31	0x36e	0x3331	0x46e	0x1731	0x56e	0x3731	0x66e	0x1f31	0x76e	0x3f31
0x06f	0x41c	0x16f	0x101c	0x26f	0xc1c	0x36f	0x301c	0x46f	0x141c	0x56f	0x341c	0x66f	0x1c1c	0x76f	0x3c1c
0x070	0x717	0x170	0x1317	0x270	0xf17	0x370	0x3317	0x470	0x1717	0x570	0x3717	0x670	0x1f17	0x770	0x3f17

Table 7 — 11b7S Encoder/Decoder Table (cont'd)

11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x071	0x714	0x171	0x1314	0x271	0xf14	0x371	0x3314	0x471	0x1714	0x571	0x3714	0x671	0x1f14	0x771	0x3f14
0x072	0x715	0x172	0x1315	0x272	0xf15	0x372	0x3315	0x472	0x1715	0x572	0x3715	0x672	0x1f15	0x772	0x3f15
0x073	0x703	0x173	0x1303	0x273	0xf03	0x373	0x3303	0x473	0x1703	0x573	0x3703	0x673	0x1f03	0x773	0x3f03
0x074	0x737	0x174	0x1337	0x274	0xf37	0x374	0x3337	0x474	0x1737	0x574	0x3737	0x674	0x1f37	0x774	0x3f37
0x075	0x734	0x175	0x1334	0x275	0xf34	0x375	0x3334	0x475	0x1734	0x575	0x3734	0x675	0x1f34	0x775	0x3f34
0x076	0x735	0x176	0x1335	0x276	0xf35	0x376	0x3335	0x476	0x1735	0x576	0x3735	0x676	0x1f35	0x776	0x3f35
0x077	0x43d	0x177	0x103d	0x277	0xc3d	0x377	0x303d	0x477	0x143d	0x577	0x343d	0x677	0x1c3d	0x777	0x3c3d
0x078	0x71f	0x178	0x131f	0x278	0xf1f	0x378	0x331f	0x478	0x171f	0x578	0x371f	0x678	0x1f1f	0x778	0x3f1f
0x079	0x71c	0x179	0x131c	0x279	0xf1c	0x379	0x331c	0x479	0x171c	0x579	0x371c	0x679	0x1f1c	0x779	0x3f1c
0x07a	0x71d	0x17a	0x131d	0x27a	0xf1d	0x37a	0x331d	0x47a	0x171d	0x57a	0x371d	0x67a	0x1f1d	0x77a	0x3f1d
0x07b	0x43f	0x17b	0x103f	0x27b	0xc3f	0x37b	0x303f	0x47b	0x143f	0x57b	0x343f	0x67b	0x1c3f	0x77b	0x3c3f
0x07c	0x73f	0x17c	0x133f	0x27c	0xf3f	0x37c	0x333f	0x47c	0x173f	0x57c	0x373f	0x67c	0x1f3f	0x77c	0x3f3f
0x07d	0x73c	0x17d	0x133c	0x27d	0xf3c	0x37d	0x333c	0x47d	0x173c	0x57d	0x373c	0x67d	0x1f3c	0x77d	0x3f3c
0x07e	0x73d	0x17e	0x133d	0x27e	0xf3d	0x37e	0x333d	0x47e	0x173d	0x57e	0x373d	0x67e	0x1f3d	0x77e	0x3f3d
0x07f	0x43c	0x17f	0x103c	0x27f	0xc3c	0x37f	0x303c	0x47f	0x143c	0x57f	0x343c	0x67f	0x1c3c	0x77f	0x3c3c
0x080	0x547	0x180	0x1147	0x280	0xd47	0x380	0x3147	0x480	0x1547	0x580	0x3547	0x680	0x1d47	0x780	0x3d47
0x081	0x544	0x181	0x1144	0x281	0xd44	0x381	0x3144	0x481	0x1544	0x581	0x3544	0x681	0x1d44	0x781	0x3d44
0x082	0x545	0x182	0x1145	0x282	0xd45	0x382	0x3145	0x482	0x1545	0x582	0x3545	0x682	0x1d45	0x782	0x3d45
0x083	0x541	0x183	0x1141	0x283	0xd41	0x383	0x3141	0x483	0x1541	0x583	0x3541	0x683	0x1d41	0x783	0x3d41
0x084	0x553	0x184	0x1153	0x284	0xd53	0x384	0x3153	0x484	0x1553	0x584	0x3553	0x684	0x1d53	0x784	0x3d53
0x085	0x550	0x185	0x1150	0x285	0xd50	0x385	0x3150	0x485	0x1550	0x585	0x3550	0x685	0x1d50	0x785	0x3d50
0x086	0x551	0x186	0x1151	0x286	0xd51	0x386	0x3151	0x486	0x1551	0x586	0x3551	0x686	0x1d51	0x786	0x3d51
0x087	0x045	0x187	0x105	0x287	0x0c5	0x387	0x305	0x487	0x145	0x587	0x345	0x687	0x1c5	0x787	0x3c5
0x088	0x54f	0x188	0x114f	0x288	0xd4f	0x388	0x314f	0x488	0x154f	0x588	0x354f	0x688	0x1d4f	0x788	0x3d4f
0x089	0x54c	0x189	0x114c	0x289	0xd4c	0x389	0x314c	0x489	0x154c	0x589	0x354c	0x689	0x1d4c	0x789	0x3d4c
0x08a	0x54d	0x18a	0x114d	0x28a	0xd4d	0x38a	0x314d	0x48a	0x154d	0x58a	0x354d	0x68a	0x1d4d	0x78a	0x3d4d
0x08b	0x047	0x18b	0x107	0x28b	0x0c7	0x38b	0x307	0x48b	0x147	0x58b	0x347	0x68b	0x1c7	0x78b	0x3c7
0x08c	0x573	0x18c	0x1173	0x28c	0xd73	0x38c	0x3173	0x48c	0x1573	0x58c	0x3573	0x68c	0x1d73	0x78c	0x3d73
0x08d	0x570	0x18d	0x1170	0x28d	0xd70	0x38d	0x3170	0x48d	0x1570	0x58d	0x3570	0x68d	0x1d70	0x78d	0x3d70
0x08e	0x571	0x18e	0x1171	0x28e	0xd71	0x38e	0x3171	0x48e	0x1571	0x58e	0x3571	0x68e	0x1d71	0x78e	0x3d71
0x08f	0x044	0x18f	0x104	0x28f	0x0c4	0x38f	0x304	0x48f	0x144	0x58f	0x344	0x68f	0x1c4	0x78f	0x3c4
0x090	0x557	0x190	0x1157	0x290	0xd57	0x390	0x3157	0x490	0x1557	0x590	0x3557	0x690	0x1d57	0x790	0x3d57
0x091	0x554	0x191	0x1154	0x291	0xd54	0x391	0x3154	0x491	0x1554	0x591	0x3554	0x691	0x1d54	0x791	0x3d54
0x092	0x555	0x192	0x1155	0x292	0xd55	0x392	0x3155	0x492	0x1555	0x592	0x3555	0x692	0x1d55	0x792	0x3d55
0x093	0x543	0x193	0x1143	0x293	0xd43	0x393	0x3143	0x493	0x1543	0x593	0x3543	0x693	0x1d43	0x793	0x3d43
0x094	0x577	0x194	0x1177	0x294	0xd77	0x394	0x3177	0x494	0x1577	0x594	0x3577	0x694	0x1d77	0x794	0x3d77
0x095	0x574	0x195	0x1174	0x295	0xd74	0x395	0x3174	0x495	0x1574	0x595	0x3574	0x695	0x1d74	0x795	0x3d74
0x096	0x575	0x196	0x1175	0x296	0xd75	0x396	0x3175	0x496	0x1575	0x596	0x3575	0x696	0x1d75	0x796	0x3d75

Table 7 — 11b7S Encoder/Decoder Table (cont'd)

11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x097	0x051	0x197	0x111	0x297	0x0d1	0x397	0x311	0x497	0x151	0x597	0x351	0x697	0x1d1	0x797	0x3d1
0x098	0x55f	0x198	0x115f	0x298	0xd5f	0x398	0x315f	0x498	0x155f	0x598	0x355f	0x698	0x1d5f	0x798	0x3d5f
0x099	0x55c	0x199	0x115c	0x299	0xd5c	0x399	0x315c	0x499	0x155c	0x599	0x355c	0x699	0x1d5c	0x799	0x3d5c
0x09a	0x55d	0x19a	0x115d	0x29a	0xd5d	0x39a	0x315d	0x49a	0x155d	0x59a	0x355d	0x69a	0x1d5d	0x79a	0x3d5d
0x09b	0x053	0x19b	0x113	0x29b	0x0d3	0x39b	0x313	0x49b	0x153	0x59b	0x353	0x69b	0x1d3	0x79b	0x3d3
0x09c	0x57f	0x19c	0x117f	0x29c	0xd7f	0x39c	0x317f	0x49c	0x157f	0x59c	0x357f	0x69c	0x1d7f	0x79c	0x3d7f
0x09d	0x57c	0x19d	0x117c	0x29d	0xd7c	0x39d	0x317c	0x49d	0x157c	0x59d	0x357c	0x69d	0x1d7c	0x79d	0x3d7c
0x09e	0x57d	0x19e	0x117d	0x29e	0xd7d	0x39e	0x317d	0x49e	0x157d	0x59e	0x357d	0x69e	0x1d7d	0x79e	0x3d7d
0x09f	0x050	0x19f	0x110	0x29f	0x0d0	0x39f	0x310	0x49f	0x150	0x59f	0x350	0x69f	0x1d0	0x79f	0x3d0
0x0a0	0x747	0x1a0	0x1347	0x2a0	0xf47	0x3a0	0x3347	0x4a0	0x1747	0x5a0	0x3747	0x6a0	0x1f47	0x7a0	0x3f47
0x0a1	0x744	0x1a1	0x1344	0x2a1	0xf44	0x3a1	0x3344	0x4a1	0x1744	0x5a1	0x3744	0x6a1	0x1f44	0x7a1	0x3f44
0x0a2	0x745	0x1a2	0x1345	0x2a2	0xf45	0x3a2	0x3345	0x4a2	0x1745	0x5a2	0x3745	0x6a2	0x1f45	0x7a2	0x3f45
0x0a3	0x741	0x1a3	0x1341	0x2a3	0xf41	0x3a3	0x3341	0x4a3	0x1741	0x5a3	0x3741	0x6a3	0x1f41	0x7a3	0x3f41
0x0a4	0x753	0x1a4	0x1353	0x2a4	0xf53	0x3a4	0x3353	0x4a4	0x1753	0x5a4	0x3753	0x6a4	0x1f53	0x7a4	0x3f53
0x0a5	0x750	0x1a5	0x1350	0x2a5	0xf50	0x3a5	0x3350	0x4a5	0x1750	0x5a5	0x3750	0x6a5	0x1f50	0x7a5	0x3f50
0x0a6	0x751	0x1a6	0x1351	0x2a6	0xf51	0x3a6	0x3351	0x4a6	0x1751	0x5a6	0x3751	0x6a6	0x1f51	0x7a6	0x3f51
0x0a7	0x04d	0x1a7	0x10d	0x2a7	0x0cd	0x3a7	0x30d	0x4a7	0x14d	0x5a7	0x34d	0x6a7	0x1cd	0x7a7	0x3cd
0x0a8	0x74f	0x1a8	0x134f	0x2a8	0xf4f	0x3a8	0x334f	0x4a8	0x174f	0x5a8	0x374f	0x6a8	0x1f4f	0x7a8	0x3f4f
0x0a9	0x74c	0x1a9	0x134c	0x2a9	0xf4c	0x3a9	0x334c	0x4a9	0x174c	0x5a9	0x374c	0x6a9	0x1f4c	0x7a9	0x3f4c
0x0aa	0x74d	0x1aa	0x134d	0x2aa	0xf4d	0x3aa	0x334d	0x4aa	0x174d	0x5aa	0x374d	0x6aa	0x1f4d	0x7aa	0x3f4d
0x0ab	0x04f	0x1ab	0x10f	0x2ab	0x0cf	0x3ab	0x30f	0x4ab	0x14f	0x5ab	0x34f	0x6ab	0x1cf	0x7ab	0x3cf
0x0ac	0x773	0x1ac	0x1373	0x2ac	0xf73	0x3ac	0x3373	0x4ac	0x1773	0x5ac	0x3773	0x6ac	0x1f73	0x7ac	0x3f73
0x0ad	0x770	0x1ad	0x1370	0x2ad	0xf70	0x3ad	0x3370	0x4ad	0x1770	0x5ad	0x3770	0x6ad	0x1f70	0x7ad	0x3f70
0x0ae	0x771	0x1ae	0x1371	0x2ae	0xf71	0x3ae	0x3371	0x4ae	0x1771	0x5ae	0x3771	0x6ae	0x1f71	0x7ae	0x3f71
0x0af	0x04c	0x1af	0x10c	0x2af	0x0cc	0x3af	0x30c	0x4af	0x14c	0x5af	0x34c	0x6af	0x1cc	0x7af	0x3cc
0x0b0	0x757	0x1b0	0x1357	0x2b0	0xf57	0x3b0	0x3357	0x4b0	0x1757	0x5b0	0x3757	0x6b0	0x1f57	0x7b0	0x3f57
0x0b1	0x754	0x1b1	0x1354	0x2b1	0xf54	0x3b1	0x3354	0x4b1	0x1754	0x5b1	0x3754	0x6b1	0x1f54	0x7b1	0x3f54
0x0b2	0x755	0x1b2	0x1355	0x2b2	0xf55	0x3b2	0x3355	0x4b2	0x1755	0x5b2	0x3755	0x6b2	0x1f55	0x7b2	0x3f55
0x0b3	0x743	0x1b3	0x1343	0x2b3	0xf43	0x3b3	0x3343	0x4b3	0x1743	0x5b3	0x3743	0x6b3	0x1f43	0x7b3	0x3f43
0x0b4	0x777	0x1b4	0x1377	0x2b4	0xf77	0x3b4	0x3377	0x4b4	0x1777	0x5b4	0x3777	0x6b4	0x1f77	0x7b4	0x3f77
0x0b5	0x774	0x1b5	0x1374	0x2b5	0xf74	0x3b5	0x3374	0x4b5	0x1774	0x5b5	0x3774	0x6b5	0x1f74	0x7b5	0x3f74
0x0b6	0x775	0x1b6	0x1375	0x2b6	0xf75	0x3b6	0x3375	0x4b6	0x1775	0x5b6	0x3775	0x6b6	0x1f75	0x7b6	0x3f75
0x0b7	0x071	0x1b7	0x131	0x2b7	0x0f1	0x3b7	0x331	0x4b7	0x171	0x5b7	0x371	0x6b7	0x1f1	0x7b7	0x3f1
0x0b8	0x75f	0x1b8	0x135f	0x2b8	0xf5f	0x3b8	0x335f	0x4b8	0x175f	0x5b8	0x375f	0x6b8	0x1f5f	0x7b8	0x3f5f
0x0b9	0x75c	0x1b9	0x135c	0x2b9	0xf5c	0x3b9	0x335c	0x4b9	0x175c	0x5b9	0x375c	0x6b9	0x1f5c	0x7b9	0x3f5c
0x0ba	0x75d	0x1ba	0x135d	0x2ba	0xf5d	0x3ba	0x335d	0x4ba	0x175d	0x5ba	0x375d	0x6ba	0x1f5d	0x7ba	0x3f5d
0x0bb	0x073	0x1bb	0x133	0x2bb	0x0f3	0x3bb	0x333	0x4bb	0x173	0x5bb	0x373	0x6bb	0x1f3	0x7bb	0x3f3
0x0bc	0x77f	0x1bc	0x137f	0x2bc	0xf7f	0x3bc	0x337f	0x4bc	0x177f	0x5bc	0x377f	0x6bc	0x1f7f	0x7bc	0x3f7f

Table 7 — 11b7S Encoder/Decoder Table (cont'd)

11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x0bd	0x77c	0x1bd	0x137c	0x2bd	0xf7c	0x3bd	0x337c	0x4bd	0x177c	0x5bd	0x377c	0x6bd	0x1f7c	0x7bd	0x3f7c
0x0be	0x77d	0x1be	0x137d	0x2be	0xf7d	0x3be	0x337d	0x4be	0x177d	0x5be	0x377d	0x6be	0x1f7d	0x7be	0x3f7d
0x0bf	0x070	0x1bf	0x130	0x2bf	0x0f0	0x3bf	0x330	0x4bf	0x170	0x5bf	0x370	0x6bf	0x1f0	0x7bf	0x3f0
0x0c0	0x5c7	0x1c0	0x11c7	0x2c0	0xdc7	0x3c0	0x31c7	0x4c0	0x15c7	0x5c0	0x35c7	0x6c0	0x1dc7	0x7c0	0x3dc7
0x0c1	0x5c4	0x1c1	0x11c4	0x2c1	0xdc4	0x3c1	0x31c4	0x4c1	0x15c4	0x5c1	0x35c4	0x6c1	0x1dc4	0x7c1	0x3dc4
0x0c2	0x5c5	0x1c2	0x11c5	0x2c2	0xdc5	0x3c2	0x31c5	0x4c2	0x15c5	0x5c2	0x35c5	0x6c2	0x1dc5	0x7c2	0x3dc5
0x0c3	0x5c1	0x1c3	0x11c1	0x2c3	0xdc1	0x3c3	0x31c1	0x4c3	0x15c1	0x5c3	0x35c1	0x6c3	0x1dc1	0x7c3	0x3dc1
0x0c4	0x5d3	0x1c4	0x11d3	0x2c4	0xdd3	0x3c4	0x31d3	0x4c4	0x15d3	0x5c4	0x35d3	0x6c4	0x1dd3	0x7c4	0x3dd3
0x0c5	0x5d0	0x1c5	0x11d0	0x2c5	0xdd0	0x3c5	0x31d0	0x4c5	0x15d0	0x5c5	0x35d0	0x6c5	0x1dd0	0x7c5	0x3dd0
0x0c6	0x5d1	0x1c6	0x11d1	0x2c6	0xdd1	0x3c6	0x31d1	0x4c6	0x15d1	0x5c6	0x35d1	0x6c6	0x1dd1	0x7c6	0x3dd1
0x0c7	0x055	0x1c7	0x115	0x2c7	0x0d5	0x3c7	0x315	0x4c7	0x155	0x5c7	0x355	0x6c7	0x1d5	0x7c7	0x3d5
0x0c8	0x5cf	0x1c8	0x11cf	0x2c8	0xdcf	0x3c8	0x31cf	0x4c8	0x15cf	0x5c8	0x35cf	0x6c8	0x1dcf	0x7c8	0x3dcf
0x0c9	0x5cc	0x1c9	0x11cc	0x2c9	0xdcc	0x3c9	0x31cc	0x4c9	0x15cc	0x5c9	0x35cc	0x6c9	0x1dcc	0x7c9	0x3dcc
0x0ca	0x5cd	0x1ca	0x11cd	0x2ca	0xdcd	0x3ca	0x31cd	0x4ca	0x15cd	0x5ca	0x35cd	0x6ca	0x1dcd	0x7ca	0x3dcd
0x0cb	0x057	0x1cb	0x117	0x2cb	0x0d7	0x3cb	0x317	0x4cb	0x157	0x5cb	0x357	0x6cb	0x1d7	0x7cb	0x3d7
0x0cc	0x5f3	0x1cc	0x11f3	0x2cc	0xdf3	0x3cc	0x31f3	0x4cc	0x15f3	0x5cc	0x35f3	0x6cc	0x1df3	0x7cc	0x3df3
0x0cd	0x5f0	0x1cd	0x11f0	0x2cd	0xdf0	0x3cd	0x31f0	0x4cd	0x15f0	0x5cd	0x35f0	0x6cd	0x1df0	0x7cd	0x3df0
0x0ce	0x5f1	0x1ce	0x11f1	0x2ce	0xdf1	0x3ce	0x31f1	0x4ce	0x15f1	0x5ce	0x35f1	0x6ce	0x1df1	0x7ce	0x3df1
0x0cf	0x054	0x1cf	0x114	0x2cf	0x0d4	0x3cf	0x314	0x4cf	0x154	0x5cf	0x354	0x6cf	0x1d4	0x7cf	0x3d4
0x0d0	0x5d7	0x1d0	0x11d7	0x2d0	0xdd7	0x3d0	0x31d7	0x4d0	0x15d7	0x5d0	0x35d7	0x6d0	0x1dd7	0x7d0	0x3dd7
0x0d1	0x5d4	0x1d1	0x11d4	0x2d1	0xdd4	0x3d1	0x31d4	0x4d1	0x15d4	0x5d1	0x35d4	0x6d1	0x1dd4	0x7d1	0x3dd4
0x0d2	0x5d5	0x1d2	0x11d5	0x2d2	0xdd5	0x3d2	0x31d5	0x4d2	0x15d5	0x5d2	0x35d5	0x6d2	0x1dd5	0x7d2	0x3dd5
0x0d3	0x5c3	0x1d3	0x11c3	0x2d3	0xdc3	0x3d3	0x31c3	0x4d3	0x15c3	0x5d3	0x35c3	0x6d3	0x1dc3	0x7d3	0x3dc3
0x0d4	0x5f7	0x1d4	0x11f7	0x2d4	0xdf7	0x3d4	0x31f7	0x4d4	0x15f7	0x5d4	0x35f7	0x6d4	0x1df7	0x7d4	0x3df7
0x0d5	0x5f4	0x1d5	0x11f4	0x2d5	0xdf4	0x3d5	0x31f4	0x4d5	0x15f4	0x5d5	0x35f4	0x6d5	0x1df4	0x7d5	0x3df4
0x0d6	0x5f5	0x1d6	0x11f5	0x2d6	0xdf5	0x3d6	0x31f5	0x4d6	0x15f5	0x5d6	0x35f5	0x6d6	0x1df5	0x7d6	0x3df5
0x0d7	0x075	0x1d7	0x135	0x2d7	0x0f5	0x3d7	0x335	0x4d7	0x175	0x5d7	0x375	0x6d7	0x1f5	0x7d7	0x3f5
0x0d8	0x5df	0x1d8	0x11df	0x2d8	0xddf	0x3d8	0x31df	0x4d8	0x15df	0x5d8	0x35df	0x6d8	0x1ddf	0x7d8	0x3ddf
0x0d9	0x5dc	0x1d9	0x11dc	0x2d9	0xddc	0x3d9	0x31dc	0x4d9	0x15dc	0x5d9	0x35dc	0x6d9	0x1ddc	0x7d9	0x3ddc
0x0da	0x5dd	0x1da	0x11dd	0x2da	0xddd	0x3da	0x31dd	0x4da	0x15dd	0x5da	0x35dd	0x6da	0x1ddd	0x7da	0x3ddd
0x0db	0x077	0x1db	0x137	0x2db	0x0f7	0x3db	0x337	0x4db	0x177	0x5db	0x377	0x6db	0x1f7	0x7db	0x3f7
0x0dc	0x5ff	0x1dc	0x11ff	0x2dc	0xdff	0x3dc	0x31ff	0x4dc	0x15ff	0x5dc	0x35ff	0x6dc	0x1dff	0x7dc	0x3dff
0x0dd	0x5fc	0x1dd	0x11fc	0x2dd	0xdfc	0x3dd	0x31fc	0x4dd	0x15fc	0x5dd	0x35fc	0x6dd	0x1dfc	0x7dd	0x3dfc
0x0de	0x5fd	0x1de	0x11fd	0x2de	0xdfd	0x3de	0x31fd	0x4de	0x15fd	0x5de	0x35fd	0x6de	0x1dfd	0x7de	0x3dfd
0x0df	0x074	0x1df	0x134	0x2df	0x0f4	0x3df	0x334	0x4df	0x174	0x5df	0x374	0x6df	0x1f4	0x7df	0x3f4
0x0e0	0x7c7	0x1e0	0x13c7	0x2e0	0xfc7	0x3e0	0x33c7	0x4e0	0x17c7	0x5e0	0x37c7	0x6e0	0x1fc7	0x7e0	0x3fc7
0x0e1	0x7c4	0x1e1	0x13c4	0x2e1	0xfc4	0x3e1	0x33c4	0x4e1	0x17c4	0x5e1	0x37c4	0x6e1	0x1fc4	0x7e1	0x3fc4
0x0e2	0x7c5	0x1e2	0x13c5	0x2e2	0xfc5	0x3e2	0x33c5	0x4e2	0x17c5	0x5e2	0x37c5	0x6e2	0x1fc5	0x7e2	0x3fc5

Table 7 — 11b7S Encoder/Decoder Table (cont'd)

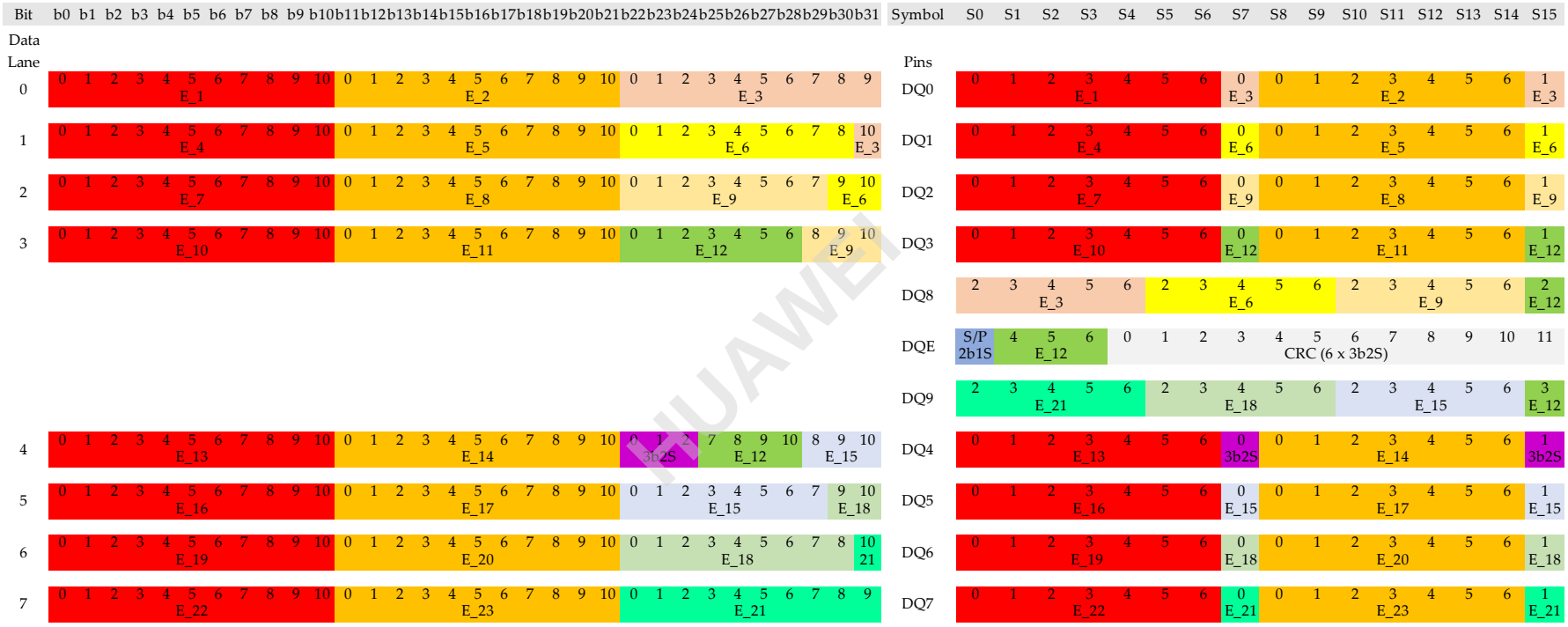
11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S	11b	7S
b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]	b[10:0]	b[13:0]
0x0e3	0x7c1	0x1e3	0x13c1	0x2e3	0xfc1	0x3e3	0x33c1	0x4e3	0x17c1	0x5e3	0x37c1	0x6e3	0x1fc1	0x7e3	0x3fc1
0x0e4	0x7d3	0x1e4	0x13d3	0x2e4	0xfd3	0x3e4	0x33d3	0x4e4	0x17d3	0x5e4	0x37d3	0x6e4	0x1fd3	0x7e4	0x3fd3
0x0e5	0x7d0	0x1e5	0x13d0	0x2e5	0xfd0	0x3e5	0x33d0	0x4e5	0x17d0	0x5e5	0x37d0	0x6e5	0x1fd0	0x7e5	0x3fd0
0x0e6	0x7d1	0x1e6	0x13d1	0x2e6	0xfd1	0x3e6	0x33d1	0x4e6	0x17d1	0x5e6	0x37d1	0x6e6	0x1fd1	0x7e6	0x3fd1
0x0e7	0x05d	0x1e7	0x11d	0x2e7	0x0dd	0x3e7	0x31d	0x4e7	0x15d	0x5e7	0x35d	0x6e7	0x1dd	0x7e7	0x3dd
0x0e8	0x7cf	0x1e8	0x13cf	0x2e8	0xfcf	0x3e8	0x33cf	0x4e8	0x17cf	0x5e8	0x37cf	0x6e8	0x1fcf	0x7e8	0x3fcf
0x0e9	0x7cc	0x1e9	0x13cc	0x2e9	0xfcc	0x3e9	0x33cc	0x4e9	0x17cc	0x5e9	0x37cc	0x6e9	0x1fcc	0x7e9	0x3fcc
0x0ea	0x7cd	0x1ea	0x13cd	0x2ea	0xfcd	0x3ea	0x33cd	0x4ea	0x17cd	0x5ea	0x37cd	0x6ea	0x1fcd	0x7ea	0x3fcd
0x0eb	0x05f	0x1eb	0x11f	0x2eb	0x0df	0x3eb	0x31f	0x4eb	0x15f	0x5eb	0x35f	0x6eb	0x1df	0x7eb	0x3df
0x0ec	0x7f3	0x1ec	0x13f3	0x2ec	0xff3	0x3ec	0x33f3	0x4ec	0x17f3	0x5ec	0x37f3	0x6ec	0x1ff3	0x7ec	0x3ff3
0x0ed	0x7f0	0x1ed	0x13f0	0x2ed	0xff0	0x3ed	0x33f0	0x4ed	0x17f0	0x5ed	0x37f0	0x6ed	0x1ff0	0x7ed	0x3ff0
0x0ee	0x7f1	0x1ee	0x13f1	0x2ee	0xff1	0x3ee	0x33f1	0x4ee	0x17f1	0x5ee	0x37f1	0x6ee	0x1ff1	0x7ee	0x3ff1
0x0ef	0x05c	0x1ef	0x11c	0x2ef	0x0dc	0x3ef	0x31c	0x4ef	0x15c	0x5ef	0x35c	0x6ef	0x1dc	0x7ef	0x3dc
0x0f0	0x7d7	0x1f0	0x13d7	0x2f0	0xfd7	0x3f0	0x33d7	0x4f0	0x17d7	0x5f0	0x37d7	0x6f0	0x1fd7	0x7f0	0x3fd7
0x0f1	0x7d4	0x1f1	0x13d4	0x2f1	0xfd4	0x3f1	0x33d4	0x4f1	0x17d4	0x5f1	0x37d4	0x6f1	0x1fd4	0x7f1	0x3fd4
0x0f2	0x7d5	0x1f2	0x13d5	0x2f2	0xfd5	0x3f2	0x33d5	0x4f2	0x17d5	0x5f2	0x37d5	0x6f2	0x1fd5	0x7f2	0x3fd5
0x0f3	0x7c3	0x1f3	0x13c3	0x2f3	0xfc3	0x3f3	0x33c3	0x4f3	0x17c3	0x5f3	0x37c3	0x6f3	0x1fc3	0x7f3	0x3fc3
0x0f4	0x7f7	0x1f4	0x13f7	0x2f4	0xff7	0x3f4	0x33f7	0x4f4	0x17f7	0x5f4	0x37f7	0x6f4	0x1ff7	0x7f4	0x3ff7
0x0f5	0x7f4	0x1f5	0x13f4	0x2f5	0xff4	0x3f5	0x33f4	0x4f5	0x17f4	0x5f5	0x37f4	0x6f5	0x1ff4	0x7f5	0x3ff4
0x0f6	0x7f5	0x1f6	0x13f5	0x2f6	0xff5	0x3f6	0x33f5	0x4f6	0x17f5	0x5f6	0x37f5	0x6f6	0x1ff5	0x7f6	0x3ff5
0x0f7	0x07d	0x1f7	0x13d	0x2f7	0x0fd	0x3f7	0x33d	0x4f7	0x17d	0x5f7	0x37d	0x6f7	0x1fd	0x7f7	0x3fd
0x0f8	0x7df	0x1f8	0x13df	0x2f8	0xfdf	0x3f8	0x33df	0x4f8	0x17df	0x5f8	0x37df	0x6f8	0x1fdf	0x7f8	0x3fdf
0x0f9	0x7dc	0x1f9	0x13dc	0x2f9	0xfdc	0x3f9	0x33dc	0x4f9	0x17dc	0x5f9	0x37dc	0x6f9	0x1fdc	0x7f9	0x3fdc
0x0fa	0x7dd	0x1fa	0x13dd	0x2fa	0xfdd	0x3fa	0x33dd	0x4fa	0x17dd	0x5fa	0x37dd	0x6fa	0x1fdd	0x7fa	0x3fdd
0x0fb	0x07f	0x1fb	0x13f	0x2fb	0x0ff	0x3fb	0x33f	0x4fb	0x17f	0x5fb	0x37f	0x6fb	0x1ff	0x7fb	0x3ff
0x0fc	0x7ff	0x1fc	0x13ff	0x2fc	0xfff	0x3fc	0x33ff	0x4fc	0x17ff	0x5fc	0x37ff	0x6fc	0x1fff	0x7fc	0x3fff
0x0fd	0x7fc	0x1fd	0x13fc	0x2fd	0xffc	0x3fd	0x33fc	0x4fd	0x17fc	0x5fd	0x37fc	0x6fd	0x1ffc	0x7fd	0x3ffc
0x0fe	0x7fd	0x1fe	0x13fd	0x2fe	0xffd	0x3fe	0x33fd	0x4fe	0x17fd	0x5fe	0x37fd	0x6fe	0x1ffd	0x7fe	0x3ffd
0x0ff	0x07c	0x1ff	0x13c	0x2ff	0x0fc	0x3ff	0x33c	0x4ff	0x17c	0x5ff	0x37c	0x6ff	0x1fc	0x7ff	0x3fc

NOTE 1 The values in columns are encoded in [MSB:LSB].

NOTE 2 The 7S columns show the internal binary representation of the PAM3 symbols that translate in signal levels as described in *TABLE 4*.

2.9 PAM3 (cont'd)

The total burst transfer payload per channel is encoded using 23 x 11b7S and 1 x 3b2S for the data, 6 x 3b2S for the CRC and 1 x 2b1S for the SEV/PSN, it adds up to the 176 PAM3 symbols that can be allocated for a 16 burst over 11 data lines. The logical to physical lane mapping, burst order and bit-to-symbol encoder grouping is shown in *FIGURE 7*.



NOTE 1 The 11b7S encoders/decoders are labelled as E_*, indexed from 1 to 23. The data remainder is labelled as 3bS, same as the CRC. The Severity and Poison is labelled as 2b1S.

NOTE 2 The bit and symbol indexes indicate input/output order (LSB first, MSB last).

NOTE 3 For details about the DQE burst CRC bit order please check the *READ AND WRITE CRC* section.

Figure 7 — PAM3 Burst Data Mapping

2.9.3 Recommended Implementation of the 11b7S Encoder / Decoder

The implementation of the 11b7S encoder/decoder can be efficiently achieved by a combination of 5b3S and 7b4S encoders/decoders and the check bit truth table shown in *TABLE 8*, *TABLE 9*, and *TABLE 10*, respectively. The resulting 11b7S truth table is shown in *TABLE 7*. The RTL code implementation of the encoder / decoder is shown in :

- **Encoding**: The encoder first checks if the first 5 data bits can be mapped to one of 26 sets of 3 PAM3 symbols. The 12 bits of intermediate data are generated according to the result (check bit) and *TABLE 9* (b_p table). Finally, the first 5 bits and the remaining 7 bits of intermediate data are encoded by *TABLE 8* (5b3S table) and *TABLE 10* (7b4S table), respectively, to produce 7 PAM3 symbols.
- **Decoding**: The 5b3S and 7b4S decoders are used to decode the first 3 and the remaining 4 PAM3 symbols to produce 12 bits of intermediate data. 11 bits of original data is recovered according to the check bit (obtained directly from 7b4S decoding result) and *TABLE 9* (b_p table).

```
module pam3_enc(input [10:0] i_din, output [13:0] o_encoded);
        wire [10:0] b_i;
                                                wire [11:0] b;
        wire [5:0] t0_5;
        wire [7:0] t6_13;
        wire chk;

        assign b_i = i_din;
        assign chk = (b_i[0] & b_i[1] & b_i[3]) | (b_i[0] & b_i[1] & b_i[2]);
        assign b = chk ? {b_i[10:7], 3'b001, b_i[6], b_i[5], b_i[4], ~b_i[3], b_i[2] & b_i[3]} : {b_i[10:5], 1'b0,
                    b_i[4:0]};
        assign t0_5 = {(b[2] & b[4]) | (b[2] & b[3]), ((~b[0]) & b[4]) | ((~b[1]) & b[4]) | b[2], ((~b[2]) & b[3]) |
                    (b[3] & b[4]), ((~b[0]) & (~b[2])) | ((~b[1]) & (~b[2])) | (b[2] & b[4]), (b[0] & b[1] & b[4]) |
                    ((~b[0]) & (~b[1])), (~b[0]) | b[1]};
        assign t6_13 = {((~b[5]) & b[9] & b[11]) | ((~b[8]) & b[9] & b[11]) | ((~b[5]) & b[9] & b[10]) | ((~b[8]) &
                    b[9] & b[10]), ((~b[5]) & b[11]) | ((~b[8]) & b[11]) | ((~b[5]) & b[9]) | ((~b[8]) & b[9]),
                    ((~b[5]) & (~b[9]) & b[10]) | ((~b[8]) & (~b[9]) & b[10]) | ((~b[5]) & b[10] & b[11]) | ((~b[8])
                    & b[10] & b[11]),((~b[5]) & (~b[9])) | ((~b[8]) & (~b[9])) | ((~b[5]) & b[11]) | ((~b[8]) & b[11]),
                    (b[5] & b[8] & b[9] & b[11]) | (b[5] & b[8] & b[9] & b[10]) | (b[6] & b[8]) | (b[6] & b[7]),
                    ((~b[5]) & b[8]) | (b[8] & b[11]) | (b[8] & b[9]) | b[6], (b[5] & b[8] & (~b[9]) & b[10]) | (b[5]
                    & b[8] & b[10] & b[11]) | ((~b[6]) & b[7]) | (b[7] & b[8]), ((~b[5]) & (~b[6])) | (b[8] & (~b[9]))
                    | (b[8] & b[11]) | (b[6] & b[8])};
        assign o_encoded = {t6_13, t0_5};
endmodule

module pam3_dec(input [13:0] i_din, output [10:0] o_decoded);
        wire [13:0] t;
        wire [11:0] b;
        wire chk;
```

Figure 8 — RTL Code Implementation of the 11b7S Encoder/Decoder

2.9.3 Recommended Implementation of the 11b7S Encoder / Decoder (cont'd)

Table 8 — 5b3S Encoder/Decoder Table

b[0:4]	t[0:1]	t[2:3]	t[4:5]	chk	b[0:4]	t[0:1]	t[2:3]	t[4:5]	chk
00000	+1	0	-1	0	10000	0	0	-1	0
00001	+1	0	0	0	10001	-1	0	0	0
00010	+1	+1	-1	0	10010	-1	+1	-1	0
00011	+1	+1	0	0	10011	-1	+1	0	0
00100	+1	-1	0	0	10100	-1	-1	0	0
00101	+1	0	+1	0	10101	-1	0	+1	0
00110	+1	-1	+1	0	10110	-1	-1	+1	0
00111	+1	+1	+1	0	10111	-1	+1	+1	0
01000	0	0	-1	0	11000	0	-1	-1	0
01001	0	0	0	0	11001	+1	-1	-1	0
01010	0	+1	-1	0	11010	X	X	X	+1
01011	0	+1	0	0	11011	X	X	X	+1
01100	0	-1	0	0	11100	X	X	X	+1
01101	0	0	+1	0	11101	X	X	X	+1
01110	0	-1	+1	0	11110	X	X	X	+1
01111	0	+1	+1	0	11111	X	X	X	+1

Table 9 — Check Bit Truth Table (b_p)

b[0:4]	b_p[0]	b_p[1]	b_p[2]
11010	0	0	0
11011	0	0	1
11100	0	1	0
11101	0	1	1
11110	1	0	0
11111	1	0	1

2.9.3 Recommended Implementation of the 11b7S Encoder / Decoder (cont'd)

Table 10 — 7b4S Encoding/Decoding Table

b[5:11]	t[6:7]	t[8:9]	t[10:11]	t[12:13]	b[5:11]	t[6:7]	t[8:9]	t[10:11]	t[12:13]	b[5:11]	t[6:7]	t[8:9]	t[10:11]	t[12:13]
0000000	0	-1	0	-1	0100000	-1	0	0	-1	1000000	-1	-1	0	-1
0000001	0	-1	0	0	0100001	-1	0	0	0	1000001	-1	-1	0	0
0000010	0	-1	+1	-1	0100010	-1	0	+1	-1	1000010	-1	-1	+1	-1
0000011	0	-1	+1	0	0100011	-1	0	+1	0	1000011	-1	-1	+1	0
0000100	0	-1	-1	0	0100100	-1	0	-1	0	1000100	-1	-1	-1	0
0000101	0	-1	0	+1	0100101	-1	0	0	+1	1000101	-1	-1	0	+1
0000110	0	-1	-1	+1	0100110	-1	0	-1	+1	1000110	-1	-1	-1	+1
0000111	0	-1	+1	+1	0100111	-1	0	+1	+1	1000111	-1	-1	+1	+1
0001000	0	0	0	-1	0101000	0	+1	0	-1	1001000	0	-1	-1	-1
0001001	0	0	0	0	0101001	0	+1	0	0	1001001	0	0	-1	-1
0001010	0	0	+1	-1	0101010	0	+1	+1	-1	1001010	+1	-1	-1	-1
0001011	0	0	+1	0	0101011	0	+1	+1	0	1001011	+1	0	-1	-1
0001100	0	0	-1	0	0101100	0	+1	-1	0	1001100	-1	0	-1	-1
0001101	0	0	0	+1	0101101	0	+1	0	+1	1001101	0	+1	-1	-1
0001110	0	0	-1	+1	0101110	0	+1	-1	+1	1001110	-1	+1	-1	-1
0001111	0	0	+1	+1	0101111	0	+1	+1	+1	1001111	+1	+1	-1	-1
0010000	+1	-1	0	-1	0110000	-1	+1	0	-1	1010000	X	X	X	X
0010001	+1	-1	0	0	0110001	-1	+1	0	0	1010001	X	X	X	X
0010010	+1	-1	+1	-1	0110010	-1	+1	+1	-1	1010010	X	X	X	X
0010011	+1	-1	+1	0	0110011	-1	+1	+1	0	1010011	X	X	X	X
0010100	+1	-1	-1	0	0110100	-1	+1	-1	0	1010100	X	X	X	X
0010101	+1	-1	0	+1	0110101	-1	+1	0	+1	1010101	X	X	X	X
0010110	+1	-1	-1	+1	0110110	-1	+1	-1	+1	1010110	X	X	X	X
0010111	+1	-1	+1	+1	0110111	-1	+1	+1	+1	1010111	X	X	X	X
0011000	+1	0	0	-1	0111000	+1	+1	0	-1	1011000	X	X	X	X
0011001	+1	0	0	0	0111001	+1	+1	0	0	1011001	X	X	X	X
0011010	+1	0	+1	-1	0111010	+1	+1	+1	-1	1011010	X	X	X	X
0011011	+1	0	+1	0	0111011	+1	+1	+1	0	1011011	X	X	X	X
0011100	+1	0	-1	0	0111100	+1	+1	-1	0	1011100	X	X	X	X
0011101	+1	0	0	+1	0111101	+1	+1	0	+1	1011101	X	X	X	X
0011110	+1	0	-1	+1	0111110	+1	+1	-1	+1	…	…	…	…	…
0011111	+1	0	+1	+1	0111111	+1	+1	+1	+1	1111111	X	X	X	X

2.10 Static Data Scrambler

In PAM3 and NRZ modes, GDDR7 SGRAMs feature a programmable static data scrambler in their internal read and write data paths that is intended to help overcome potential data dependent signaling issues on system level by mapping the payload data to different codes. The feature is intended to help improve system margin.

The feature scrambles memory array read data before they are transmitted, and de-scrambles write data before they are written to the memory array. Refer to *FIGURE 5* and *FIGURE 6* for the location of the data scramblers in the read and write data paths, respectively.

The data scrambler performs a bitwise XOR operation on the 256 data bits per read or write burst with a user programmed static scramble code. The same XOR operation with identical scramble code shall be performed on host side, such that the data scramble on the transmitting side (host or DRAM) is reversed on the receiving side (DRAM or host). The memory array always stores non-scrambled data.

For the data scramble operation itself the 256-bit data per read or write burst are organized as 8 DQ x 32 UI (BL32), exactly as they are transmitted in NRZ mode. *FIGURE 9* illustrates the correspondence of the scramble code to these data; the logic is the same for reads and writes. The 32-bit scramble code is programmed via MR20 and applied to all 8 DQs, with the following options depending on implementation choice per DRAM vendor:

- **32-bit code**: four MRS commands convey 8 code bits each on OP[7:0] while bits OP[11:9] select the position (byte address) within the 32-bit code. The same 8-bit code for all 4 bytes can be programmed for devices that support a 32-bit code by setting the byte address to 111_B in OP[11:9], allowing the host to program the scramble code with a single MRS as opposed to four MRS commands.
- **8-bit code**: a single MRS command conveys 8 code bits on OP[7:0]; these 8 bits are internally replicated to a 32-bit code, and the byte address bits in OP[11:9] are ignored in this case.

Vendor ID3 (IRA 2) bit DQ3 identifies whether the 8-bit or 32-bit scramble code is supported by a device. The scramble code is set to 0_B by default, resulting in no data scramble.

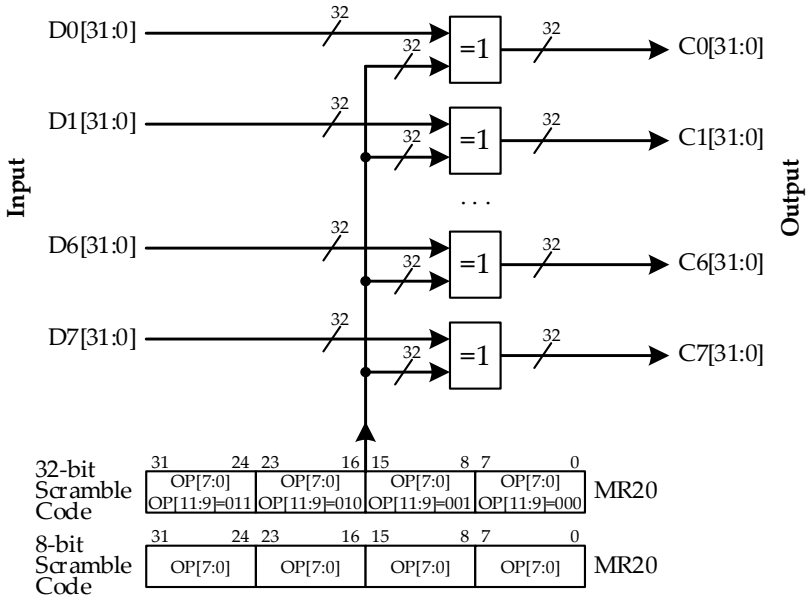


Figure 9 — Code Mapping of the Static Data Scrambler

2.11 ERR Signal

GDDR7 DRAMs include an error signal (ERR) per channel that is an output only signal used to convey either Write CRC errors (WRCRC) or Command Address Parity errors (CAPAR) to the host.

ERR is a PAM3 output in both PAM3 and NRZ mode. ERR output is encoded as shown in *TABLE 11* in both PAM3 mode and NRZ mode when SEV2ERR is disabled. When SEV2ERR is enabled the ERR is encoded as in *Table 12*. ERR pulse width is 1 nCK4 cycle and includes an analogue delay, tWCK2ERRO as shown in *Figure 10*.

Table 11 — ERR PAM3 Encoding (PAM3 and NRZ Mode)

Trit	Binary	Error Type	Mode Register	ERR Latency
+1	11	No Error	Either CAPAR or WRCRC or both	Either CAPAR2ERR or WRCRC2ERR or both
0	01	WRCRC Error	WRCRC (MR0 OP4)	WRCRC2ERR (MR2 OP[11:7])
-1	00	CAPAR Error	CAPAR (MR15 OP0)	CAPAR2ERR (MR15 OP[11:8])

If enabled, WRCRC errors are checked on the WR burst, and the outcome is sent to the host after the programmed WRCRC2ERR according to the following formula.

WRCRC2ERR * tCK4 + tWCK2ERRO

If enabled, CA parity is checked every CK4 cycle, and the outcome is sent to the host via the ERR signal after the programmed CAPAR2ERR. GDDR7 devices may support a fixed latency, a variable latency or both for CAPAR2ERR. Vendor datasheets should be consulted to see what modes are supported.

For fixed CAPAR2ERR latencies (MR15 OP[11:8] = 0001_B to 1111_B), the outcome is sent to the host via the ERR after the programmed value according to the following formula.

CAPAR2ERR * tCK4 + tWCK2ERRO

In the case of variable CAPAR2ERR latency (MR15 OP[11:8] = 0), the latency range can be calculated per the following formula.

RU { tWCK2ERRINT / tCK4 } + tWCK2ERRO

See the Latency Timings table (*TABLE 144*) in the *AC TIMINGS* section for the min/max for CAPAR2ERR (either or both modes) and WRCRC2ERR at the frequency of operation.

If both CAPAR and WRCRC error types need to be signaled on the same cycle, only the CAPAR error will be signaled to the host as shown in *Figure 134.*

2.11 ERR Signal (cont'd)

GDDR7 SGRAMs support an optional feature, SEV2ERR mode, whereby the Severity (SEV) info that is normally sent to the host on the DQE signal on reads is transmitted on the ERR signal. The SEV2ERR mode is only supported in NRZ mode when WRCRC, RDCRC, and Poison are disabled, and Severity is on. The SEV2ERR mode is a power savings feature that allows the host to also disable the DQE (High-Z) in NRZ mode and still retain Severity. The Severity is encoded in place of the WRCRC as shown in *TABLE 12*. The Severity is sent to the host after tSEV2ERR from the Read command. SEV2ERR mode is enabled with MR5 OP9 = 1_B. DQE is disabled (High-Z) with MR5 OP10 = 0_B (DQE_HZ). See *FIGURE 101* in the READ section for a NRZ Read with both SEV2ERR mode and DQE_HZ enabled.

Table 12 — ERR PAM3 Encoding (Optional SEV2ERR Mode)

Trit	Binary	Error Type	Mode Register	ERR Latency
+1	11	No Error	CAPAR or SEV2ERR or both	CAPAR2ERR or tSEV2ERR or both
0	01	Severity (SEV)	SEV2ERR (MR5 OP9)	tSEV2ERR
-1	00	CAPAR Error	CAPAR (MR15 OP0)	CAPAR2ERR (MR15 OP[11:8])

The host can determine if the DRAM supports SEV2ERR mode by using the Info Read to check the SEV2ERR field.

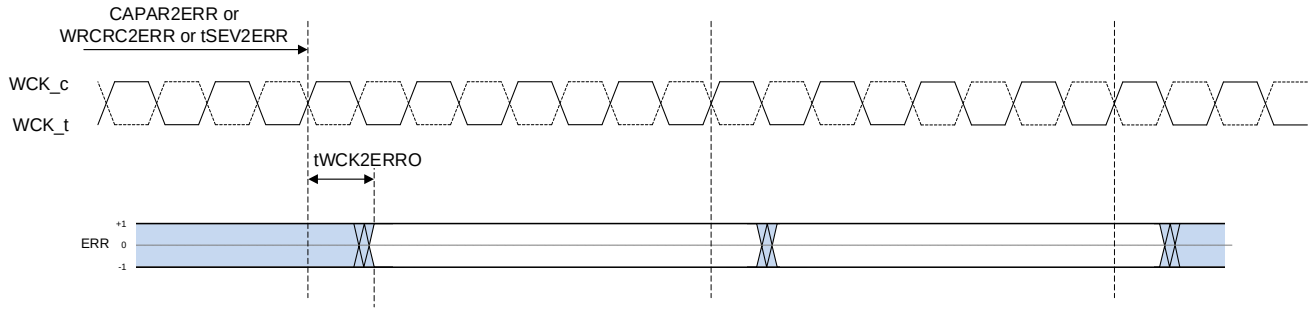
TABLE 13 summarizes the state of the ERR signal when certain features that use the ERR signal are enabled or disabled. If CAPAR, WRCRC, SEV2ERR and CSP feedback are disabled, then the state of the ERR signal is High-Z. The ERR signal is High-Z, tMOD after the last feature that requires ERR is disabled. The ERR signal is High-Z in Sleep modes after tCPDED expires.

2.11 ERR Signal (cont'd)

Table 13 — Features using ERR and the ERR Signal State

ERR state	Setting	ERR State (Sync)	ERR State (Async)
CAPAR MR15 OP0	Enabled	No Error: +1 WRCRC error: 0 CAPAR error: -1	High-Z
	Disabled	+1, but High-Z if all disabled	
WRCRC MR0 OP4	Enabled	No Error: +1 WRCRC error: 0 CAPAR error: -1	
	Disabled	+1, but High-Z if all disabled	
SEV2ERR MR5 OP9	Enabled	No Error: +1 SEV error: 0 CAPAR error: -1	
	Disabled	+1, but High-Z if all disabled	
CSP_FB (opt.) MR15 OP2	Enabled	CSP/No Error: +1 No CSP: 0 CAPAR error: -1	
	Disabled	+1, but High-Z if all disabled	

See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details on the ERR behavior when CAPAR is enabled. See the *WRITE* section for more details on the ERR behavior when WRCRC is enabled.



NOTES:

1. CAPAR2ERR is the CA Parity Error Latency programmed in Mode Register MR15.
2. WRCRC2ERR is the Write CRC Error Latency programmed in Mode Register MR2.
3. tSEV2ERR is the Severity Latency during SEV2ERR mode.

Figure 10 — ERR Lane Timing

2.12 Simplified State Diagram

The state diagram provides a simplified illustration of the allowed state transitions and the related commands to control them. The following operations are not or not completely shown in the diagram:

- state transitions involving more than one bank.
- device configurations set at the exit from reset state, including 4 channel mode vs. 2 channel mode.
- enabling or disabling of on-die termination.
- the procedures for training, WCK frequency change and dynamic voltage switching (DVS).

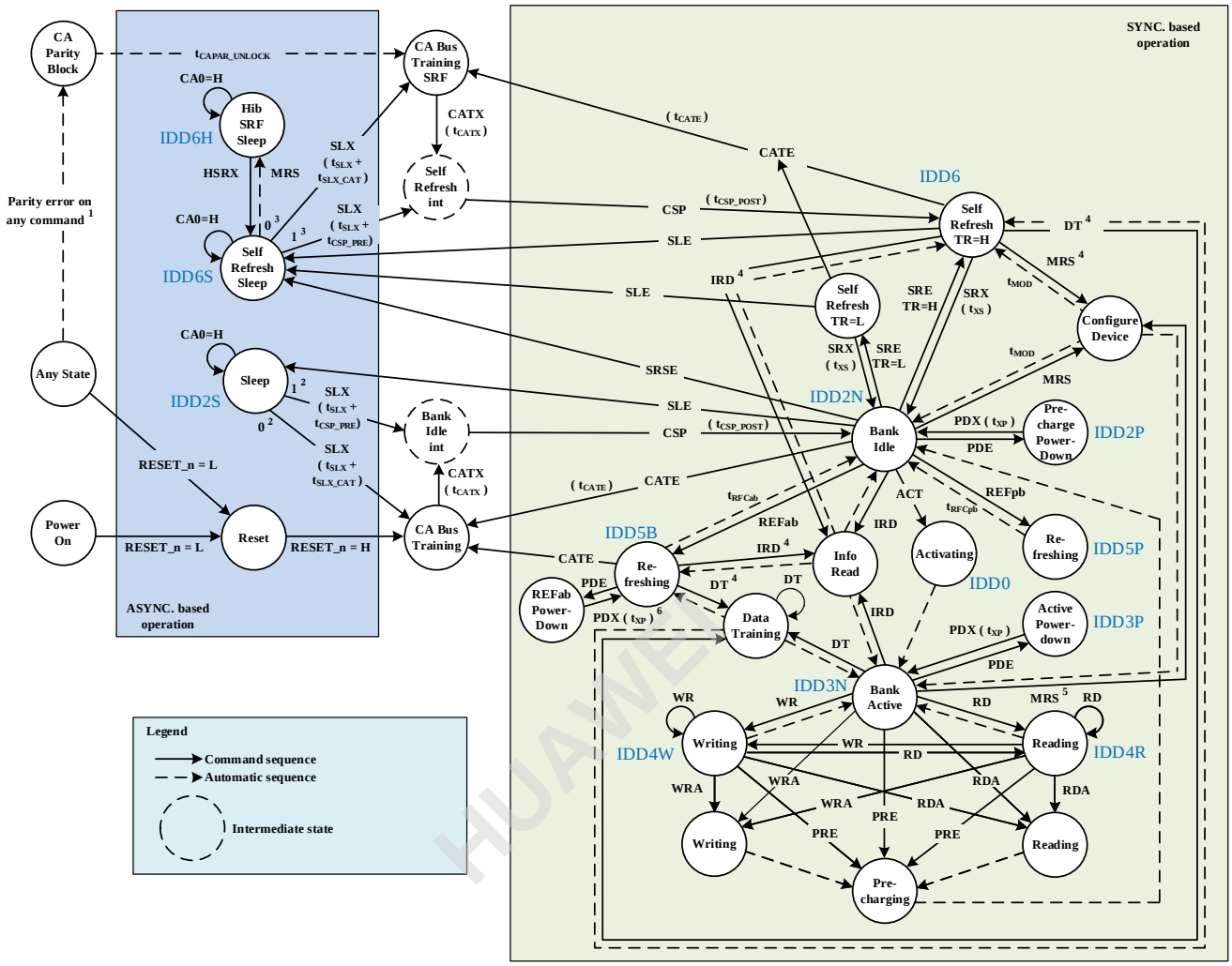
For a complete description of the device behavior, use the information provided in the state diagram along with the *COMMAND TRUTH TABLE* as well as the *IDD* and *AC TIMINGS* specifications.

FIGURE 11 uses the code names for the commands listed in *TABLE 14*.

Table 14 — State Diagram Command Codes

Code	Command	Code	Command
ACT	Activate	MRS	Mode Register Set
PRE	PREab or PREpb	IRD	Info Read
RD	Read	PDE	Power-down Entry
RDA	Read w/ AP	PDX	Power-Down Exit
WR	Write	SRE	Self Refresh Entry
WRA	Write w/ AP	SRX	Self Refresh Exit
DT	LDFF, WRTR or RDTR	SLE	Sleep Entry
CATE	CA Training Entry	SRSE	Self Refresh Sleep Entry
CATX	CA Training Exit	SLX	Sleep or Self Refresh Sleep Exit
REF	REFpb; RFMab or RFMpb	HSRX	Hibernate Self Refresh Sleep Exit
REFab	REFab	CSP	Command Start Point

2.12 Simplified State Diagram (cont'd)



NOTES:

1. Commands are blocked if MR15 OP0 CAPAR on = 1_B and MR15 OP1 CAPAR Command Blocking (CAPARBLK) = 1_B.
2. MR0 OP10 SLX2CAT Mode Register. If the register is set to 0_B before entering Sleep, after the SLX command and tSLX + tSLX_CAT has been met, the DRAM will enter CA Bus Training. If the register is set to 1_B before entering Sleep, the DRAM will await the CSP command after the SLX command and tSLX + tCSP_PRE has been met.
3. MR0 OP11 SRSLX2CAT Mode Register. If the register is set to 0_B before entering Self Refresh Sleep, after the SLX command and tSLX + tSLX_CAT has been met, the DRAM will enter CA Bus Training Self Refresh. If the register is set to 1_B before entering Self Refresh Sleep, the DRAM will await the CSP command after the SLX command and tSLX + tCSP_PRE has been met.
4. Data Training and Info Read (IRD) commands are only allowed when the training flag (TR) is set in the REFab command. Data Training, Info Read and MRS commands are only allowed with the training flag (TR) is set in the SRE command. See the *REFRESH* and *SELF REFRESH* sections for details.
5. A MRS command is legal in Bank Active to only certain registers. See the *MODE REGISTERS* section for details.
6. On receiving a PDX command, the DRAM exits to Refreshing state if tRFCab has not expired and bank idle state if tRFCab has expired.

Figure 11 — Simplified State Diagram

3 Initialization and Power-Off

GDDR7 SGRAMs must be powered up and initialized in a predefined manner. Operational procedures other than those specified may result in undefined operation. In general Mode Registers do not have reset default values except for some selected bits. Please refer to Mode Register section for the Default Mode Register Setting. If the mode registers are not set during the initialization sequence, it may lead to unspecified operation.

3.1 Power-Up Initialization

Following RESET during power up initialization or initialization with stable power, CA training mode is entered automatically. Please refer to Command Address Training section and State Diagram section for the detail. The WCK frequency of this automatic CA training is within fWCKNRZmax. And after this automatic CA training exit, the device is in NRZ mode by the Default Mode Register Setting. Please note that the CA inputs will only be guaranteed after the complete CA trainings have been executed. A Command Start Point command is required to be issued exactly once in the following state transitions to normal operation. Please see *COMMAND START POINT (CSP)* section for the detail.

Please note that the ECS Error Log registers associated with UEAL and CEAL and the UECL UE count register have no defined default upon power up or device reset. Please refer to *ERROR CHECK AND SCRUB (AUTO ECS)* section.

Power-up Initialization Sequence

1) Apply power to VPP. Apply power to VDD at the same time or after power is applied to VPP. Apply power to VDDQ at the same time or after power is applied to VDD. VPP must be higher than VDD and VDDQ at all times the device is powered up. For details on voltage requirements for VDD, VDDQ and VPP during power supply ramp up and down see *FIGURE 14*. During power supply ramp time tINIT0, RESET_n and all other input signals may be in an undefined state (driven LOW or HIGH, or High-Z) and RESET_n is recommended to set to LOW (≤0.2 x VDD).
2) The voltage levels on all signal balls must be less than or equal to VDD and VDDQ on one side and must be larger than or equal to VSS on the other side.
3) Assert RESET_n LOW. Maintain RESET_n LOW for a minimum time of tINIT1. After tINIT4 time has elapsed, the ERR and RCK_t/c are High-Z, and all DQ and DQE ODT are off.
4) Set system configuration info at least a time tATS before RESET_n is driven HIGH:
    a. Drive CA0_A, CA0_ B, CA0_C and CA0_D HIGH for 4 CH mode
    or CA0_A and CA0_C LOW for 2 CH mode.
    In 2CH mode, CA0_B and CA0_D are don't care in this configuration selection and all CA, WCK_t and WCK_c input of channels B and D may be left floating.
    b. For each active channel, drive CA2 per *TABLE 16* to select WCK Termination.
    c. For each active channel, drive CA1 per *TABLE 17* to select CA bus termination of each channel.
    d. For each active channel, drive CA[4:3] high.
    e. For each active channel, drive WCK_t and WCK_c to static LOW and HIGH levels, respectively.
    f. After meeting tATS requirement, drive RESET_n HIGH

3.1 Power-Up Initialization (cont'd)

5) After RESET_n is pulled HIGH, maintain CA[4:0] for a minimum time of that.

6) Drive CA[4:0] HIGH after that is satisfied. Assert and hold RNOP2 and CNOP2 commands. The device performs the initial impedance calibration during this time; this will be done without external clocks. Latest after tINIT2 the device enables the WCK and CA ODT as determined in steps 4 and 5.

7) Provide a stable WCK clock for a minimum of tINIT3 cycles. The maximum allowed WCK frequency during device initialization is fWCKNRZ(max), the WCK frequency can be changed after initialization to any other value within the allowed fWCK range, following the procedure described in the *FREQUENCY CHANGE SEQUENCE (FWCK)* section. Maintain a stable WCK clock indefinitely. The WCK clock may only be stopped when the device is put into self refresh sleep or sleep (see Sleep for timing requirements on stopping and enabling the WCK clock). The device state will be undefined when the WCK clock is stopped outside of self refresh sleep or sleep, and a chip reset as outlined in the Initialization with Stable Power section would be required before the GDDR7 SGRAM can return to normal operation.

8) After tINIT3 time, the device automatically enters CA training mode.

9) Complete CA training (optional).

10) After CA Training exit by CATX command, assert and hold RNOP2 and CNOP2 command for a minimum time of tCSP_PRE.

11) Issue CSP command and hold RNOP2 and CNOP2 commands until for a minimum time of tCSP_POST.

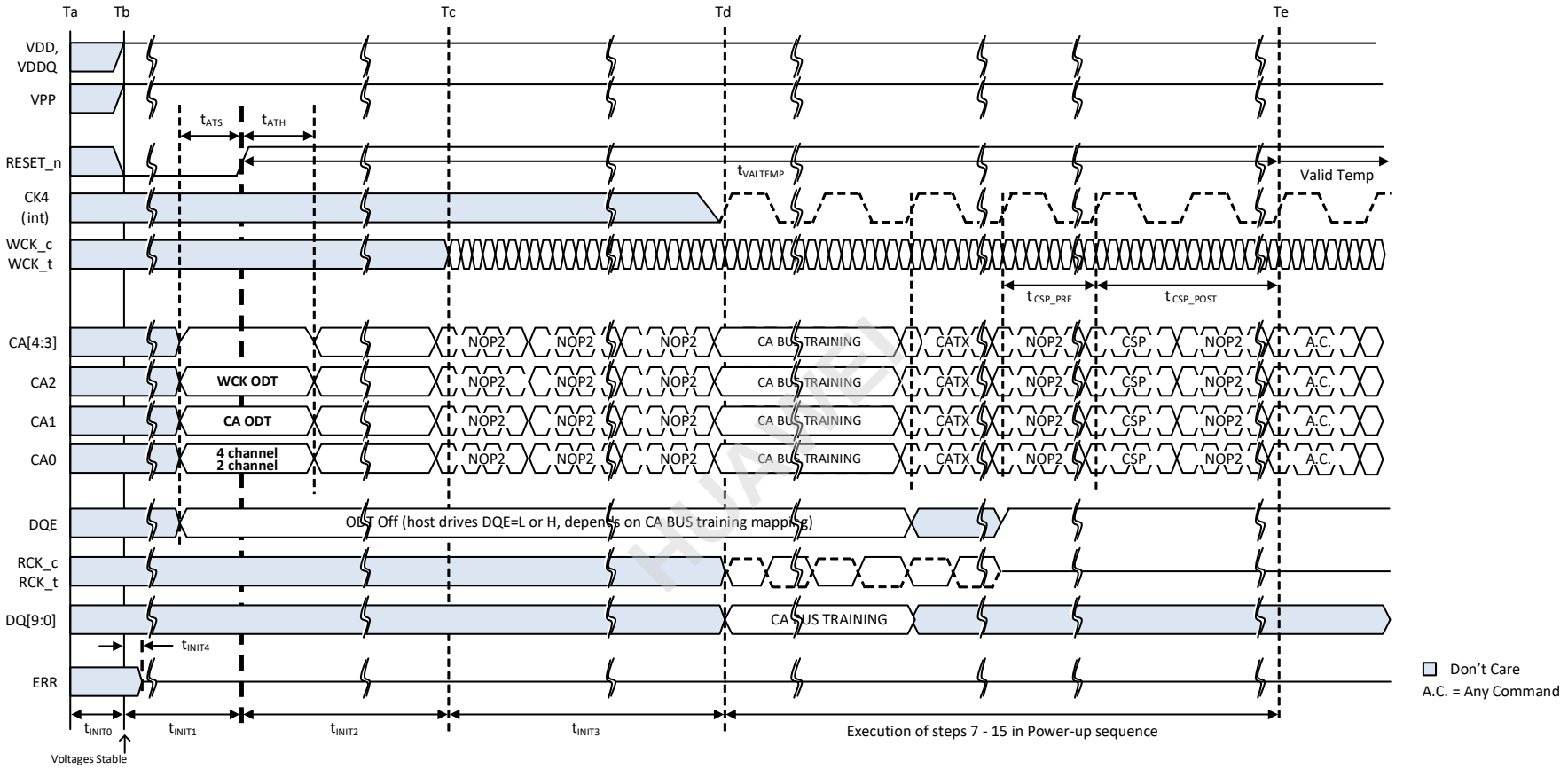
12) Issue MR11 to map Logical Signal of DQ[9:0]. (optional) The Logical Signals to Physical Pin Mapping for DQ[9:0] are also specified in Signals of Package Specification section.

13) Issue Info Read commands to read the Vendor ID. (optional). As there is no default RL during initialization, the default latency for the Info Read command is vendor specific and will be between 10 and 18 nCK4. See vendor datasheets for the default value or the host can use back-to-back IRD commands to capture the IRD output, as described in the Info Read section.

14) Issue MRS commands to the mode registers in any order with appropriate values. All these features must be programmed before Read and Write training, and tMOD must be met during this procedure. Note that by default, the settings for the Mode Registers 10 to 15, also known as Frequency Dependent Mode Registers, are postponed until the next Sleep or Self Refresh Sleep entry/exit sequence, unless the FD_FLAG is set (MR0 OP11 = 1_B). In that case the registers are updated after tMOD or tMOD15. This is particularly important for the case that the device requires the host to program the WCK Frequency (MR12 OP[8:4]), as MR12 can only be updated with a Sleep or Self Refresh Sleep entry/exit sequence. Failure to do so may lead to operation not guaranteed as the default value of WCK Frequency register is meant for power up and reading of vendor ID only.

15) After any necessary training sequences such as READ training (LDFF, RDTR) and WRITE training (WRTR, RDTR, RDTREC), the device is ready for normal operation.

3.1 Power-Up Initialization (cont'd)



NOTE 1 CK4 (int) is sync. for illustration. (may not be sync. before CSP Command)

NOTE 2 RCKMODE is disabled by Mode Register Default Settings.

NOTE 3 (*) Channel A and Channel C only in 2-channel mode and all Channel in 4-channel mode.

NOTE 4 Any commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

NOTE 5 Ta is the point when any supply first reaches 0.3V. Tb is the point when all supply voltages are within their defined ranges.

Figure 12 — Power-up Initialization

3.1 Power-Up Initialization (cont'd)

Table 15 — Device Initialization Timings

Parameter		Symbol	Value		Unit	Notes
			Min	Max		
Power supply ramp time		tINIT0	0.01	200	ms	
RESET_n initial LOW time after power-up		tINIT1	200	-	us	
Time after RESET_n de-assertion before starting the WCK clock		tINIT2	4	-	ms	
Stable CK4 (int.) clock cycles before CA Bus Training		tINIT3	100	-	nCK4	
Time from RESET_n assertion to ERR and RCK_t/_c are High-Z, and all DQ and DQE ODT are off		tINIT4	-	100	ns	
CA[4:0] setup time before RESET_n de-assertion		tATS	10	-	ns	
CA[4:0] hold time after RESET_n de-assertion		tATH	10	-	ns	
RESET_n LOW time with stable power		tRES	100	-	ns	
Temperature sensing time after RESET_n deassertion		tVALTEMP	4		ms	
CSP command preamble delay		tCSP_PRE			ns	
CSP command to valid command delay	CAPARBLK disabled	tCSP_POST			ns	
	CAPARBLK enabled				ns	

Table 16 — WCK Termination

CA2	Default WCK Termination Value
LOW	Disabled
HIGH	48 Ohm

Table 17 — CA Termination

CA1	Default CA Termination Value
LOW	Disabled
HIGH	48 Ohm

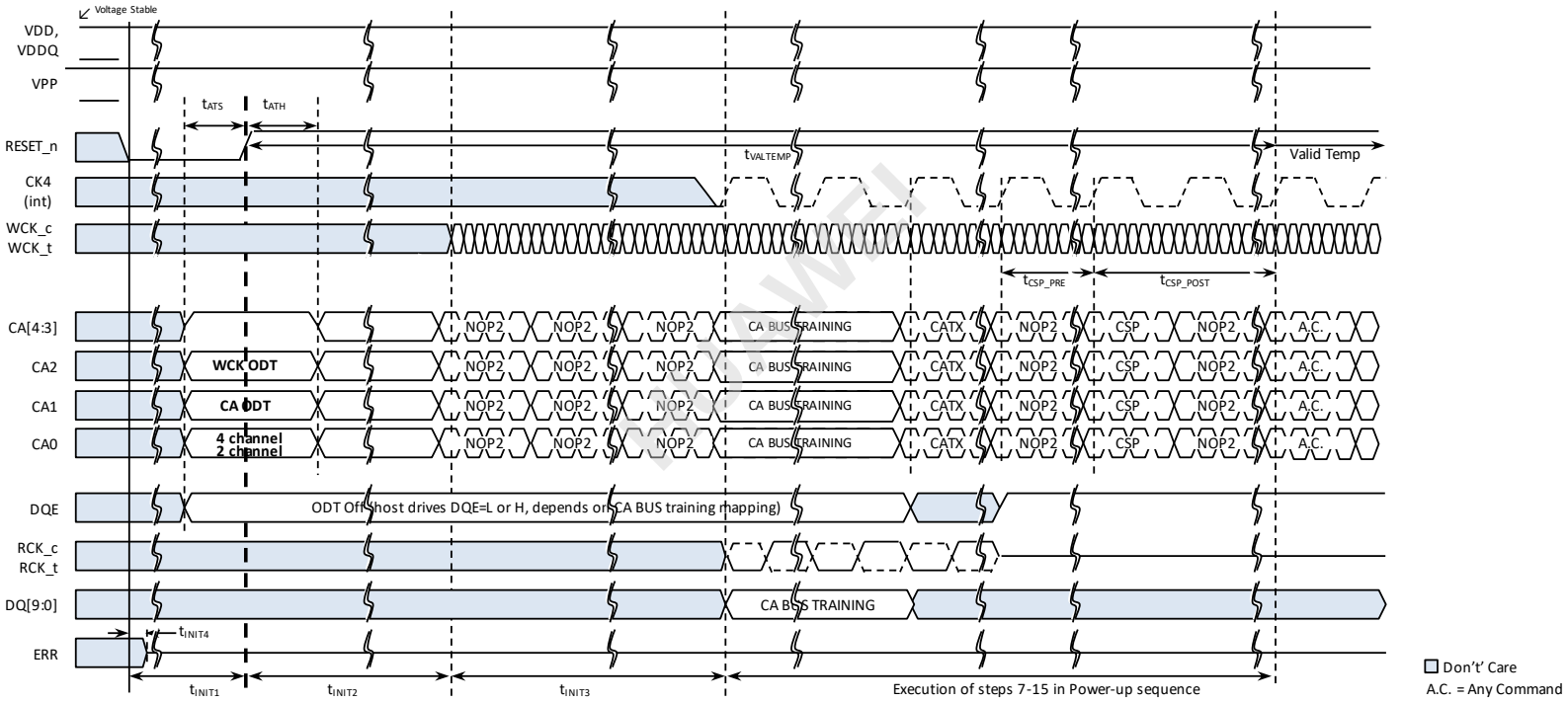
Table 18 — 4CH / 2CH Mode

CA0_A CA0_C	CA0_B CA0_D	Channel Mode
HIGH	HIGH	4 Channel Mode
LOW	X	2 Channel Mode

3.2 Initialization With Stable Power

The following sequence is required for reset after power-up initialization. This requires that the power has been stable within the specified VDD, VDDQ, and VPP ranges since power-up initialization (see *FIGURE 13):*

1) Assert RESET_n LOW anytime when reset is needed.
2) Hold RESET_n LOW for minimum tRES. Assert and hold RNOP2 and CNOP2 commands.
3) Continue with step 4 of the power-up initialization sequence.



NOTE 1 CK4 (int) is sync. for illustration. (may not be sync. before CSP Command)

NOTE 2 RCKMODE is disabled by Mode Register Default Settings.

NOTE 3 (*) Channel A and Channel C only in 2-channel mode and all Channel in 4-channel mode.

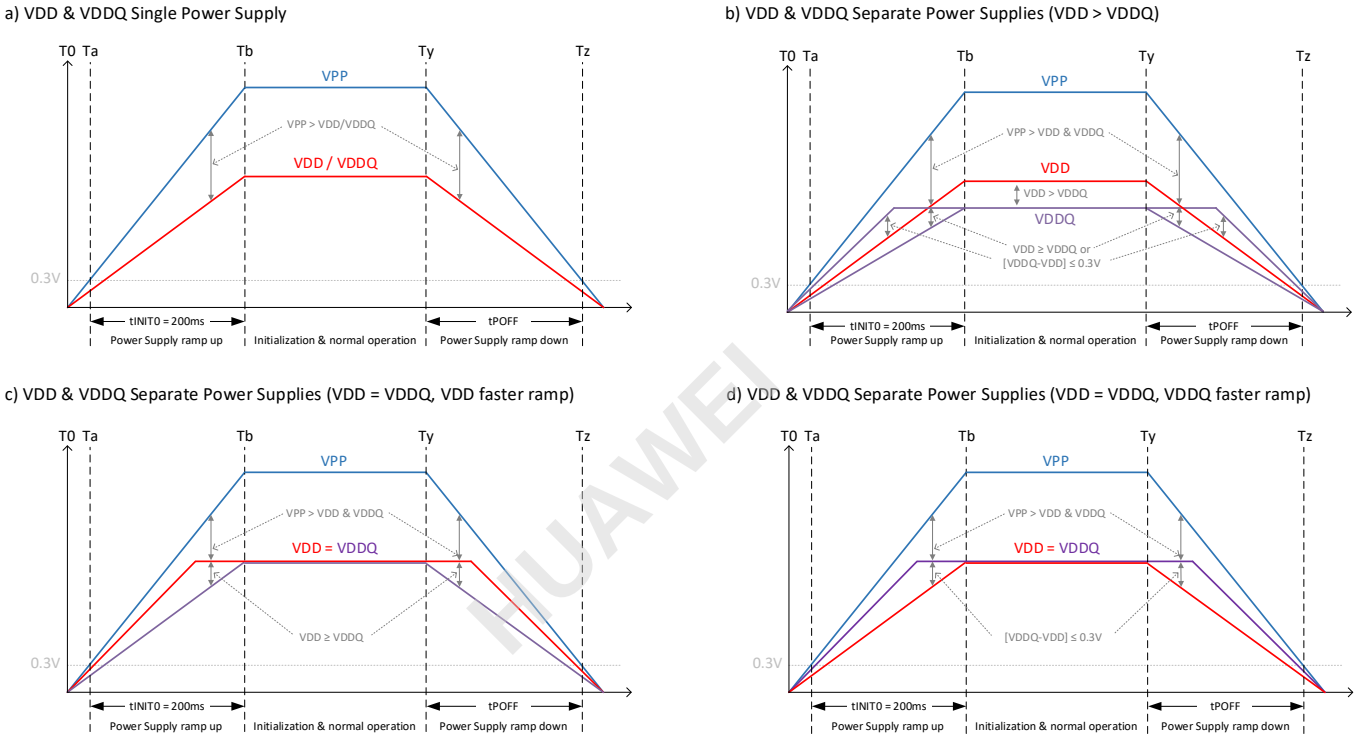
NOTE 4 Any commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 13 — Initialization with Stable Power

3.3 Controlled Power-Off Sequence

For a controlled GDDR7 power off sequence, when all the power supplies are below 0.3 V, the device is considered powered off and the VPP > VDD and VDDQ requirement does not need to be observed below that level. The controlled power-off sequence begins at the point when any supply drops below the minimum operating voltage and must be completed within tPOFF.

3.4 Operating Voltages During Power-Up And Power-Off



NOTE 1 GDDR7 SGRAMs are designed to support PCB designs with separate VDD and VDDQ power supplies.

NOTE 2 Ta is the point when any supply first reaches 0.3V. Tb is the point when all supply voltages are within their defined operating range. Ty is the point when any power supply drops below the minimum operating value. Tz is the point where all supplies are below 0.3 V.

NOTE 3 During normal operation (from Tb to Ty) the max allowed voltage difference between VDD and VDDQ is limited by the allowed operating supply ranges as per *TABLE 133 — DC OPERATING CONDITIONS*. Absolute Maximum Ratings must also be observed. For devices that support VDD > VDDQ during normal operation, [VDDQ-VDD] ≤ 0.3 V is only required during power supply ramp up and ramp down as illustrated in example b).

NOTE 4 Examples a) thru d) illustrate some of the possible scenarios for power supply ramp up and ramp down. The examples also show possible VDD and VDDQ voltage levels during normal operation that are allowed in this specification. Vendor datasheets should be consulted to see what VDD and VDDQ are supported.

Figure 14 — Examples of Power Supply Ramp-Up and Ramp-Down

4 Mode Registers

GDDR7 specifies 64 mode registers (MR0 to MR63) to define the specific mode of operation. MR0 to MR47 are defined as shown in the overview in *TABLE 19*. MR48 to MR63 are not defined and reserved for vendor specific features. Reprogramming the mode registers will not alter the contents of the memory array.

Mode registers are programmed via the Mode Register Set (MRS) command and will retain the stored information until they are reprogrammed, chip reset, or the device loses power. All mode registers must be fully initialized to the desired values upon power-up or after a subsequent chip reset as part of the initialization sequence.

Mode registers must be loaded when all banks are idle and no bursts are in progress, and the host must wait the specified time tMOD before initiating any subsequent operation. Some features require a settling time larger than tMOD as captured in the notes of the respective register fields. Violating either of these requirements will result in unspecified operation.

MRS commands to mode registers that control I/O characteristics and interface training related features may also be issued in bank idle state, as these features do not interfere with the memory array. Mode registers that can also be programmed in bank active state are:

- MR5 to MR7
- MR16 to MR21
- MR23 to MR26

No default states are defined for mode registers except for those functions that are required to be initialized at power-up or after a subsequent chip reset to prevent the DRAM from functioning improperly. Refer to *TABLE 54* for a list of default settings.

Reserved states should not be used, as unknown operation or incompatibility with future versions may result. RFU bits are reserved for future use and must be programmed to 0_B. If bits in an optional field are set, either the optional field is activated if the option is implemented in the device, or no action is taken by the device if the option is not implemented.

Mode registers 10 to 15 are so-called Frequency Dependent Mode Registers (FDMR) that can be configured such that the internal update of the related features will be postponed until after the next sleep mode or self refresh sleep mode entry. This will ensure that updates of these registers do not conflict with any ongoing operation and only change when the CA interface is disabled. The FD_FLAG bit in MR0 OP11 is associated with this function.

Conditions:

- The internal postponement is enabled when the FD_FLAG bit is set to 0_B (default). The host is allowed to issue any number of MRS commands to FDMRs while the FD_FLAG bit is programmed to 0_B. The DRAM will only store the codes issued with the last MRS commands issued to the FDMRs, and only these codes will be updated internally with a subsequent sleep mode entry.

  Example MRS command sequence: MR12(1) -> MR14(1) -> MR14(2) -> MR12(2). Code from MR12(2) and MR14(2) will be internally updated with a subsequent sleep mode entry.

- This postponement is disabled with the FD_FLAG bit set to 1_B. In this case the FDMRs behave like any other mode register and the codes will be updated internally within t_{MOD}.

4 Mode Registers (cont'd)

- A change of the FD_FLAG bit from 0_B (Enable) to 1_B (Disable) is only allowed when there are no FDMR changes waiting for the next sleep mode entry. Pending FDMR changes can be effectively cancelled by issuing MRS commands with the old, currently programmed codes. Such MRS commands must be issued before setting the FD_FLAG bit to 1_B. Failure to do so may lead to unknown behavior. The host is responsible to remember the currently programmed codes.

  Example MRS command sequence: MR12(new) -> MR14(new) -> MR14(old) -> MR12(old) -> FD_FLAG = 1_B. The change of the FD_FLAG bit from 0_B to 1_B will in this scenario preserve (not change) the content of MR12 and MR14.

- MR12 controls features that are allowed to only change in sleep mode. Changes to MR12 are therefore only allowed to be issued when the FD_FLAG bit is 0_B.

- Changes to FDMR other than MR12 are allowed with the FD_FLAG bit set to 1_B, however, functional issues resulting from such FDMR code changes cannot be excluded and therefore this setting is in general not recommended. As an example, changing WCK ODT in MR13 on-the-fly (not in sleep mode when WCK is off) could lead to glitches on the device's WCK input that could result in a device hang-up. Users shall align with the DRAM vendors on conditions and restrictions for updating any FDMR with the FD_FLAG bit set to 1_B.

Table 19 — Mode Register Overview

MR	M[5:0]	OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
MR0	00_H	FD_ FLAG	POISON	SEVE-RITY	PAM3	TM	SRSLX2CAT	SLX2 CAT	WRCRC	RDCRC	RFU	RFU	CABI
MR1	01_H	DQERL				RFU		RL					
MR2	02_H	WRCRC2ERR					RFU	WL					
MR3	03_H	RFU					RAS						
MR4	04_H	RTPSB				RFU	WR						
MR5	05_H	RFU	DQE_ HZ	SEV2 ERR	CAL_UPD		RFU	ERR Driver Strength	RFU	DQ/DQE Termination		DQ/DQE/RCK Driver Strength	
MR6	06_H	RFU				DQ/DQE/RCK Pulldown Driver Offset (Leg 2)				DQ/DQE/RCK Pulldown Driver Offset (Leg 1)			
MR7	07_H	DQ/DQE Data Termination Offset				DQ/DQE/RCK Pullup Driver Offset (Leg 2)				DQ/DQE/RCK Pullup Driver Offset (Leg 1)			
MR8	08_H	Hiber-nate	Self Refresh		RFU		DCC		BRC		DRFM	ARFM	
MR9	09_H	RCKSTOP_LAT			RFU	RCKEN					RCK_LS	RCKMODE	
MR10	$0A_H$	RFU											
MR11	$0B_H$	RFU											
MR12	$0C_H$	RCK LEVEL	RCK TYPE	RFU	WCK Frequency					VDDQ Range		VDD Range	
MR13	$0D_H$	WCK Termination Offset			WCK Termination			CA Termination Offset			CA Termination		
MR14	$0E_H$	Half VREFCA	DFECA				VREFCA						
MR15	$0F_H$	CAPAR2ERR				RFU	CAPARBLK_LAT Control	CAPARBLK_CAT			CSP_FB	CAPAR BLK	CAPAR

Table 19 — Mode Register Overview (cont'd)

MR	M[5:0]	OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
MR16	10_H	VREFDL or VREFD2 I/O Sub-Address				RFU	VREFDL or VREFD2 Reference Voltage Levels and Offset						
MR17	11_H	VREFDH I/O Sub-Address				RFU	VREFDH Reference Voltage Level and Offset						
MR18	12_H	DQ/DQE CTLE I/O Sub-Address				DQ/DQE CTLE RX Sub-Address		RFU		DQ/DQE CTLE			
MR19	13_H	DQ/DQE DFE I/O Sub-Address				DQ/DQE DFE RX Sub-Address		RFU		DQ/DQE DFE			
MR20	14_H	Scramble Code Byte Address			RFU	Read/Write Data Scramble Code							
MR21	15_H	DQ/DQE/RCK TX EQ I/O Sub-Addr.				RFU				DQ/DQE/RCK TX EQ			
MR22	16_H	ECS_ ON	ECS_ RESET	ECS_ FLAG_ RESET	ECS_ LOG_ RULES	RFU							
MR23	17_H	DT_ERR_PATTERN			FIFO_ PTR_RST	LFSR_ CNT_ MODE	CAOSC	DT_EYE_MASK		LFSR_ RST_ MODE	LFSR_ CNT_ RESET	LFSR_ TYPE	DT_ LFSR
MR24	18_H	LFSR Sub-Address		LFSR Seed Bits [9:0]									
MR25	19_H	LFSR Sub-Address		RFU				LFSR Seed Bits [14:10]					
MR26	$1A_H$	I/O Sub-Address				RFU				DT_ LANE_ MASK	LFSR_ INV	LFSR_SHIFT_SEL	
MR27	$1B_H$	RFU				ERR Pullup Driver Offset				ERR Pulldown Driver Offset			
MR28	$1C_H$	RFU											
MR29	$1D_H$	RFU								DQ Map Mode			
MR30	$1E_H$	RFU			ECC_ 2CH	ECC_PAR_SEL						ECC_ VEC	ECC_ TM
MR31	$1F_H$	hPPR	RFU			PPR Guard Key							
MR32	20_H	RFU											
…	…	…											
MR47	$2F_H$	RFU											
MR48	30_H	Reserved for Vendor Specific Features											
…	…	…											
MR63	$3F_H$	Reserved for Vendor Specific Features											

4.1 Mode Register Assignment and Definition

Table 20 — Mode Register 0 (MR0)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
FD_ FLAG	POISON	SEVERITY	PAM3	TM	SRSLX2 CAT	SLX2 CAT	WRCRC	RDCRC	RFU	RFU	CABI

Function	Operand	Data	Notes
CABI	OP0	Command/address bus inversion 0_B: Enable (default) 1_B: Disable	1, 8
RFU	OP[2:1]	00_B: Reserved	
RDCRC	OP3	Read CRC 0_B: Disable (default) 1_B: Enable	2
WRCRC	OP4	Write CRC 0_B: Disable (default) 1_B: Enable	2
SLX2CAT	OP5	Automatic exit from Sleep Mode to CA Training 0_B: Enable (default) 1_B: Disable	3
SRSLX2CAT	OP6	Automatic exit from Self Refresh Sleep Mode to CA Training 0_B: Enable (default) 1_B: Disable	3
TM	OP7	Test Mode 0_B: Normal mode (default) 1_B: Test mode	4
PAM3	OP8	Signaling for DQ[9:0] and DQE signals 0_B: NRZ (default) 1_B: PAM3	5
SEVERITY	OP9	ECC error severity reporting during Reads 0_B: Disable 1_B: Enable	6
POISON	OP10	Data poisoning for Writes and Reads 0_B: Disable 1_B: Enable	6
FD_FLAG	OP11	Flag to postpone MR10 to MR15 updates until after next Sleep mode or Self Refresh Sleep mode entry 0_B: Enable (default) 1_B: Disable	7

Table 20 — Mode Register 0 (MR0) (cont'd)

NOTE 1	Refer to the *COMMAND ADDRESS BUS INVERSION (CABI)* section for details.
NOTE 2	Refer to the *READ* and Write CRC section for details.
NOTE 3	Refer to the *SLEEP MODES* and *COMMAND ADDRESS BUS TRAINING* sections for details.
NOTE 4	The normal operating mode is selected by issuing an MRS command with bit OP7 set to 0_B, and bits OP[6:0] and OP[11:8] set to the desired values. Programming bit OP7 to 1_B places the device into a test mode that is only to be used by the DRAM manufacturer. No functional operation is specified with test mode enabled.
NOTE 5	The bit determines the signaling of the DQ and DQE signals. It also selects the reference voltages for DQ and DQE inputs as programmed in MR16 and MR17. Refer to the *PAM3* section for details.
NOTE 6	Refer to the *ON-DIE ECC* section for details.
NOTE 7	Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B.
NOTE 8	With a change to the CABI register, it is required to wait tMOD instead of tMRD after an MRS command that changes CABI and any subsequent MRS command. During tMOD , RNOP1/CNOP1 is required to be issued with all "H", except for the CAPAR bit.

Table 21 — Mode Register 1 (MR1)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
DQERL				RFU		RL					

Function	Operand	Data	Notes
RL	OP[5:0]	Read latency 001010_B: 10 001011_B: 11 . . . 101110_B: 46 101111_B: 47 All others: reserved	1, 2
RFU	OP[7:6]	00_B: Reserved	
DQERL	OP[11:8]	DQE additional read latency 0000_B: 0 0001_B: 1 . . . 1011_B: 11 1100_B: 12 All others: reserved	1

NOTE 1	GDDR7 SGRAMs are not required to support all encodings but the supported min-max range must be contiguous. Refer to the *READ* section for details concerning the RL and DQERL latency definitions.
NOTE 2	RL is the programmed value. See the *READ* section for the definition of Read latency when CAPARBLK is disabled or enabled as well as if enabled whether the DRAM implements implicit, explicit CAPARBLK latency or both.

4.1 Mode Register Assignment and Definition (cont'd)

Table 22 — Mode Register 2 (MR2)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
WRCRC2ERR					RFU		WL				

Function	Operand	Data	Notes
WL	OP[4:0]	Write latency 00110_B: 6 00111_B: 7 . . . 11110_B: 30 11111_B: 31 All others: reserved	1, 2
RFU	OP[6:5]	000_B: Reserved	
WRCRC2ERR	OP[11:7]	CRC write error latency 01010_B: 10 01011_B: 11 . . . 11110_B: 30 11111_B: 31 All others: reserved	1

NOTE 1 GDDR7 SGRAMs are not required to support all encodings but the supported min-max range must be contiguous. Refer to the *WRITE* section for details concerning the WL and WRCRC2ERR latency definitions.

NOTE 2 WL is the programmed value. See the *WRITE* section for the definition of Write latency when CAPARBLK is disabled or enabled as well as if enabled whether the DRAM implements implicit, explicit CAPARBLK latency or both.

4.1 Mode Register Assignment and Definition (cont'd)

Table 23 — Mode Register 3 (MR3)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU					RAS						

Function	Operand	Data	Notes
RAS	OP[6:0]	RAS 0000100_B: 4 0000101_B: 5 . . . 1001111_B: 79 1010000_B: 80 All others: reserved	1, 2
RFU	OP[11:7]	00000_B: Reserved	

NOTE 1 GDDR7 SGRAMs are not required to support all encodings but the supported min-max range must be contiguous. Refer to the *ROW ACTIVATION* command section for details.

NOTE 2 RAS must be programmed with a value greater than or equal to RU{tRAS/tCK4}, where RU stands for round up, tRAS is the analog value from the vendor datasheet and tCK4 is the cycle time of the internal CK4 clock. If a GDDR7 SGRAM does not support the mode register definition of tRAS in clock cycles, the RAS mode register settings will be ignored.

Table 24 — Mode Register 4 (MR4)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RTPSB				RFU	WR						

Function	Operand	Data	Notes
WR	OP[6:0]	Write Recovery for Auto Precharge 0000100_B: 4 0000101_B: 5 . . . 1001111_B: 79 1010000_B: 80 All others: reserved	1, 2
RFU	OP7	0_B: Reserved	
RTPSB	OP[11:8]	Read to Auto Precharge (same bank) 0100_B: 4 0101_B: 5 . . . 1110_B: 14 1111_B: 15 All others: reserved	1, 3

NOTE 1 GDDR7 SGRAMs are not required to support all encodings but the supported min-max range must be contiguous.

NOTE 2 WR must be programmed with a value greater than or equal to RU{tWR/tCK4}, where RU stands for round up, tWR is the analog value from the vendor datasheet and tCK4 is the cycle time of the internal CK4 clock. If a GDDR7 SGRAM does not support the mode register definition of tWR in clock cycles, the WR mode register settings will be ignored.

NOTE 3 RTP must be programmed with a value greater than or equal to RU{tRTPSB/tCK4}, where RU stands for round up, tRTPSB is the analog value from the vendor datasheet and tCK4 is the cycle time of the internal CK4 clock. If a GDDR7 SGRAM does not support the mode register definition of tRTPSB in clock cycles, the RTPSB mode register settings will be ignored.

4.1 Mode Register Assignment and Definition (cont'd)

Table 25 — Mode Register 5 (MR5)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU	DQE_HZ	SEV2 ERR	CAL_UPD		RFU	ERR Driver Strength	RFU	DQ/DQE Termination		DQ/DQE/RCK Driver Strength	

Function	Operand	Data	Notes
DQ/DQE/RCK Driver Strength	OP[1:0]	Driver strength for DQ[9:0], DQE and RCK signals 00_B: 40 Ohm, auto-calibrated (default) 01_B: 80 Ohm, auto-calibrated 10_B: Vendor specific (optional) 11_B: Vendor specific (optional)	1
DQ/DQE Data Termination	OP[3:2]	Data termination for DQ[9:0] and DQE signals 00_B: Disable (data termination off; default) 01_B: 40 Ohm 10_B: 48 Ohm 11_B: 80 Ohm	2
RFU	OP4	0_B: Reserved	
ERR Driver Strength	OP5	Driver Strength for ERR output 0_B: 40 Ohm, auto-calibrated (default) 1_B: 80 Ohm, auto-calibrated	1
RFU	OP6	0_B: Reserved	
CAL_UPD	OP[8:7]	Calibration update 00_B: Enable for all signals (default) 01_B: Disable for WCK_t/WCK_c only (optional) 10_B: Reserved 11_B: Disable for all signals	~~3~~
SEV2ERR	OP9	Severity and error signaling on ERR output during Read bursts (optional) 0_B: Disable (default) 1_B: Enable	4, 5
DQE_HZ	OP10	DQE in High-Z State (optional) 0_B: Disable (default) 1_B: Enable	6, 7
RFU	OP11	RFU	

Table 25 — Mode Register 5 (MR5) (cont'd)

NOTE 1	The 40 Ohm default driver strength is intended for normal operation, and the 80 Ohm driver strength may be used at lower data rates, depending on system characteristics. Both settings utilize the Auto Calibration functionality when enabled by the CAL_UPD bit in OP11. For DQ/DQE and RCK, vendors may optionally use the other settings to specify additional driver strengths.
NOTE 2	All DQ/DQE termination settings utilize the Auto Calibration functionality when enabled by the CAL_UPD bit in OP[8:7].
NOTE 3	The Calibration Update setting enables the calibration value to be updated automatically by the auto calibration engine. The function is enabled upon power-up to reduce update induced jitter. The user may decide to suppress updates from the auto calibration engine by disabling Calibration Update for WCK_t/_c only when this option is supported, or for all signals. The calibration updates can occur with any REFab command. The update is not complete for a time tKO after the latching of the REFab command. During this tKO time, only RNOP2 and CNOP2 commands may be issued.
NOTE 4	Vendor ID3 (IRA 2) bit DQ0 identifies whether the optional SEV2ERR feature is supported or not.
NOTE 5	When enabled, the severity flag during read burst that normally is transmitted on the DQE signal will be transmitted on the ERR signal. This feature may only be enabled when the signaling is set to NRZ mode (MR0 OP8), and when RDCRC (MR0 OP3), WRCRC (MR0 OP4) and Poison (MR0 OP10) are all disabled and Severity (MR0 OP9) is enabled.
NOTE 6	Vendor ID3 (IRA 2) bit DQ5 identifies whether the optional DQE_HZ feature is supported or not.
NOTE 7	When enabled, the DQE signal will be in High-Z state. The optional feature is intended to be used only when the signaling is set to NRZ mode (MR0 OP8), when RDCRC (MR0 OP3), WRCRC (MR0 OP4) and Poison (MR0 OP10) are all disabled, and when either Severity (MR0 OP9) is disabled as well or being rerouted to the ERR signal with SEV2ERR set to 1_8.

4.1 Mode Register Assignment and Definition (cont'd)

Table 26 — Mode Registers 6 (MR6)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU				DQ/DQE/RCK PD Driver Offset (Leg 2)				DQ/DQE/RCK PD Driver Offset (Leg 1)			

Function	Operand	Data	Notes
DQ/DQE/RCK Pulldown Offset (Leg 1)	OP[3:0]	Leg 1 pulldown driver offset for DQ[9:0], DQE and RCK signals 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	1, 2
DQ/DQE/RCK Pulldown Offset (Leg 2)	OP[7:4]	Leg 2 pulldown driver offset for DQ[9:0], DQE and RCK signals 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	1, 2
RFU	OP[11:8]	0000_B: Reserved	

NOTE 1 Refer to *FIGURE 15* for an example driver implementation including the definition of "Leg 1" and "Leg 2".

NOTE 2 The offset impedance step values may be non-linear and will vary across DRAM vendors and across PVT. With negative offset steps the driver strength will be decreased, and RON will be increased; with positive offset steps the driver strength will be increased, and RON will be decreased.

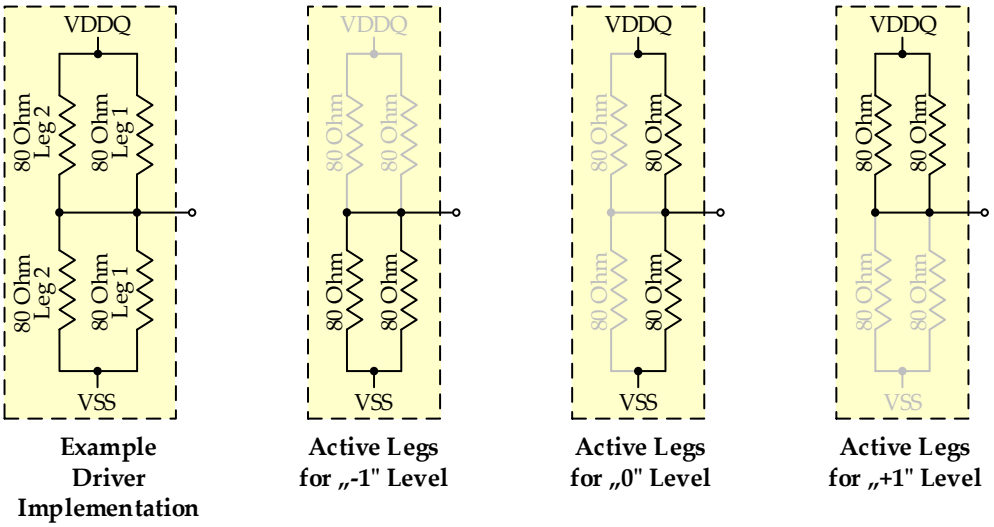


Figure 15 — Example Driver Implementation and Active Legs in PAM3 Mode

4.1 Mode Register Assignment and Definition (cont'd)

Table 27 — Mode Registers 7 (MR7)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
DQ/DQE Data Termination Offset				DQ/DQE/RCK PU Driver Offset (Leg 2)				DQ/DQE/RCK PU Driver Offset (Leg 1)			

Function	Operand	Data	Notes
DQ/DQE/RCK Pullup Offset (Leg 1)	OP[3:0]	Leg 1 pullup driver offset for DQ[9:0], DQE and RCK signals 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	1, 2
DQ/DQE/RCK Pullup Offset (Leg 2)	OP[7:4]	Leg 2 pullup driver offset for DQ[9:0], DQE and RCK signals 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	1, 2
DQ/DQE Data Termination Offset	OP[11:8]	Data termination offset for DQ[9:0] and DQE signals 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	3

NOTE 1 Refer to *FIGURE 15* for an example driver implementation including the definition of "Leg 1" and "Leg 2".

NOTE 2 The offset impedance step values may be non-linear and will vary across DRAM vendors and across PVT. With negative offset steps the driver strength will be decreased, and RON will be increased; with positive offset steps the driver strength will be increased, and RON will be decreased.

NOTE 3 The offset impedance step values may be non-linear and will vary across DRAM vendors and across PVT. With negative offset steps the termination strength will be decreased and RTT will be increased; with positive offset steps the termination strength will be increased and RTT will be decreased.

4.1 Mode Register Assignment and Definition (cont'd)

Table 28 — Mode Register 8 (MR8)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
Hibernate	Self Refresh		RFU		DCC		BRC		DRFM	ARFM	

Function	Operand	Data	Notes
ARFM	OP[1:0]	Refresh Management Level 00_B: Default level (use of RFM may be required or not) 01_B: Level A (RFM is required) 10_B: Level B (RFM is required) 11_B: Level C (RFM is required)	1
DRFM	OP2	Directed Refresh Management (DRFM) 0_B: Disable (default) 1_B: Enable	2
BRC	OP[4:3]	Bounded Refresh Configuration (BRC). Rows refreshed: 00_B: BRC2: always ±1, ratio ±2 01_B: BRC3: always ±1, ±2, ratio ±3 (optional) 10_B: BRC4: always ±1, ±2, ±3, ratio ±4 (optional) 11_B: Reserved	2
DCC	OP[6:5]	Duty Cycle Corrector (optional) 00_B: Disable (default) 01_B: Start 10_B: Reserved 11_B: Hold	3, 4
RFU	OP[8:7]	00_B: Reserved	
Self Refresh	OP[10:9]	Self Refresh 00_B: Fixed self refresh interval (default) 01_B: Vendor specific fixed refresh rate (optional) 10_B: Vendor specific fixed refresh rate (optional) 11_B: Temperature compensated self refresh rate	5
Hibernate	OP11	Hibernate self refresh sleep mode 0_B: Disable (default) 1_B: Enable (bit is self-clearing)	6

NOTE 1 The support of Adaptive Refresh Management (ARFM) is optional for the DRAM vendor. GDDR7 SGRAMs not supporting ARFM will define these bits as RFU. Refer to the *ADAPTIVE REFRESH MANAGEMENT (ARFM)* section for details.

NOTE 2 Refer to the *DIRECTED REFRESH MANAGEMENT (DRFM)* section for details.

NOTE 3 The DCC bits control the operation of the optional duty cycle corrector (DCC). The DCC can be used to cancel out a static duty cycle error on the WCK clocks. The settings will be ignored if a GDDR7 SGRAM does not include a DCC. Refer to the *DUTY CYCLE CORRECTOR (DCC)* section for details.

NOTE 4 Vendor ID3 (IRA 2) bit DQ4 identifies whether the optional DCC is supported or not.

NOTE 5 The refresh interval in Self Refresh mode may be a fixed value determined by the DRAM vendor or being controlled by an integrated temperature sensor. DRAM vendors may support additional fixed refresh rate settings related to other temperatures. Refer to the *SELF REFRESH* section for details.

NOTE 6 With bit OP11 set to 1_B, the device enters Hibernate Self Refresh Sleep mode with the next SELF REFRESH SLEEP ENTRY command. The bit is self-clearing meaning that it returns to the default 0_B upon exit from Hibernate Self refresh sleep mode. Refer to the *HIBERNATE SELF REFRESH SLEEP MODE* section for details.

4.1 Mode Register Assignment and Definition (cont'd)

Table 29 — Mode Register 9 (MR9)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RCKSTOP_LAT			RFU	RCKEN					RCK_LS	RCKMODE	

Function	Operand	Data	Notes
RCKMODE	OP[1:0]	RCK mode 00_B: Disable (default) 01_B: Start with Read (RD, RDA, IRD, RDTR or RDWTEC) command 10_B: Start with RCKSTRT command 11_B: Enable (always on)	1, 2,3
RCK_LS	OP2	Number of half frequency RCK preamble cycles 0_B: 0 1_B: 2	3, 4
RCKEN	OP[7:3]	RCK enable latency 00000_B: 6 00001_B: 7 . . . 11110_B: 36 11111_B: 37	3, 4, 5, 6
RFU	OP8	0_B: Reserved	
RCKSTOP_LAT	OP[11:9]	RCK stop latency 000_B: 10 001_B: 12 010_B: 14 011_B: 16 100_B: 18 101_B: 20 110_B: 22 111_B: 24	3, 4, 7, 8

NOTE 1 The RCKMODE and RCKTYPE (in MR12 OP10) settings define the operation of the RCK_t and RCK_c outputs. Both RCK_t and RCK_c are disabled by default. RCK can be configured to be pseudo-differential or single ended. It can also be configured to be always on, or be started with any Read command (RD, RDA, IRD, RDTR or RDWTEC) or with the RCKSTRT command.

NOTE 2 A change of RCKMODE from either 00B (disable) to 11B (always on) or back to 00B (disable) requires tMODRCK time instead of tMOD for the RCK_t and RCK_c outputs to settle.

NOTE 3 The RCK related settings in MR9 and MR12 are ignored but preserved when in CA Training mode. See *COMMAND ADDRESS BUS TRAINING* for details on RCK in that mode. The RCK returns to the programmed settings upon exit from CA Training mode.

NOTE 4 The RCK_LS, RCKEN and RCKSTOP_LAT settings define latencies associated with starting and stopping RCK. GDDR7 SGRAMs are not required to support all RCKEN and RCKSTOP_LAT latencies but the supported min-max range must be contiguous. For RCKEN, vendors may support even latencies only (OP3 = 0B). See *READ CLOCK (RCK)* section and the vendor datasheets for details.

NOTE 5 The latency values may have vendor specific frequency limits.

NOTE 6 RCKEN is the programmed value. See the *READ CLOCK (RCK)* section for the definition of RCK enabled latency when CAPARBLK is disabled or enabled as well as if enabled whether the DRAM implements implicit, explicit CAPARBLK latency or both.

NOTE 7 The latency values may have vendor specific upper frequency limits.

NOTE 8 RCKSTOP_LAT is the programmed value. See the *READ CLOCK (RCK)* section for the definition of RCK stop latency when CAPARBLK is disabled or enabled as well as if enabled whether the DRAM implements implicit, explicit CAPARBLK latency or both.

4.1 Mode Register Assignment and Definition (cont'd)

Table 30 — Mode Register 10 (MR10)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU											

Function	Operand	Data	Notes
RFU	OP[11:0]	0000 0000 0000_B: Reserved	1

NOTE 1 Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B.

Table 31 — Mode Register 11 (MR11)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU											

Function	Operand	Data	Notes
RFU	OP[11:0]	0000 0000 0000_B: Reserved	1

NOTE 1 Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B.

4.1 Mode Register Assignment and Definition (cont'd)

Table 32 — Mode Register 12 (MR12)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RCK LEVEL	RCK TYPE	RFU	WCK Frequency					VDDQ Range		VDD Range	

Function	Operand	Data	Notes
VDD Range	OP[1:0]	VDD Range 00_B: Default level (default) 01_B: Voltage level lower than 00_B (optional) 10_B: Voltage level lower than 01_B (optional) 11_B: Voltage level lower than 10_B (optional)	1, 2
VDDQ Range	OP[3:2]	VDDQ Range 00_B: Default level (default) 01_B: Voltage level lower than 00_B (optional) 10_B: Voltage level lower than 01_B (optional) 11_B: Voltage level lower than 10_B (optional)	1, 2
WCK Frequency	OP[8:4]	WCK Frequency range 00000_B: $f_{WCK} \leq f_{WCKNRZ}$ (default) 00001_B: f_{WCK} <0.5 GHz (optional) 00010_B: 0.5 GHz ≤ f_{WCK} <1.0 GHz (optional) 00011_B: 1.0 GHz ≤ f_{WCK} <1.5 GHz (optional) . . . 01110_B: 6.5 GHz ≤ f_{WCK} <7.0 GHz (optional) 01111_B: 7.0 GHz ≤ f_{WCK} <7.5 GHz (optional) 10000_B: 7.5 GHz ≤ f_{WCK} <8.0 GHz (optional) . . . 11000_B: 11.5 GHz ≤ f_{WCK} <12.0 GHz (optional) All others: Reserved	1, 3
RFU	OP9	0_B: Reserved	
RCKTYPE	OP10	RCK type 0_B: Single-ended (RCK_c output is in High-Z state) 1_B: Differential	1, 4, 5, 6
RCKLEVEL	OP11	RCK level in PAM3 mode 0_B: Full swing (+1/-1 levels) 1_B: Half swing (+1/0 levels)	1, 5, 7

NOTE 1 Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B. Changes in MR12 are only allowed with the FD_FLAG bit set to 0_B.

NOTE 2 The VDD and VDDQ Range bits are used to adapt DRAM characteristics like internal supply voltages when the actual V_{DD} or VDDQ is lower than the default operating range. The default 00_B setting represents the highest VDD and VDDQ supply voltage range supported by the device. All other field values are optional, and the VDD and VDDQ ranges of the optional fields represent supply voltage ranges lower than the default range in a decreasing voltage order. The actual VDD and VDDQ supply voltage ranges itself are vendor specific. The VDD and VDDQ Range settings must be the same on all four channels.

NOTE 3 The WCK Frequency setting is used to adapt DRAM characteristics to explicit WCK frequency ranges. The default setting is provided to support initial device initialization in NRZ mode. The frequency increment is 0.5GHz which is equivalent to a data symbol rate of 1 GBaud in PAM3 mode. The maximum supported encoding shall equal the maximum supported WCK frequency of the device. The settings will be ignored if a GDDR7 SGRAM does not support the mode register definition of the WCK frequency. Programming the default WCK Frequency setting (MR12 OP[8:4] = 00000b) after the initialization sequence has completed will lead to operation not guaranteed.

NOTE 4 The RCKMODE (in MR9 OP[1:0]) and RCKTYPE settings define the operation of the RCK_t and RCK_c outputs. Both RCK_t and RCK_c are disabled by default. RCK can be configured to be pseudo-differential or single ended. It can also be configured to be always on, or be started with any Read command (RD, RDA, IRD, RDTR or RDWTEC) or with the RCKSTRT command.

NOTE 5 The RCK related settings in MR9 and MR12 are ignored but preserved when in CA Training mode. See *COMMAND ADDRESS BUS TRAINING* for details on RCK in that mode. The RCK returns to the programmed settings upon exit from CA Training mode.

NOTE 6 The single-ended option may have a vendor specific upper frequency limit.

NOTE 7 The RCKLEVEL setting is valid in PAM3 mode only and ignored in NRZ mode. The 1_B setting may have a vendor specific upper frequency limit.

4.1 Mode Register Assignment and Definition (cont'd)

Table 33 — Mode Register 13 (MR13)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
WCK Termination Offset			WCK Termination			CA Termination Offset			CA Termination		

Function	Operand	Data	Notes
CA Termination	OP[2:0]	Termination for CA[4:0] inputs 000_B: value latched on CA1 input at exit from reset (default) 001_B: 48 Ohm 010_B: 60 Ohm 011_B: 96 Ohm 101_B: 80 Ohm(optional) 111_B: Disable (CA ODT off) All others: Reserved	1, 2
CA Termination Offset	OP[5:3]	Termination Offset for CA[4:0] inputs 000_B: 0 (no offset; default) 001_B: +1 010_B: +2 011_B: +3 100_B: -4 101_B: -3 110_B: -2 111_B: -1	1, 4
WCK Termination	OP[8:6]	Termination for WCK_t and WCK_c inputs 000_B: value latched on CA2 input at exit from reset (default) 001_B: 48 Ohm 010_B: 60 Ohm 011_B: 96 Ohm 100_B: 40 Ohm 101_B: 80 Ohm(optional) 111_B: Disable (WCK ODT off) All others: Reserved	1, 2
WCK Termination Offset	OP[11:9]	Termination Offset for WCK_t and WCK_c inputs 000_B: 0 (no offset; default) 001_B: +1 010_B: +2 011_B: +3 100_B: -4 101_B: -3 110_B: -2 111_B: -1	1, 3

NOTE 1 Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B.

NOTE 2 The 000_B encodings hold the values that were latched at exit from reset state.

NOTE 3 The offset impedance step values may be non-linear and will vary across DRAM vendors and across PVT. With negative offset steps the termination strength will be decreased and RTT will be increased; with positive offset steps the termination strength will be increased and RTT will be decreased.

4.1 Mode Register Assignment and Definition (cont'd)

Table 34 — Mode Register 14 (MR14)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
Half VREFCA	DFECA				VREFCA						

Function	Operand	Data	Notes
VREFCA	OP[6:0]	Reference voltage for CA[4:0] inputs Default: 0.725 × V_{DDQ} (0101111_B)	1, 2
DFECA	OP[10:7]	DFE for CA[4:0] inputs 0000_B: No DFE (default) 0001_B: ±1 VREFCA step 0010_B: ±2 VREFCA steps . . . 1110_B: ±14 VREFCA steps 1111_B: ±15 VREFCA steps	1, 3
Half VREFCA	OP11	Reference voltage for CA[4:0] inputs 0_B: V_{REFCA} level: reference voltage programmed in OP[6:0] (default) 1_B: V_{REFCA2} level: 0.5 × V_{DDQ}	1, 4

NOTE 1: Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B.

NOTE 2: The reference voltage for the CA inputs (VREFCA) is generated internally, and a common VREFCA circuit is associated with the five CA[4:0] inputs. The VREFCA level is linear with a total range of 96 steps and a nominal step size of 1/200 × VDDQ (or 0.5%) in a range from 0.490 × VDDQ to 0.965 × VDDQ as illustrated in *TABLE 35* and *FIGURE 17*. The mid-point of 0.725 × VDDQ has been set as default to match the ideal vertical data eye center with nominal 48 Ohms termination and 40 Ohms host-side pulldown driver strength. The setting is highlighted in bold in *TABLE 35*. The VREFCA settling time tVREFCA is a constant value for the device; it is referenced from the MRS command to when the 90% level of the delta between old and new VREFCA voltage has been reached as illustrated in *FIGURE 18*.

NOTE 3: The CA input data eye may be closed at the DRAM due to Inter Symbol Interference (ISI) on the channel. GDDR7 therefore provides means for improving (or opening up) the CA data eye at the receiver by the use of a 1-tap Decision Feedback Equalization (DFE). DFE must be set by programming bits MR14 OP[10:7]. The DFE settings shall be equivalent to a positive and negative shift of the programmed VREFCA level in steps of approximately +/-0.5% × VDDQ or +/-6mV as shown in *FIGURE 17*. The programmed VREFCA level plus DFE voltage cannot be outside the range of the VREFCA level supported in MR14 OP[6:0]. The actual DFE implementation is vendor specific. The VREFCA settling time tVREFCA must be satisfied for any DFE changes to settle.

NOTE 4: Half VREFCA mode enables the VREFCA level to be adjusted when the CA inputs operate without termination. With bit OP11 set to 1_B a level of nominally 0.5 x VDDQ is generated. The maximum operating frequency for this mode is defined by fWCKVREFC2. Disabling Half VREFCA mode restores the programmed VREFCA level and DFECA values. A Half VREFCA mode reference voltage change requires a time tVREFCA2 to settle, instead of tMOD.

Table 35 — VREFCA Level (V_{DDQ} = 1.20 V)

VREFCA Level Code		VREFCA Level		
MR14 OP[6:0]	Decimal	% V_{DDQ}	Divider	V_{REFCA} [V]
1 0 1 1 1 1 1	95	0.965	193 / 200	1.158
1 0 1 1 1 1 0	94	0.960	192 / 200	1.152
1 0 1 1 1 0 1	93	0.955	191 / 200	1.146
…	…	…	…	…
0 1 1 0 0 0 0	48	0.730	146 / 200	0.876
0 1 0 1 1 1 1	**47**	**0.725**	**145 / 200**	**0.870**
0 1 0 1 1 1 0	46	0.720	144 / 200	0.864
…	…	…	…	…
0 0 0 0 0 1 0	2	0.500	100 / 200	0.600
0 0 0 0 0 0 1	1	0.495	99 / 200	0.594
0 0 0 0 0 0 0	0	0.490	98 / 200	0.588

4.1 Mode Register Assignment and Definition (cont'd)

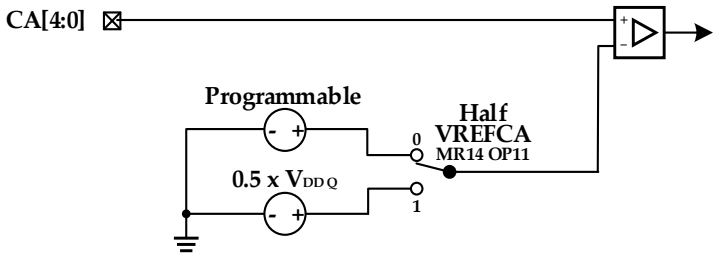


Figure 16 — VREFCA Options

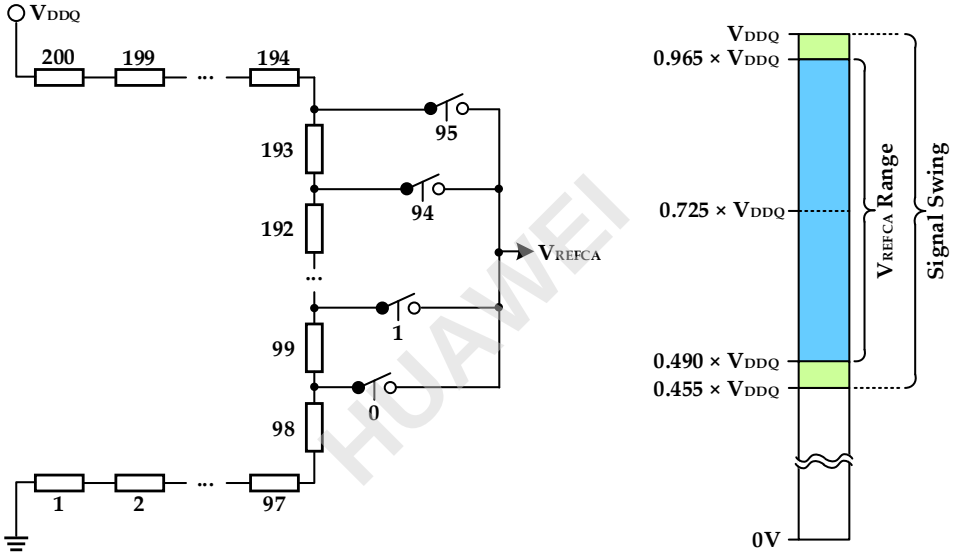


Figure 17 — VREFCA Circuit and Range

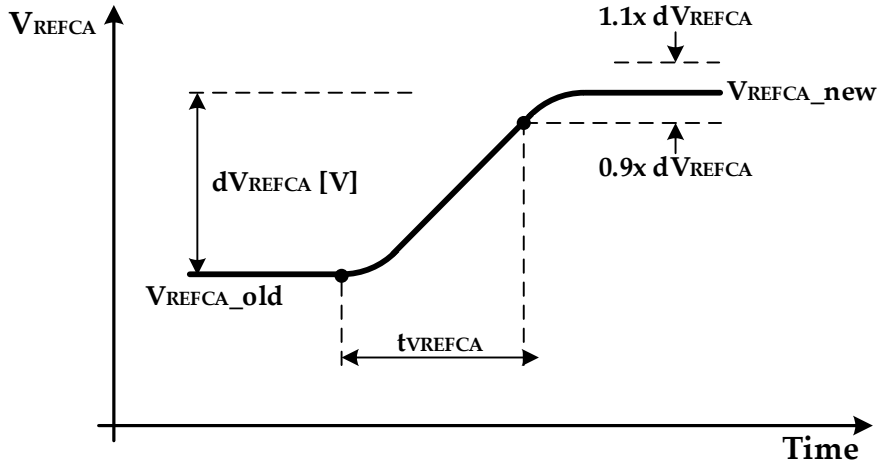


Figure 18 — VREFCA Settling Time

4.1 Mode Register Assignment and Definition (cont'd)

Table 36 — Mode Register 15 (MR15)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
CAPAR2ERR				RFU	CAPARBLK_LAT Control	CAPARBLK_LAT			CSP_FB	CAPARBLK	CAPAR

Function	Operand	Data	Notes
CAPAR	OP0	Command/address parity 0_B: Disable (default) 1_B: Enable	1, 2, 3
CAPARBLK	OP1	Command/address parity blocking. The bit must be set to 0_B when CAPAR is disabled in OP0. 0_B: Disable (default) 1_B: Enable	1, 2, 3
CSP_FB	OP2	Feedback on execution of a CSP command (optional) 0_B: Disable (default) 1_B: Enable	4
CAPARBLK_LAT	OP[5:3]	CA parity error latency 000_B: 0 (Implicit) 001_B: 1 (optional) 010_B: 2 011_B: 3 100_B: 4 101_B: Reserved 110_B: Reserved 111_B: Reserved	5
CAPARBLK_LAT Control (optional)	OP[6]	CA parity error latency 0_B: Implicit CAPARBLK_LAT (default) 1_B: Explicit CAPARBLK_LAT	6
RFU	OP[7]	0_B: Reserved	
CAPAR2ERR	OP[11:8]	CA parity error latency 0000_B: Variable latency (range: 1 to 15) 0001_B: 1 . . . 1110_B: 14 1111_B: 15	1, 2, 3, 7, 8, 9

NOTE 1 Any update of bits in mode registers 10 to 15 (MR10 to MR15) will internally be postponed until after the next Sleep mode or Self Refresh Sleep mode entry when the FD_FLAG bit in MR0 OP11 is set to 0_B. This internal postponement is disabled with the FD_FLAG bit set to 1_B.

NOTE 2 Refer to the *ERR SIGNAL* and *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* sections for details on CA parity, command blocking and CA parity error reporting.

NOTE 3 Updates of the CAPAR, CAPARBLK and CAPAR2ERR fields require a settling time of tMOD15.

NOTE 4 Refer to the CSP Feedback section for details. Vendor ID3 (IRA 2) bit DQ1 identifies whether the optional feature is supported or not.

NOTE 5 GDDR7 DRAM supports either explicit or implicit CAPARBLK_LAT. IRA3 DQ[3:2] enables host to determine which is implemented in the device. For implicit the register will be ignored. For explicit, the host is required to program MR15 OP[5:3] to a non 000_B value per the Latency Timings table (Part 1 Explicit CA blocking latency). If the GDDR7 DRAM supports explicit CAPARBLK_LAT and host sets MR15 OP[5:3] to 000_B, CAPAR with command blocking will be turned off, because it overrides MR15_OP1. Refer to the *CAPAR WITH COMMAND BLOCKING (CAPARBLK)* section for details.

NOTE 6 If GDDR7 DRAM supports only implicit or explicit (IRD3 DQ[3:2]=11_B or 10_B), this register is ignored, and DRAM operates in the supported CAPARBLK_LAT mode.

NOTE 7 GDDR7 SGRAMs may support the variable latency setting (0000_B), one or more fixed latency settings (0001_B to 1111_B), or both. In case of fixed latencies, it is required that the supported min-max range is contiguous.

NOTE 8 The variable latency setting means that the latency can change on-the-fly during normal operation as a result of temperature and voltages changes. The clock cycle at which such a latency change occurs is determined by the DRAM and not controllable by the host. The effective latency range with variable latency can be calculated as RU{tWCK2ERRINT / tCK4} + tWCK2ERRO, where RU stands for round up, tWCK2ERRINT is the minimum or maximum internal WCK-to-CA parity error delay with variable latency, tCK4 is the cycle time of the internal CK4 clock, and tWCK2ERRO is the WCK to ERR offset as measured at the DRAM balls. Refer to the vendor's datasheet for the tWCK2ERRINT and tWCK2ERRO values.

NOTE 9 The effective latency for the fixed latency settings can be calculated as CAPAR2ERR × tCK4 + tWCK2ERRO, where tCK4 is the cycle time of the internal CK4 clock, and tWCK2ERRO is the WCK to ERR offset as measured at the DRAM balls. Refer to the vendor's datasheet for the supported latencies and tWCK2ERRO value.

4.1 Mode Register Assignment and Definition (cont'd)

Table 37 — Mode Registers 16 (MR16)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
VREFDL or VREFD2 I/O Sub-Address				RFU	VREFDL or VREFD2 Reference Voltage Levels and Offset						

Function	Operand	Data	Notes
VREFDL or VREFD2 Reference Voltage	OP[6:0]	Reference voltage level and offset for lower sub-receiver for DQ[9:0] and DQE inputs in PAM3 mode, and reference voltage level for DQ[7:0] and DQE receivers in NRZ mode (no offset) See *TABLE 39* for details	1, 3, 4
RFU	OP7	0_B: Reserved	
VREFDL or VREFD2 I/O Sub-Address	OP[11:8]	I/O sub-address for VREFDL or VREFD2 reference voltage level and offset 0000_B: DQ0 0001_B: DQ1 . . . 0110_B: DQ6 0111_B: DQ7 1000_B: DQ8 1001_B: DQ9 1010_B: DQE 1100_B: DQ[7:0] and DQE (NRZ mode) 1110_B: Common offset for DQ[9:0] and DQE 1111_B: All DQ[9:0] and DQE All others: reserved	2

NOTE 1 The reference voltages for the DQ[9:0] and DQE inputs is generated internally, and separate VREF circuits are associated with each data input signal. The VREF levels itself are linear with a total range of 128 steps and a nominal step size of 3.3 mV for VDDQ = 1.20 V as listed in *TABLE 39* and illustrated in *FIGURE 21*. The levels scale linearly with VDDQ. The ranges for the VREFDL and VREFDH sub-receivers overlap as illustrated in *FIGURE 21*.

The actual VREFDL voltage for PAM3 mode is determined by the sum of an offset (range 0 to 31) and a level (range 0 to 127) as shown in *FIGURE 20*. The sum of both is calculated for each input and is bounded by the lowest and highest available levels. Using an offset allows to shift the effective VREF voltage up and down for all inputs via a single MRS command while any programmed per-DQ level is being preserved. This may be useful e.g. to efficiently track vertical movements of the data eyes due to changes in voltage and temperature.

The VREFD2 reference voltage for the DQ[7:0] and DQE inputs in NRZ mode uses the same VREFDL source (see *TABLE 39* and *FIGURE 21*), however, the programmed value is stored separate from VREFDL, is common to all DQ[7:0] and DQE inputs and applies when NRZ mode is selected by the PAM3 bit in MR0 OP8. The maximum operating frequency for NRZ mode is defined by fWCKNRZ. Switching between NRZ and PAM3 modes in MR0 OP8 will restore the corresponding programmed VREFDL, VREFDH or VREFD2 levels, and requires a time tVREFD2 to settle.

NOTE 2 I/O sub-addresses 0000_B to 1010_B select a single input for programming a per-DQ reference voltage for use in PAM3 mode; sub-address 1111_B allows programming a common reference voltage for all 11 DQ[9:0] and DQE inputs. Sub-address 1110_B also selects all DQ[9:0] and DQE inputs but is used to program a common offset to the reference voltage. Sub-address 1100_B selects the DQ[7:0] and DQE inputs to program a common VREFD2 reference voltage for use in NRZ mode.

NOTE 3 The VREFDL and VREFDH settling time tVREFD is a constant value for the device; it is referenced from the MRS command to when the 90% level of the delta between old and new VREFDL and VREFDH voltage has been reached as illustrated in *FIGURE 22*.

NOTE 4 DQE is being used in CA Training as in input, to select the WCK phases to be returned to the host. For that purpose a VREF level of nominally 0.5 x VDDQ is internally assumed. The PAM3, VREFDL and VREFDH settings are ignored while in CA Training. See *COMMAND ADDRESS BUS TRAINING* for more details.

4.1 Mode Register Assignment and Definition (cont'd)

Table 38 — Mode Registers 17 (MR17)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
VREFDH I/O Sub-Address				RFU	VREFDH Reference Voltage Level and Offset						

Function	Operand	Data	Notes
VREFDH Reference Voltage	OP[6:0]	Reference voltage level and offset for upper sub-receiver for DQ[9:0] and DQE inputs in PAM3 mode See *TABLE 39* for details	1, 3, 4
RFU	OP7	0_B: Reserved	
VREFDH I/O Sub-Address	OP[11:8]	I/O sub-address for VREFDH reference voltage level and offset 0000_B: DQ0 0001_B: DQ1 . . . 0110_B: DQ6 0111_B: DQ7 1000_B: DQ8 1001_B: DQ9 1010_B: DQE 1110_B: Common offset for DQ[9:0] and DQE 1111_B: All DQ[9:0] and DQE All others: reserved	2

NOTE 1 The reference voltages for the DQ[9:0] and DQE inputs is generated internally, and separate V_{REF} circuits are associated with each data input signal. The VREF levels itself are linear with a total range of 128 steps and a nominal step size of 3.3 mV for V_{DDQ} = 1.20 V as listed in *TABLE 39* and illustrated in *FIGURE 21*. The levels scale linearly with VDDQ. The ranges for the VREFDL and VREFDH sub-receivers overlap as illustrated in *FIGURE 21*.

The actual VREFDH voltage for PAM3 mode is determined by the sum of an offset (range 0 to 31) and a level (range 0 to 127) as shown in *FIGURE 20*. The sum of both is calculated for each input and is bounded by the lowest and highest available levels. Using an offset allows to shift the effective VREF voltage up and down for all inputs via a single MRS command while any programmed per-DQ level is being preserved. This may be useful e.g., to efficiently track vertical movements of the data eyes due to changes in voltage and temperature.

NOTE 2 I/O sub-addresses 0000_B to 1010_B select a single input for programming a per-DQ reference voltage for use in PAM3 mode; sub-address 1111_B allows programming a common reference voltage for all 11 DQ[9:0] and DQE inputs. Sub-address 1110_B also selects all DQ[9:0] and DQE inputs but is used to program a common offset to the reference voltage.

NOTE 3 The V_{REFDL} and V_{REFDH} settling time tVREFD is a constant value for the device; it is referenced from the MRS command to when the 90% level of the delta between old and new VREFDL and VREFDH voltage has been reached as illustrated in *FIGURE 22*.

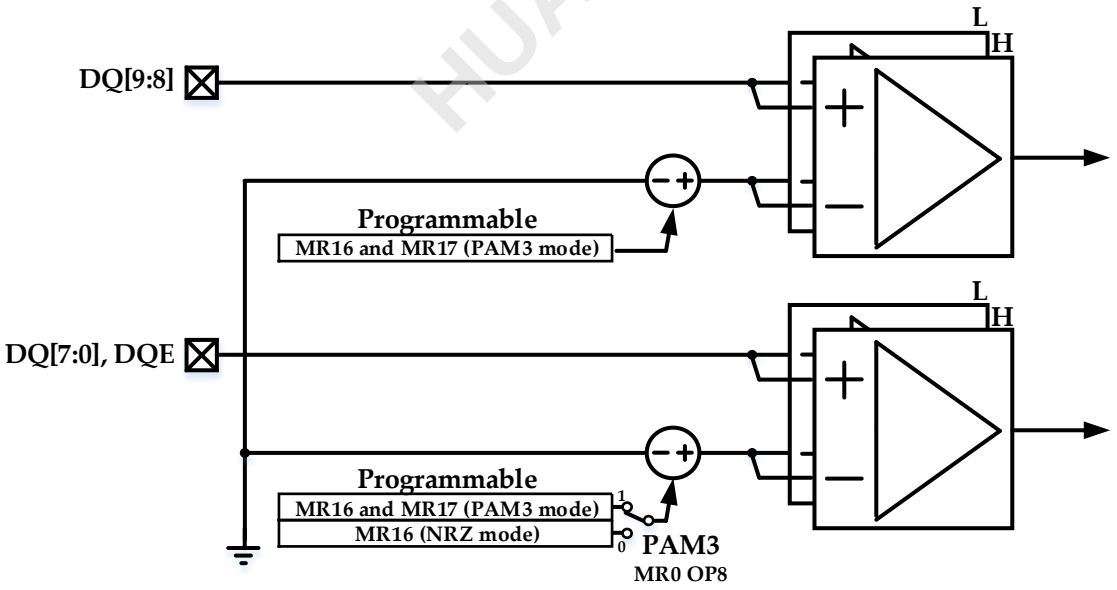
NOTE 4 DQE is being used in CA Training as in input, to select the WCK phases to be returned to the host. For that purpose a V_{REF} level of nominally 0.5 x V_{DDQ} is internally assumed. The PAM3, VREFDL and VREFDH settings are ignored while in CA Training. See *COMMAND ADDRESS BUS TRAINING* for more details.

4.1 Mode Register Assignment and Definition (cont'd)

Table 39 — VREFDL, VREFDH, and VREFD2 Levels (V_{DDQ} = 1.20 V)

VREFDL Level Code		VREFDL / VREFD2 Level			VREFDH Level Code		VREFDH Level		
MR16 OP[6:0]	Decimal	% V_{DDQ}	Divider	V_{REFDL} [V]	MR17 OP[6:0]	Decimal	% V_{DDQ}	Divider	V_{REFDH} [V]
1 1 1 1 1 1 1	127	0.7917	285 / 360	0.9500	1 1 1 1 1 1 1	127	0.9917	357 / 360	1.1900
1 1 1 1 1 1 0	126	0.7889	284 / 360	0.9467	1 1 1 1 1 1 0	126	0.9889	356 / 360	1.1867
1 1 1 1 1 0 1	125	0.7861	283 / 360	0.9433	1 1 1 1 1 0 1	125	0.9861	355 / 360	1.1833
…	…	…	…	…	…	…	…	…	…
1 0 0 0 1 0 0	68	0.6278	226 / 360	0.7533	1 0 1 0 1 1 0	86	0.8778	316 / 360	1.0533
1 0 0 0 0 1 1	**67**	**0.6250**	**225 / 360**	**0.7500**	**1 0 1 0 1 0 1**	**85**	**0.8750**	**315 / 360**	**1.0500**
1 0 0 0 0 1 0	66	0.6222	244 / 360	0.7467	1 0 1 0 1 0 0	84	0.8722	314 / 360	1.0467
…	…	…	…	…	…	…	…	…	…
0 0 0 0 0 1 0	2	0.4444	160 / 360	0.5333	0 0 0 0 0 1 0	2	0.6444	232 / 360	0.7733
0 0 0 0 0 0 1	1	0.4417	159 / 360	0.5300	0 0 0 0 0 0 1	1	0.6417	231 / 360	0.7700
0 0 0 0 0 0 0	0	0.4389	158 / 360	0.5267	0 0 0 0 0 0 0	0	0.6389	230 / 360	0.7667

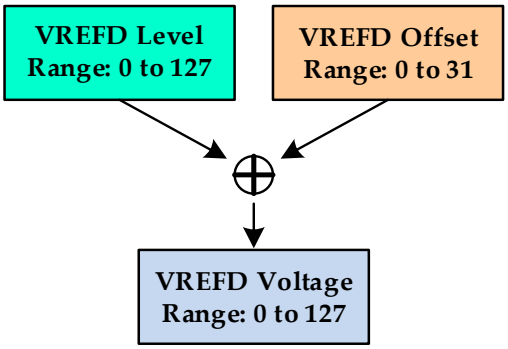
NOTE 1 The values in **bold** represent the setting for the ideal data eye center with 40 Ohm ODT and 40 Ohm driver strength in PAM3 mode.



NOTE 1 In NRZ mode the two reference voltages VREFDL and VREFDH for PAM3 signaling will be replaced by a single programmable VREFD2 reference voltage.

Figure 19 — VREFD Options

4.1 Mode Register Assignment and Definition (cont'd)



NOTE 1 The sum of the programmed offset and programmed VREFDL or VREFDH levels in PAM3 mode is bounded by the lowest and highest available levels as given in *TABLE 39*.

Figure 20 — VREFDL and VREFDH Levels and Offsets in PAM3 Mode

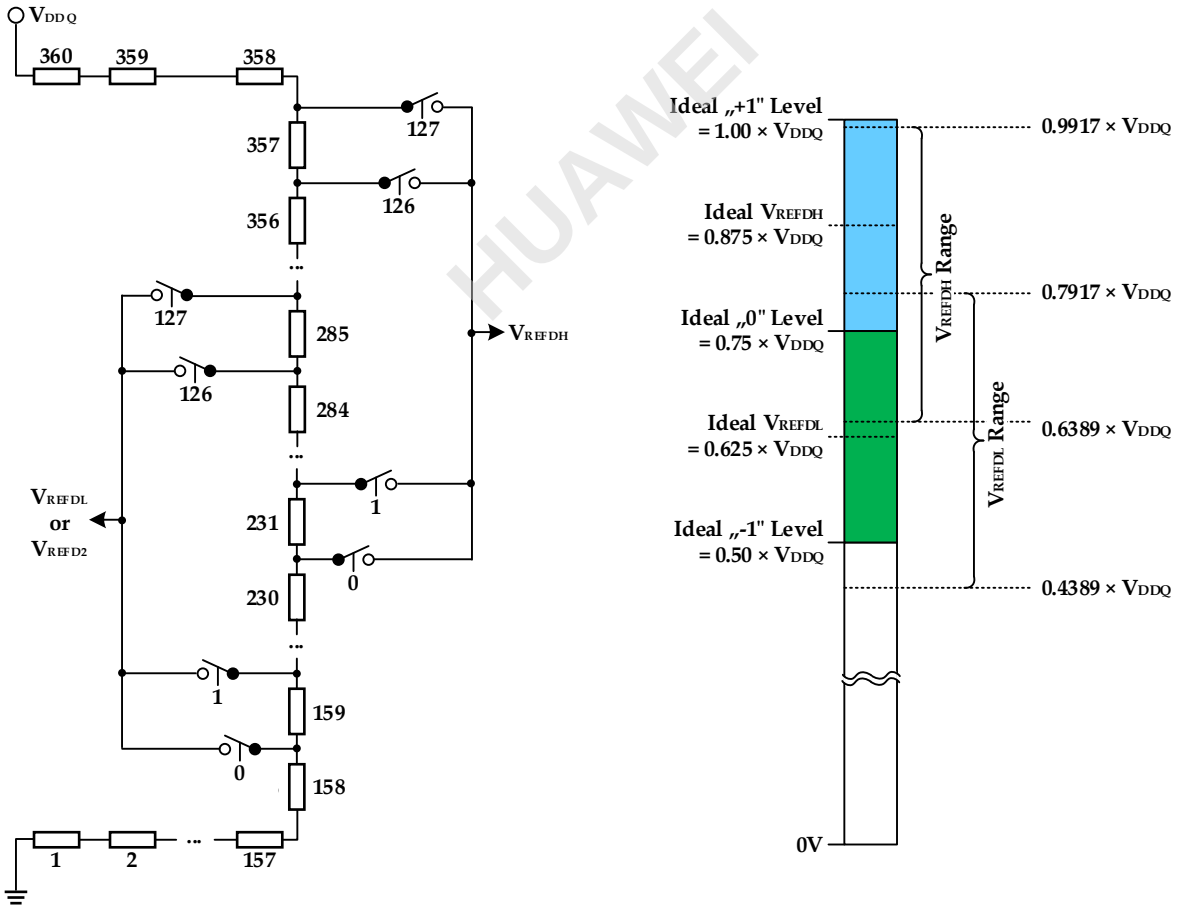
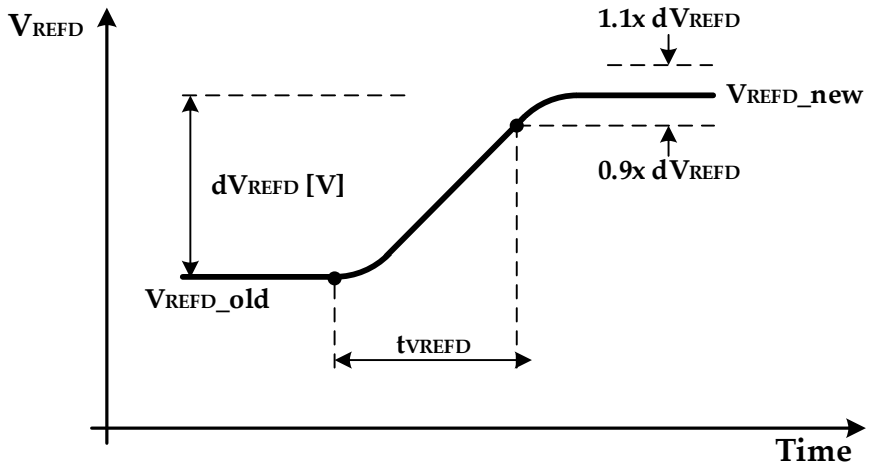


Figure 21 — VREFDL, VREFDH, and VREFD2 Circuit and Ranges

4.1 Mode Register Assignment and Definition (cont'd)



Note 1: The settling time diagram applies to both VREFDL and VREFDH.

Figure 22 — VREFD Settling Time

4.1 Mode Register Assignment and Definition (cont'd)

Table 40 — Mode Registers 18 (MR18)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
DQ/DQE CTLE I/O Sub-Address				DQ/DQE CTLE RX Sub-Address		RFU		DQ/DQE CTLE			

Function	Operand	Data	Notes
DQ/DQE CTLE	OP[3:0]	CTLE for DQ[9:0] and DQE inputs 0000_B: Off 0001_B: +1 step 0010_B: +2 steps or maximum . . . 1110_B: +14 steps or maximum 1111_B: +15 steps or maximum	1
RFU	OP[5:4]	00_B: Reserved	
DQ/DQE CTLE RX Sub-Address	OP[7:6]	Data receiver sub-address for CTLE 00_B: Upper (VREFDH) and lower (VREFDL) receivers 01_B: Upper (VREFDH) receiver only (optional) 10_B: Lower (VREFDL) receiver only (optional) 11_B: Reserved	2, 3
DQ/DQE CTLE I/O Sub-Address	OP[11:8]	I/O sub-address for CTLE 0000_B: DQ0 0001_B: DQ1 . . . 0110_B: DQ6 0111_B: DQ7 1000_B: DQ8 1001_B: DQ9 1010_B: DQE 1111_B: DQ[9:0] and DQE inputs All others: reserved	4

NOTE 1 The maximum step count supported for CTLE and the step size are vendor specific. CTLE saturates at the highest supported step count, which means that all binary encodings higher than the one that corresponds to the maximum supported step count shall result in the same maximum supported step count. With that saturation scheme, all binary encodings are always valid.

NOTE 2 With receiver sub-address 00_B a common CTLE value is being programmed for both the upper and lower data receiver. With optional receiver sub-addresses 01_B and 10_B receiver specific CTLE values can be programmed for the upper or the lower data receiver.

NOTE 3 Vendor ID3 (IRA 2) bit DQ6 identifies whether the per-receiver programmability option is supported.

NOTE 4 I/O sub-addresses 0000_B to 1010_B select a single input for programming a per-DQ CTLE; sub-address 1111_B allows programming a common CTLE for all 11 DQ and DQE inputs.

4.1 Mode Register Assignment and Definition (cont'd)

Table 41 — Mode Registers 19 (MR19)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
DQ/DQE DFE I/O Sub-Address				DQ/DQE DFE RX Sub-Address		RFU		DQ/DQE DFE			

Function	Operand	Data	Notes
DQ/DQE DFE	OP[3:0]	DFE for DQ[9:0] and DQE inputs (1-tap) 0000_B: Off 0001_B: +1 step 0010_B: +2 steps or maximum . . . 1110_B: +14 steps or maximum 1111_B: +15 steps or maximum	1
RFU	OP[5:4]	00_B: Reserved	
DQ/DQE DFE RX Sub-Address	OP[7:6]	Data receiver sub-address for DFE 00_B: Upper (VREFDH) and lower (VREFDL) receivers 01_B: Upper (VREFDH) receiver only (optional) 10_B: Lower (VREFDL) receiver only (optional) 11_B: Reserved	2, 3
DQ/DQE DFE I/O Sub-Address	OP[11:8]	I/O sub-address for data DFE 0000_B: DQ0 0001_B: DQ1 . . . 0110_B: DQ6 0111_B: DQ7 1000_B: DQ8 1001_B: DQ9 1010_B: DQE 1111_B: DQ[9:0] and DQE inputs All others: reserved	4

NOTE 1 The maximum step count supported for DFE and the step size are vendor specific. DFE saturates at the highest supported step count, which means that all binary encodings higher than the one that corresponds to the maximum supported step count shall result in the same maximum supported step count. With that saturation scheme, all binary encodings are always valid.

NOTE 2 With receiver sub-address 00_B a common DFE value is being programmed for both the upper and lower data receiver. With optional receiver sub-addresses 01_B and 10_B receiver specific DFE values can be programmed for the upper or the lower data receiver.

NOTE 3 Vendor ID3 (IRA 2) bit DQ6 identifies whether the per-receiver programmability option is supported.

NOTE 4 I/O sub-addresses 0000_B to 1010_B select a single input for programming a per-DQE DFE; sub-address 1111_B allows programming a common DFE for all 11 DQ and DQE inputs.

4.1 Mode Register Assignment and Definition (cont'd)

Table 42 — Mode Register 20 (MR20)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
Scramble Code Byte Address			RFU	Read/Write Data Scramble Code							

Function	Operand	Data	Notes
Read/Write Data Scramble Code	OP[7:0]	8-bit or 32-bit code to de-scramble array write data and to scramble array read data. The 256 bit of data per array write and read access are organized as 32 UI × 8 data lines. Default: no data scramble ($0000\ 0000_B$)	1, 2, 3
RFU	OP[8]	0_B: Reserved	
Scramble Code Byte Address	OP[11:9]	Byte address to select the byte(s) to be programmed via bits in OP[7:0] 000_B: Byte 0 001_B: Byte 1 010_B: Byte 2 011_B: Byte 3 111_B: All 4 bytes All others: reserved	1, 2, 3

NOTE 1 Vendor ID3 (IRA 2) bit DQ3 identifies whether 8-bit or 32-bit scramble code is supported.
NOTE 2 If only 8-bit scramble code is supported then OP[11:9] is ignored.
NOTE 3 Refer to the *STATIC DATA SCRAMBLER* section for details.

Table 43 — Mode Register 21 (MR21)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
DQ/DQE/RCK TX EQ I/O Sub-Address				RFU				DQ/DQE/RCK TX EQ			

Function	Operand	Data	Notes
DQ/DQE/RCK TX EQ	OP[3:0]	TX EQ for DQ[9:0], DQE and RCK_t/RCK_c outputs 0000_B: Off (default) 0001_B: +1 step or maximum 0010_B: +2 steps or maximum . . . 1110_B: +14 steps or maximum 1111_B: +15 steps or maximum	1
RFU	OP[7:4]	0000_B: Reserved	
DQ/DQE/RCK TX EQ I/O Sub-Address	OP[11:8]	I/O sub-address for TX EQ 0000_B: DQ0 0001_B: DQ1 . . . 0110_B: DQ6 0111_B: DQ7 1000_B: DQ8 1001_B: DQ9 1010_B: DQE 1011_B: RCK_t and RCK_c 1111_B: DQ[9:0], DQE and RCK_t/RCK_c outputs All others: reserved	2

NOTE 1 The maximum step count supported for TX EQ and the step size are vendor specific. TX EQ saturates at the highest supported step count, which means that all binary encodings higher than the one that corresponds to the maximum supported step count shall result in the same maximum supported step count. With that saturation scheme, all binary encodings are always valid.
NOTE 2 I/O sub-addresses 0000_B to 1010_B select a single output for programming a per-DQ TX EQ, and I/O sub-address 1011_B selects the RCK_t/RCK_c output pair; I/O sub-address 1111_B allows programming a common TX EQ for all DQ, DQE and RCK outputs.

4.1 Mode Register Assignment and Definition (cont'd)

Table 44 — Mode Register 22 (MR22)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
ECS_ ON	ECS_ RESET	ECS_ FLAG_ RESET	ECS_ LOG_ RULES	RFU							

Function	Operand	Data	Notes
RFU	OP[7:0]	0000 0000_B: Reserved	
ECS_LOG_RULES	OP8	Auto ECS error log overwrite rules 0_B: Maintain 1_B: Overwrite	1
ECS_FLAG_RESET	OP9	Auto ECS flag reset 0_B: Disable 1_B: Reset (bit is self-clearing)	1
ECS_RESET	OP10	Auto ECS reset 0_B: Disable 1_B: Reset (bit is self-clearing)	1
ECS_ON	OP11	Auto ECS 0_B: Disable (default) 1_B: Enable	1
NOTE 1 Refer to the *ERROR CHECK AND SCRUB (AUTO ECS)* section for details.			

4.1 Mode Register Assignment and Definition (cont'd)

Table 45 — Mode Register 23 (MR23)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
DT_ERR_PATTERN			FIFO_ PTR_RST	LFSR_ CNT_ MODE	CAOSC	DT_EYE_MASK		LFSR_ RST_ MODE	LFSR_ CNT_ RESET	LFSR_ TYPE	DT_ LFSR

Function	Operand	Data	Notes
DT_LFSR	OP0	Data training mode selection 0_B: Data training occurs via the Read FIFO (default) 1_B: Data training occurs via LFSR	1
LFSR_TYPE	OP1	LFSR Polynomial selection for LFSR based data training 0_B: PRBS15 1_B: PRBS11	1, 2
LFSR_CNT_RESET	OP2	Reset of the LFSR per-lane error counter 0_B: No reset 1_B: Reset (self-clearing).	1, 3
LFSR_RST_MODE	OP3	Reset mode of the LFSR per-lane error counter 0_B: Automatic reset after RDWTEC command 1_B: Manual reset via the LFSR_CNT_RESET bit	1
DT_EYE_MASK	OP[5:4]	Data receiver eye mask 00_B: No data receiver eye masking 01_B: Upper eye (VREFDH) active, lower eye (VREFDL) masked 10_B: Lower eye (VREFDL) active, upper eye (VREFDH) masked 11_B: Reserved	1
CAOSC	OP6	Command/address bus oscillator 0_B: Disable (default) 1_B: Enable	5
LFSR_CNT_MODE	OP7	Operation mode of the LFSR per-lane error counter 0_B: 12-bit error counter for the combined data eye 1_B: Two 6-bit error counters separate for lower and upper data eye	1
FIFO_PTR_RST	OP8	Reset of the Read FIFO input and output pointers 0_B: No reset (default) 1_B: Reset (self-clearing).	
DT_ERR_PATTERN	OP[11:9]	Training pattern driven on ERR output 000_B: Normal mode (default) 001_B: High-Z 010_B: Static +1 level 011_B: Static 0 level 100_B: Static -1 level 101_B: Clock pattern at CK4 rate (4 UI) 110_B: Clock pattern at half CK4 rate (4 CA UI Symbol Rate, optional) 111_B: Reserved	1, 4

NOTE 1 Refer to the *DATA TRAINING* section for details.

NOTE 2 The bit is ignored when training via the Read FIFO is selected in OP0.

NOTE 3 The bit is self-clearing meaning that it returns to the value 0_B after the reset has been issued.

NOTE 4 The CA parity and WRCRC error signaling on the ERR output is suspended when the DT_ERR_PATTERN bits are set to any value other than the default 000_B.

NOTE 5 Refer to the *COMMAND ADDRESS OSCILLATOR (CAOSC)* section for details.

4.1 Mode Register Assignment and Definition (cont'd)

Table 46 — Mode Registers 24 (MR24)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
LFSR Sub-Address		LFSR Seed Bits [9:0]									

Function	Operand	Data	Notes
LFSR Seed Bits [9:0]	OP[9:0]	Lower 10 bits of a 15-bit starting value for LFSR based data training. The LFSR sub-address field in OP[11:10] selects 1 out of 3 LFSRs.	1, 2
LFSR Sub-Address	OP[11:10]	LFSR sub-address for programming a per-LFSR seed. 00_B: LFSR for Group 0 signals (DQ[3:0]) 01_B: LFSR for Group 1 signals (DQ[7:4]) 10_B: LFSR for Group 2 signals (DQ[9:8] and DQE) 11_B: Reserved	1, 2

NOTE 1 Refer to the *DATA TRAINING* section for details on LFSR based data training.

NOTE 2 The LFSR sub-address selects one of 3 LFSRs. A total of 6 MRS commands to MR24 and MR25 are required to program the seed values for LFSR1, LFSR2 and LFSR3.

Table 47 — Mode Registers 25 (MR25)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
LFSR Sub-Address		RFU					LFSR Seed Bits [14:10]				

Function	Operand	Data	Notes
LFSR Seed Bits [14:10]	OP[4:0]	Upper 5 bits of a 15-bit starting value for LFSR based data training. Bits [14:11] shall be programmed to 0000_B when PRBS11 is selected by the LFSR_TYPE bit in MR23. The LFSR sub-address field in OP[11:10] selects 1 out of 3 LFSRs.	1, 2
RFU	OP[9:5]	00000_B: Reserved	
LFSR Sub-Address	OP[11:10]	LFSR sub-address for programming a per-LFSR seed. 00_B: LFSR for Group 0 signals (DQ[3:0]) 01_B: LFSR for Group 1 signals (DQ[7:4]) 10_B: LFSR for Group 2 signals (DQ[9:8] and DQE) 11_B: Reserved	1, 2

NOTE 1 Refer to the *DATA TRAINING* section for details on LFSR based data training.

NOTE 2 The LFSR sub-address selects one of 3 LFSRs. A total of 6 MRS commands to MR24 and MR25 are required to program the seed values for LFSR1, LFSR2, and LFSR3.

4.1 Mode Register Assignment and Definition (cont'd)

Table 48 — Mode Register 26 (MR26)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
I/O Sub-Address				RFU				DT_ LANE_ MASK	LFSR_ INV	LFSR_SHIFT_SEL	

Function	Operand	Data	Notes
LFSR_SHIFT_SEL	OP[1:0]	Symbol offset for LFSR generated training data 00_B: 0 symbols (no offset) 01_B: 4 symbols 10_B: 8 symbols 11_B: 12 symbols	1, 2
LFSR_INV	OP2	Symbol inversion for LFSR generated training data 0_B: Not inverted 1_B: Inverted	1, 2
DT_LANE_MASK	OP3	Lane masking for training data 0_B: No masking (Default) 1_B: Lane is masked	1, 2
RFU	OP[7:4]	0000_B: Reserved	
I/O Sub-Address	OP[11:8]	I/O sub-address for LFSR data training operation 0000_B: DQ0 0001_B: DQ1 . . . 0110_B: DQ6 0111_B: DQ7 1000_B: DQ8 1001_B: DQ9 1010_B: DQE All others: reserved	1, 2

NOTE 1 Refer to the *DATA TRAINING* section for details.

NOTE 2 I/O sub-addresses 0000_B to 1010_B select one data I/O for programming the LFSR_SHIFT_SEL, LFSR_INV and DT_MASK features using in data training.

4.1 Mode Register Assignment and Definition (cont'd)

Table 49 — Mode Register 27 (MR27)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU				ERR Pullup Driver Offset				ERR Pulldown Driver Offset			

Function	Operand	Data	Notes
ERR Pulldown Driver Offset	OP[3:0]	Pulldown driver offset for ERR signal (optional) 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	1, 2
ERR Pullup Driver Offset	OP[7:4]	Pullup driver offset for ERR signal (optional) 0000_B: 0 (no offset; default) 0001_B: +1 0010_B: +2 . . . 0111_B: +7 1000_B: -8 1001_B: -7 . . . 1111_B: -1	1, 2
RFU	OP[11:8]	0000_B: Reserved	

NOTE 1 The offset impedance step values may be non-linear and will vary across DRAM vendors and across PVT. With negative offset steps the driver strength will be decreased, and RON will be increased; with positive offset steps the driver strength will be increased, and RON will be decreased.

NOTE 2 Vendor ID3 (IRA 2) bit DQ7 identifies whether the optional pullup and pulldown driver offset capability for the ERR output is supported or not.

Table 50 — Mode Register 28 (MR28)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU											

Function	Operand	Data	Notes
RFU	OP[11:0]	0000 0000 0000_B: Reserved	1

4.1 Mode Register Assignment and Definition (cont'd)

Table 51 — Mode Register 29 (MR29)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU							DQ Map Mode				

Function	Operand	Data	Notes
DQ Map Mode	OP[4:0]	Identification of the logical to physical DQ mapping for DQ[9:0] 00000_B: Disabled (default) 10000_B: DQ0 10001_B: DQ1 . . . 11000_B: DQ8 11001_B: DQ9 All others: reserved	1
RFU	OP[11:5]	$000\ 0000_B$: Reserved	
NOTE 1: DQ map mode is disabled by default and enabled by setting bit OP4 to 1_B. Refer to the *DQ MAP MODE* section for details.			

Table 52 — Mode Register 30 (MR30)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
RFU			ECC_ 2CH	ECC_PAR_SEL						ECC_ VEC	ECC_ TM

Function	Operand	Data	Notes
ECC_TM	OP0	ECC engine test mode 0_B: Disable (default) 1_B: Enable	1
ECC_VEC	OP1	ECC engine test vector selection 0_B: Code word 0 (CW0) 1_B: Code word 1 (CW1)	1, 2
ECC_PAR_SEL	OP[7:2]	ECC parity bit error injection $00\ XXXX_B$: No error injection $01\ 0000_B$: Error injected on parity bit 0 $01\ 0001_B$: Error injected on parity bit 1 … $01\ 1110_B$: Error injected on parity bit 14 $01\ 1111_B$: Error injected on parity bit 15 $1X\ XXXX_B$: Reserved	1, 2
ECC_2CH (optional)	OP8	Additional ECC engine test at 2 Channel mode 0_B: First pass ECC engine (default) 1_B: Second pass ECC engine	1, 3
RFU	OP[11:9]	0000_B: Reserved	

NOTE 1 Refer to the *ECC ENGINE TEST MODE* section for details.
NOTE 2 The ECC_VEC and ECC_PAR_SEL bits are only evaluated when ECC engine test mode is enabled by the ECC_TM bit in OP0.
NOTE 3 If IRA3 DQ4 = 0B, the user is required to test both MR30 OP8 = 0 and 1 at 2 channel mode.

4.1 Mode Register Assignment and Definition (cont'd)

Table 53 — Mode Register 31 (MR31)

OP11	OP10	OP9	OP8	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
hPPR	RFU			PPR Guard Key							

Function	Operand	Data	Notes
PPR Guard Key	OP[7:0]	8-bit Guard Key for Post Package Repair Seq1: $1100\ 1111_B$ Seq2: $0111\ 0011_B$ Seq3: $1011\ 1011_B$ Seq4: $0011\ 1011_B$	1
RFU	OP[10:8]	000_B: Reserved	
hPPR	OP11	Hard Post Package Repair 0_B: Disable (ignore PPR guard key; default) 1_B: Enable	1
NOTE 1 Refer to the *HARD POST PACKAGE REPAIR (HPPR)* section for details.			

4.2 Mode Register Default Settings

Table 54 — Mode Register Default Settings

Item	Mode Register	Default Setting	Description
CABI	MR0 OP0	0_B	Enabled
RDCRC	MR0 OP3	0_B	Disabled
WRCRC	MR0 OP4	0_B	Disabled
SLX2CAT	MR0 OP5	0_B	Enabled
SRSLX2CAT	MR0 OP6	0_B	Enabled
TM	MR0 OP7	0_B	Normal mode
PAM3	MR0 OP8	0_B	NRZ mode
FD_FLAG	MR0 OP11	0_B	Enabled
DQ/DQE/RCK Driver Strength	MR5 OP[1:0]	00_B	40 Ohm, auto-calibrated
DQ/DQE Data Termination	MR5 OP[3:2]	00_B	Disabled
ERR Driver Strength	MR5 OP5	0_B	40 Ohm, auto-calibrated
CAL_UPD	MR5 OP[8:7]	00_B	Enabled
SEV2ERR	MR5 OP9	0_B	Disabled
DQE_HZ	MR5 OP10	0_B	Disabled
DQ/DQE/RCK PD Offset (Leg 1)	MR6 OP[3:0]	0000_B	Disabled (no offset)
DQ/DQE/RCK PD Offset (Leg 2)	MR6 OP[7:4]	0000_B	Disabled (no offset)
DQ/DQE/RCK PU Offset (Leg 1)	MR7 OP[3:0]	0000_B	Disabled (no offset)

Table 54 — Mode Register Default Settings (cont'd)

Item	Mode Register	Default Setting	Description
DQ/DQE/RCK PU Offset (Leg 2)	MR7 OP[7:4]	0000_B	Disabled (no offset)
DQ/DQE Data Termination Offset	MR7 OP[11:8]	0000_B	Disabled (no offset)
DRFM	MR8 OP2	0_B	Disabled
DCC	MR8 OP[6:5]	00_B	Disabled
Self Refresh	MR8 OP[10:9]	00_B	Fixed self refresh interval
RCKMODE	MR9 OP[1:0]	00_B	Disabled
VDD Range	MR12 OP[1:0]	00_B	Default level
VDDQ Range	MR12 OP[3:2]	00_B	Default level
WCK Frequency	MR12 OP[8:4]	00000_B	For $f_{WCK} \leq f_{WCKNRZ}$
CA Termination	MR13 OP[2:0]	000_B	Value latched on CA1 input at exit from reset state
CA Termination Offset	MR13 OP[5:3]	000_B	Disabled (no offset)
WCK Termination	MR13 OP[8:6]	000_B	Value latched on CA2 input at exit from reset state
WCK Termination Offset	MR13 OP[11:9]	000_B	Disabled (no offset)
VREFCA	MR14 OP[5:0]	0101111_B	$V_{REFCA} = 0.725 \times V_{DDQ}$
DFECA	MR14 OP[10:7]	0000_B	Disabled (no DFE)
Half VREFCA	MR14 OP11	0_B	Disabled (default V_{REFCA} level)
CAPAR	MR15 OP0	0_B	Disabled
CAPARBLK	MR15 OP1	0_B	Disabled
CSP_FB	MR15 OP2	0_B	Disabled
CAPARBLK_LAT Control	MR15 OP6	0_B	Implicit CAPARBLK_LAT
Read/Write Data Scramble Code	MR20 OP[7:0]	$0000\ 0000_B$	Disabled (no scramble code)
DQ/DQE/RCK TX EQ	MR21 OP[3:0]	0000_B	Disabled
ECS_ON	MR22 OP11	0_B	Disabled
DT_LFSR	MR23 OP0	0_B	Data training via the Read FIFO
CAOSC	MR23 OP6	0_B	Disabled
DT_ERR_PATTERN	MR23 OP[11:9]	000_B	Disabled
DT_LANE_MASK	MR26 OP3	0_B	Disabled (no lane masking)
ERR Pulldown Driver Offset	MR27 OP[3:0]	0000_B	Disabled (no offset)
ERR Pullup Driver Offset	MR27 OP[7:4]	0000_B	Disabled (no offset)
DQ Map Mode	MR29 OP[4:0]	00000_B	Disabled
ECC_TM	MR30 OP0	0_B	Disabled
hPPR	MR31 OP11	0_B	Disabled

5 Training

5.1 Interface Training Sequence

Due to the high data rates of GDDR7, it is recommended that the Command Address and Data bus interfaces be trained to operate with the optimal timings. GDDR7 SGRAM has features defined which allow for complete and efficient training of the I/O interface without the use of the device's memory array. The interface trainings are required for normal DRAM functionality unless deemed optional by the DRAM vendor or unless running in lower frequency modes as described in the low frequency section. Interface timings will only be guaranteed after all required trainings have been executed.

A recommended order of training sequences has been chosen based on the following criteria:

The Command Address (CA) training must be done first to allow full access to the Mode Registers. CA input timing shall function without training as long as tWCK2CA offset timing with sufficient eye window margin are met and CSP has been successfully issued.

CSP must then be issued before CA commands are valid.

ERR training may be optionally completed. It is suggested to train ERR signal before READ and WRITE trainings such that CA Parity information can be properly received by the host during and after these trainings.

READ training should be done before WRITE training because optimal WRITE training depends on correct READ data.

As part of WRITE training, the host has the ability to find the data-eye optimal position in the horizontal direction and also the ability to adjust VREFDH/L per DQ/DQE using the mode registers to find the data-eye optimal position value. Additionally, the host may adjust CTLE, and DFE levels per DQ/DQE or per channel using the mode registers to find the data-eye optimal equalization values.

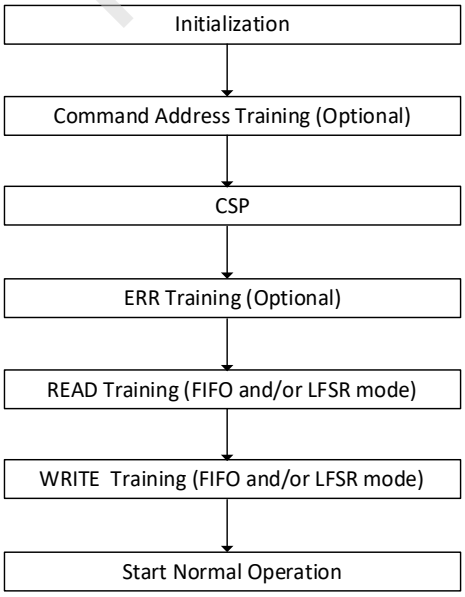


Figure 23 — Interface Training Sequence

5.2 Command Address Bus Training

GDDR7 SGRAMs provide a means for Command Address (CA) bus interface training. The host may use the CA training mode to improve the timing margins on the CA bus, and to discover which WCK positive edge currently corresponds to the internal CK4 positive edge.

CA training mode is entered in one of the following ways:

- Using the CATE command
- When Command/address parity blocking is enabled (MR0 OP2) and a CA parity error occurs
- When exiting Sleep while the SLX2CAT mode register flag is set
- When exiting Self-Refresh Sleep while the SRSLX2CAT mode register flag is set
- Following RESET during power up initialization or initialization with stable power

CA training mode uses an internal bridge between the device's CA inputs and DQ[9:0] outputs. Once the device is placed into CA training mode, the only command that can be interpreted is the CA training exit command. After CA training mode has been entered and tCATE or tSLX_CAT has been met, the CA values registered on every internal CK4_0/CK4_1 or CK4_2/CK4_3 will be transmitted concurrently to the host on DQ[9:0] tADR time later. The order of returned samples in the sample window along with RCK_t rising edge always starts with CK4_0 as the shown in *FIGURE 24*. The host is then expected to compare the DQ pattern received to the expected value and to adjust the CA transmit timing accordingly. The procedure may be repeated using different CA patterns and interface timings. *FIGURE 24* and *FIGURE 25* show the synchronous entry into CA Training mode using the CATE command.

After CA training entry the device will, within tCATE2RCK or tSLXCAT2RCK, asynchronously start driving RCK continuously at nCK4 rate. CA training mode ignores the RCKMODE, RCKTYPE, and RCKLEVEL mode register fields, and instead the device always drives a differential full-swing RCK. The device will asynchronously stop driving RCK at nCK4 rate within tCATX after it receives a CATX command. When the device is in RCKSTRT/RCKSTOP or RD/RCKSTOP mode the host must stop RCK prior to a CATE command. When the device is in RCKSTRT/RCKSTOP or RD/RCKSTOP mode, RCK will revert to stopped during tCATX. When the device is in RCK always-on mode RCK will revert to on during tRCK_AON_CATX, and the transition between nCK4 and full rate RCK is asynchronous. Before the CATX command, the host must issue 4 consecutive WCK cycles of CA[4:0] = H. CATX command requires the host to hold CA[4:0] low, with proper tAS and tAH, for 16 consecutive WCK cycles. After the CATX command, the host must drive CA[4:0] = H (NOP2) for the duration of tCATX before issuing a CSP command. *FIGURE 27* shows the sequence for CA training exit.

While in CA training mode the device drives captured CA values to the host on DQ[9:0]. DQE is driven by the host to the device and is used to select which clock phases the device is sampling. The DQE input ignores the PAM3 bit in MR0, Data ODT bit in MR5, VREDFL/H, DFE, and CTLE bits in MR16-19 while in CA training mode. No DQE related programming is necessary, and DQE is always an unterminated NRZ signal from host to device in both PAM3 and NRZ data modes, with a VREF of 0.5*VDDQ. When the device receives DQE=H, the device samples CA on CK4_0 and CK4_1. When the device receives DQE=L, the device samples CA on CK4_2 and CK4_3. Host must hold CA[4:0]=H for tDQE_PRE nCK4 cycle before DQE changes and for tDQE_POST nCK4 cycles after DQE changes. *FIGURE 26* shows the sequence for changing DQE to select a different pair of CK4 phases.

It should be noted that the host must use caution when selecting CA training patterns. Certain training patterns may cause an unintentional CATX while the host is sweeping phase near a CA UI boundary. Any CA training pattern which does not meet the following criteria is at risk of an unintentional CATX:

- At least one of the CA[4:0] lanes must be held high for at least two consecutive WCK cycles within every 16 WCK cycle window after tCATE and before issuing a CATX command

Additionally, CA-to-CA system skew must be carefully considered by the host when selecting patterns which satisfy the above criteria.

5.2 Command Address Bus Training (cont'd)

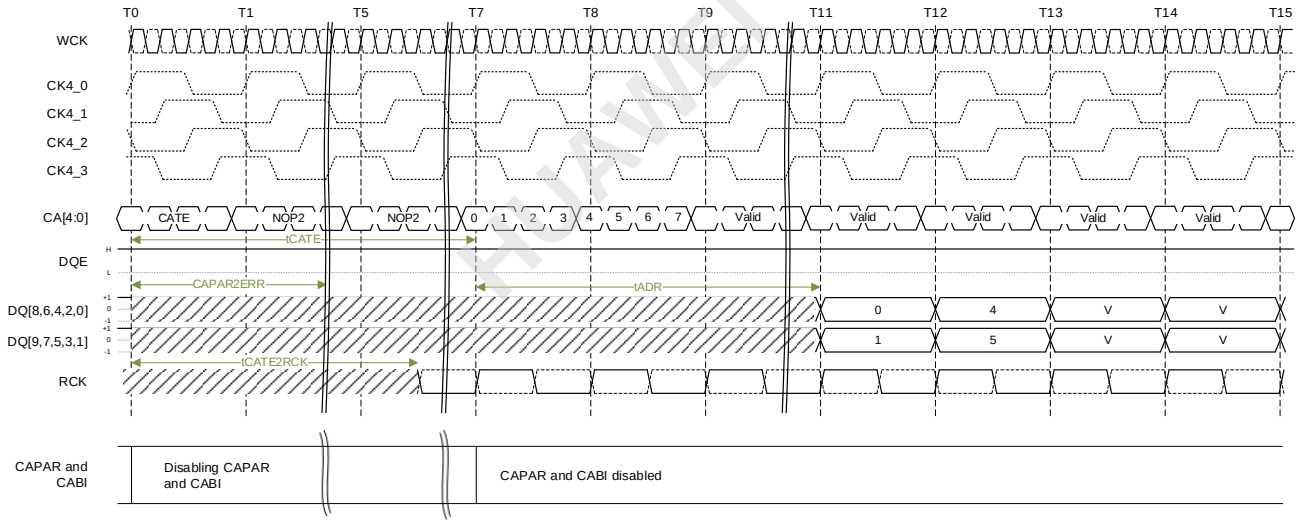
In NRZ mode a CA input of H results in an NRZ DQ output of H, and a CA input of L results in an NRZ DQ output of L.

In PAM3 mode a CA input of H results in a PAM3 DQ output of +1, and a CA input of L results in a PAM3 DQ output of -1.

The mapping of CA[4:0] input, DQE input, and DQ[9:0] output is found in *TABLE 55*.

Table 55 — CA Capture to DQ Return Mapping

DQE (input) Status	Sampled Phase	CA0	CA1	CA2	CA3	CA4
DQE = H	CK4_0	DQ0	DQ2	DQ4	DQ6	DQ8
DQE = H	CK4_1	DQ1	DQ3	DQ5	DQ7	DQ9
DQE = L	CK4_2	DQ0	DQ2	DQ4	DQ6	DQ8
DQE = L	CK4_3	DQ1	DQ3	DQ5	DQ7	DQ9

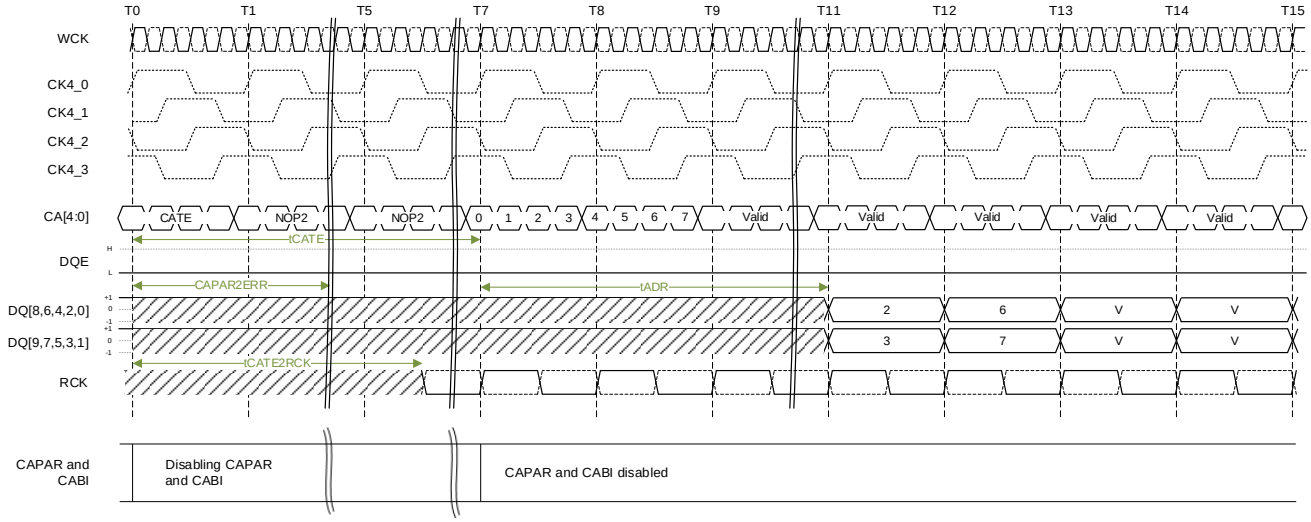


NOTES:

1. tCATE in this example is 7 nCK4.
2. tADR in this example is 4 tCK4 and 0ns analog output delay for illustration purposes.
3. tCATE2RCK in this example is less than tCATE. It may be more than tCATE, but must be less than tCATE+tADR-1 nCK4 to ensure RCK is toggling prior to the first DQ output.
4. DQ[9:0] in this example are PAM3 signals at +1/-1 levels. In NRZ mode they would be NRZ signals at H/L levels.
5. RCK state is unknown during tCATE2RCK. First pulse may be incomplete.
6. In this example RCK and DQ are shown with 0ns analog output delay.
7. The states of CAPAR and CABI calculations are applicable when the functions are enabled in the respective mode registers.
8. Valid = H or L but not floating.
9. tADR specified in ns shall be the same for all CA -->DQ pairs across all active channels.
10. tCAT_DQ2DQ shall be limited such that DQ feedback available on a given RCK edge corresponds to the phases of the CA pattern captured by edges of the same CK4 quartet (defined as a set of CK4_0 /1/2/3 clock phases)

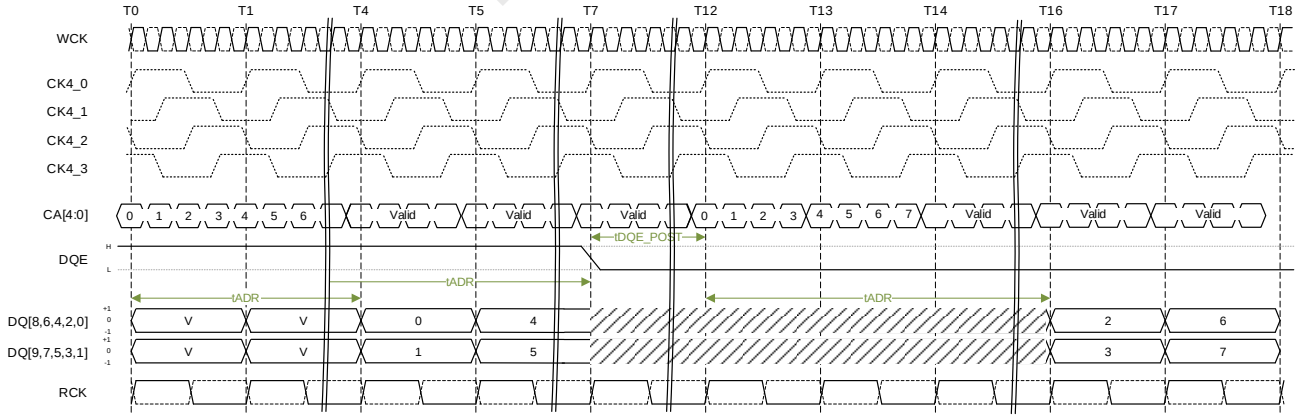
Figure 24 — CA Training Entry and CK4_0 / CK4_1 Sampling

5.2 Command Address Bus Training (cont'd)



NOTES:
1. tCATE in this example is 7 nCK4.
2. tADR in this example is 4 tCK4 and 0 ns analog output delay for illustration purposes.
3. tCATE2RCK in this example is less than tCATE. It may be more than tCATE, but must be less than tCATE+tADR-1 nCK4 to ensure RCK is toggling prior to the first DQ output.
4. DQ[9:0] in this example are PAM3 signals at +1/-1 levels. In NRZ mode they would be NRZ signals at H/L levels.
5. RCK state is unknown during tCATE2RCK. First pulse may be incomplete.
6. In this example RCK and DQ are shown with 0ns analog output delay.
7. The states of CAPAR and CABI calculations are applicable when the functions are enabled in the respective mode registers.
8. Valid = H or L but not floating.

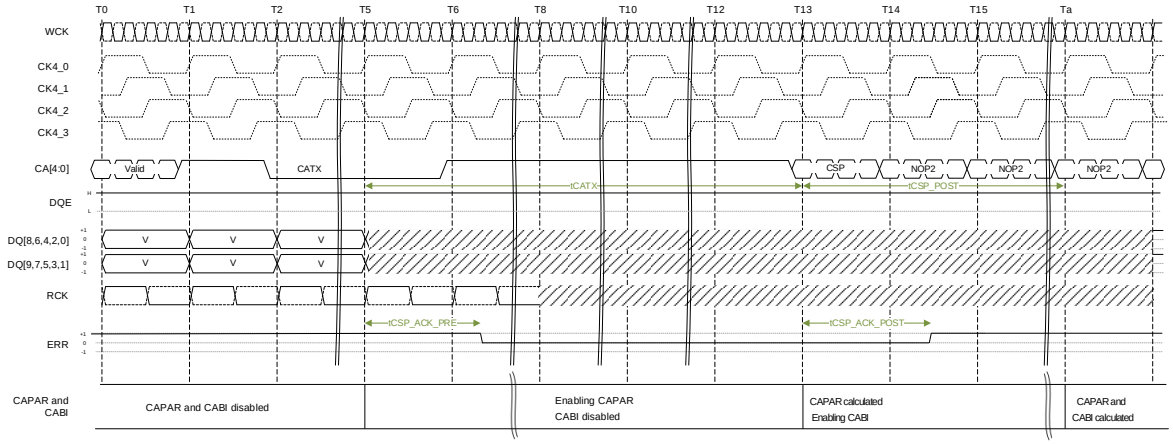
Figure 25 — CA Training and CK4_2 / CK4_3 Sampling



NOTES:
1. tADR in this example is 4 tCK4 and 0 ns analog output delay for illustration purposes.
2. tDQE_PRE in this example is 3 nCK4.
3. tDQE_POST in this example is 5 nCK4.
4. CA[4:0] Valid during tDQE_PRE and tDQE_POST must not match the CATX command.
5. In this example RCK and DQ are shown with 0 ns analog output delay.
6. Valid = H or L but not floating.

Figure 26 — DQE Change to Select Alternate CK4 Phases

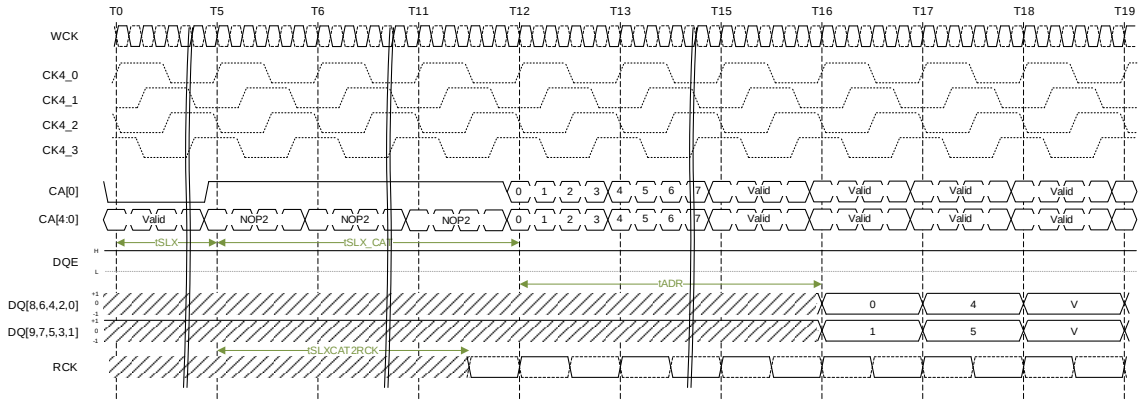
5.2 Command Address Bus Training (cont'd)



NOTES:
1. CATX command is 16 WCK of CA[4:0]=L. CATX command is shown aligned to CK4 for illustration purposes.
2. tCATX in this example is 6 nCK4.
3. tCSP_PRE in this example is 3 nCK4.
4. During tCATX the device will stop driving CK4 rate RCK. The state of RCK depends on the RCKMODE setting in MR9. Last pulse of RCK at CK4 rate may be incomplete. If RCKMODE set to RCK always on mode, then the RCK will start toggling at full rate during tRCK_AON_CATX.
5. During tCATX the device will stop driving DQ. Stopping at T5 is shown as an example of the earliest for illustration purposes.
6. In this example RCK and DQ are shown with 0ns analog output delay.
7. If the optional CSP Feedback feature is supported and enabled, the ERR signal will transition to "0" level after tCSP_ACK_PRE and will transition to "+1" level tCSP_ACK_POST after CSP decoded successfully. If not enabled, the ERR signal continues to be driven at the "+1" level. See the *COMMAND START POINT (CSP)* section for more details.
8. Valid = H or L but not floating.

Figure 27 — CA Training Exit

The device will enter CA training mode following an SLX command if woken from Sleep when the SLX2CAT mode register field is set or if woken from Self-Refresh Sleep when the SRSLX2CAT mode register field is set. *FIGURE 28* shows the transition from either Sleep mode to CA training mode.

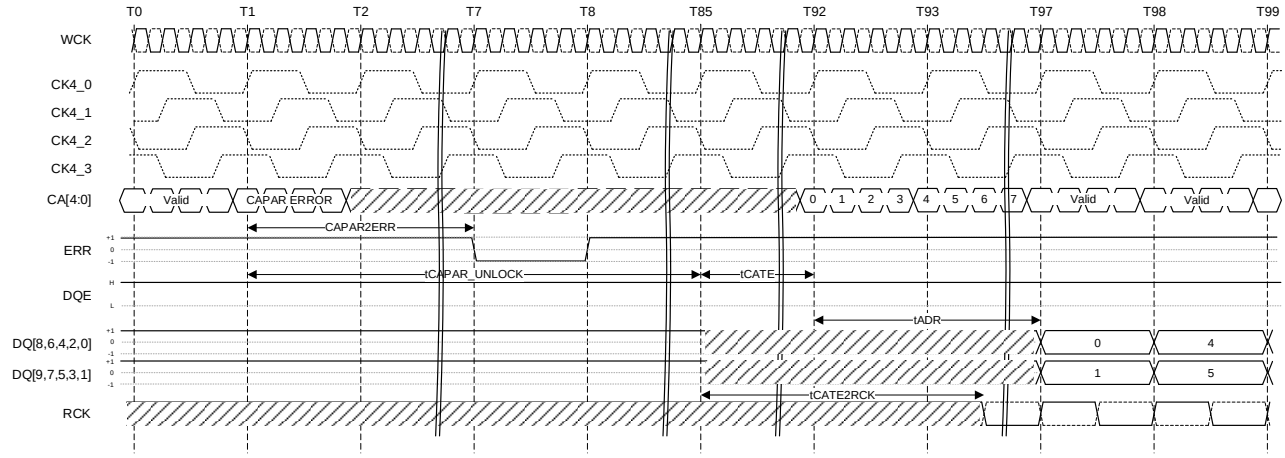


NOTES:
1. tSLX in this example is 5 nCK4.
2. tSLX_CAT in this example is 7 nCK4.
3. tADR in this example is 4 tCK4 and 0ns analog output delay for illustration purposes.
4. tSLXCAT2RCK in this example is less than tSLX_CAT. It may be more than tSLX_CAT, but must be less than tSLX_CAT+tADR-1 nCK4 to ensure RCK is toggling prior to the first DQ output.
5. DQ[9:0] in this example are PAM3 signals at +1/-1 levels. In NRZ mode they would be NRZ signals at H/L levels.
6. RCK state is unknown during tSLXCAT2RCK. First pulse may be incomplete.
7. In this example RCK and DQ are shown with 0ns analog output delay.
8. Valid = H or L but not floating.

Figure 28 — SLX to CA Training Entry

5.2 Command Address Bus Training (cont'd)

When CA Parity and CA Parity Command Blocking are enabled, the device will enter CA training mode following a detected error on the CA bus. *FIGURE 29* shows the transition from a CA parity error to CA Training mode. For more information on parity and command blocking refer to the CA parity section.



NOTES:
1. CAPAR2ERR in this example is 6 nCK4.
2. tCAPAR_UNLOCK in this example is 84 nCK4.
2. tCATE in this example is 7 nCK4.
3. tADR in this example is 4 tCK4 and 0 ns analog output delay for illustration purposes.
4. tCATE2RCK in this example is more than tCATE. It may be less than tCATE, but must be less than tCATE+tADR-1 nCK4 to ensure RCK is toggling prior to the first DQ output.
5. DQ[9:0] in this example are PAM3 signals at +1/-1 levels. In NRZ mode they would be NRZ signals at H/L levels.
6. RCK may be toggling at WCK rate prior to T85 in this example. RCK state is unknown during tCATE2RCK. First nCK4 pulse may be incomplete.
7. In this example RCK and DQ are shown with 0ns analog output delay.
8. Valid = H or L but not floating.

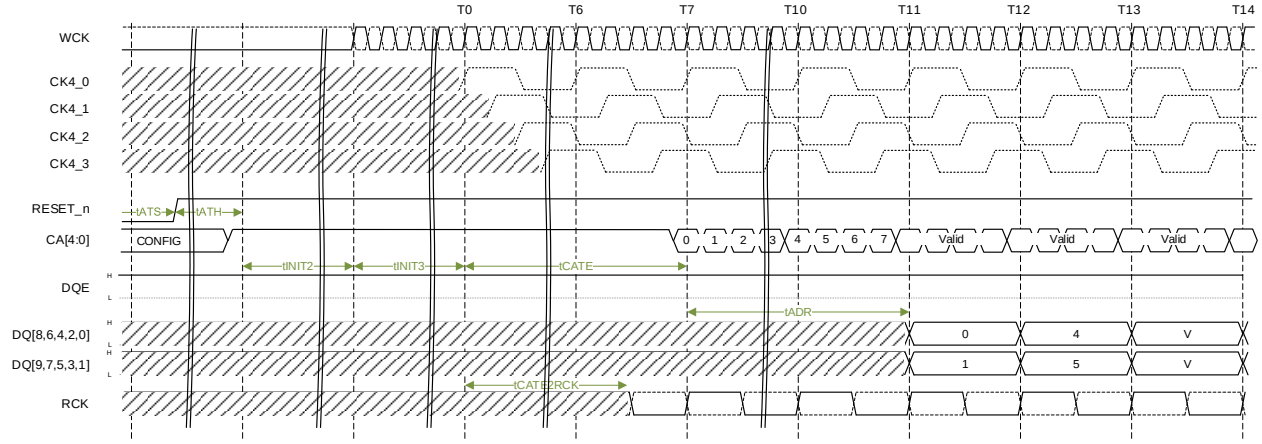
Figure 29 — CA Parity Error to CA Training Entry

Following deassertion of RESET_n the device will enter CA Training mode. Several mode register fields have default values to support CA Training immediately after reset, before any commands may have been issued. These include:

- MR0 PAM3: DQs default to 0, NRZ mode.
- MR5 Driver Strength: DQ and RCK drive strength defaults to 00, 40 Ohm auto calibrated.
- MR6 PD Offset Leg 1 and Leg 2, and MR7 PU Offset Leg 1 and Leg 2: Driver Offsets default to 0000, no offset.
- MR13 CA Termination: Termination for CA inputs defaults to 000, value latched at exit of reset.
- MR13 WCK Termination: Termination for WCK inputs defaults to 000, value latched at exit of reset.
- MR13 CA Termination Offset: CA offset defaults to 000, no offset.
- MR13 WCK Termination Offset: WCK offset defaults to 000, no offset.
- MR14 VREFCA: CA reference voltage defaults to 0101111, 0.725*VDDQ.
- MR14 DFECA: DFE for CA defaults to 0000, no DFE.
- MR14 Half VREFCA: Defaults to 0, use programmed reference voltage VREFCA.

5.2 Command Address Bus Training (cont'd)

FIGURE 30 shows the transition from RESET to CA Training mode.



NOTES:
1. tCATE in this example is 7 nCK4.
2. tADR in this example is 4 tCK4 and 0ns analog output delay for illustration purposes.
3. tCATE2RCK in this example is less than tCATE. It may be more than tCATE but must be less than tCATE+tADR-1 nCK4 to ensure RCK is toggling prior to the first DQ output.
4. DQ[9:0] are NRZ signals at H/L levels. Reset default for DQ mode is NRZ.
5. RCK state is unknown during tCATE2RCK. First pulse may be incomplete.
6. In this example RCK and DQ are shown with 0 ns analog output delay.
7. Valid = H or L but not floating.

Figure 30 — RESET Deassertion to CA Training Entry

Table 56 — AC Timings in Command Address Training Mode

Parameter	Symbol	Min	Max	Unit
CA training entry to first sampling	t_{CATE}	-		ns
CA training entry to RCK on delay	$t_{CATE2RCK}$	-	$t_{CATE} + t_{ADR} - t_{CK4}$	ns
Sleep mode exit when SLX2CAT = 1_B or Self Refresh Sleep mode exit when SRSLX2CAT = 1_B to CA training entry delay	t_{SLX_CAT}	-		ns
Sleep mode exit when SLX2CAT = 1_B or Self Refresh Sleep mode exit when SRSLX2CAT = 1_B to RCK on delay	$t_{SLXCAT2RCK}$	-	t_{SLX_CAT} (max) + t_{ADR} (max) - $t_{CK}4$	ns
CA sampling from CK4_0 (int) to data out delay	t_{ADR}			ns
CATX to CSP command delay	t_{CATX}	-		ns
CA[4:0] high following DQE transition	t_{DQE_POST}		-	ns
RCK rising to DQ valid window start in CA Training	t_{CAT_RCK2DQ}			ps
DQ to DQ output skew in CA Training	t_{CAT_DQ2DQ}			ps

5.3 ERR Training

GDDR7 SGRAMs provide a means for detecting CA Parity errors and Write CRC errors and returning this information back to the host on the ERR signal. The ERR signal will transfer error information at CK4 symbol rate using PAM3 signaling. ERR signal transfers error information using PAM3 signaling in both PAM3 mode and NRZ mode. Please refer to the *ERR SIGNAL* section for further details.

To enable robust ERR signaling the following ERR training related features are introduced which enable the host to train host ERR receiver voltage levels:

- DC patterns (+1, 0, -1)
- Clock patterns

DT_ERR_PATTERN (MR23 OP[11:9]) can be configured to one of the following modes which determine ERR signal behavior as illustrated in *FIGURE 31*:

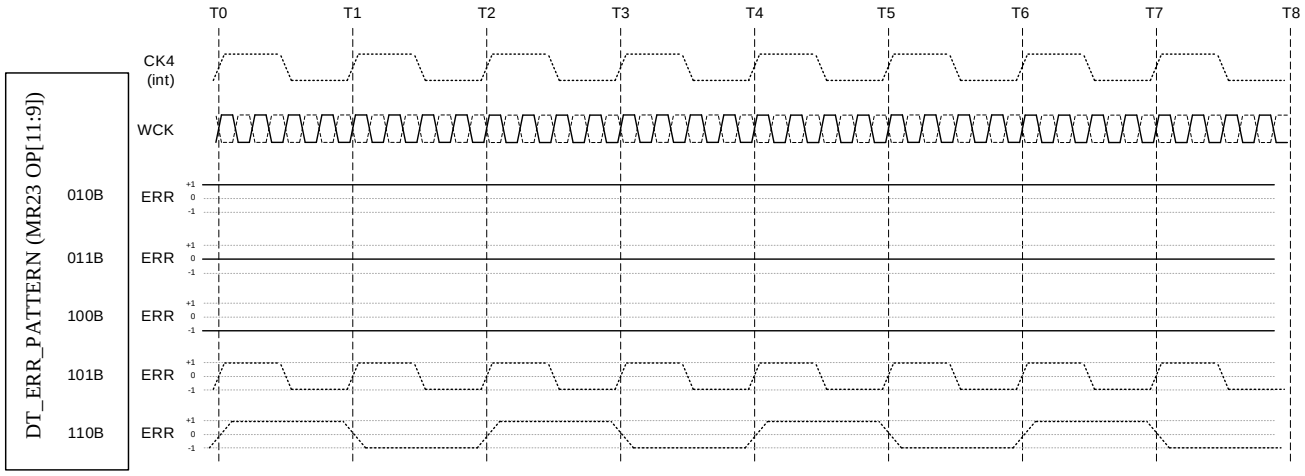
- 000 = normal mode
- 001 = off (High-Z)
- 010 = drive +1 (See note 1)
- 011 = drive 0 (See note 1)
- 100 = drive -1 (See note 1)
- 101 = drive clock pattern. CK4 rate, +1/-1. (See note 1)
- 110 = drive clock pattern. Half CK4 rate, +1/-1 (See note 1, and note 2)
- 111 = RFU

NOTES:

1. ERR training is only valid in Bank Idle, REFab and SRF with TR flag set to H
2. The support of half CK4 rate clock ERR signal training pattern is optional in 16 Gbit devices. The host can use Vendor ID4 (IRA3 DQ5) as described in the Info Read section to verify if this feature is supported in 16 Gbit devices. Half CK4 rate error training is mandatory in 24 Gbit and larger density devices.

If MR23 OP[11:0] (DT_ERR_PATTERN) is programmed to any value other than 0b000 (Normal mode) the CAPAR and WRCRC are not calculated and will be treated as a don’t care. The programmed ERR signal behavior will be valid by tMOD from MRS.

5.3 ERR Training (cont'd)



NOTES:

1. ERR pattern timings is shown edge aligned to CK4(int) for illustration purposes. Actual support values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. DT_ERR_PATTERN (MR23 OP[11:9] = 110_B is optional in 16Gbit and mandatory in 24Gbit and larger densities. Support in 16Gbit devices can be determined by reading VID4 (IRA3 DQ5).

Figure 31 — ERR Training Patterns

In addition, the host may train Command/Address parity error latency on ERR signal. ERR latency training is executed by the host using normal commands with DT_ERR_PATTERN (MR23 OP[11:9]) set to 3'b000 (normal mode). It should be noted that some configurations of the DRAM device may only support a variable CA parity latency.

The follow example training sequence can be used by the host to train ERR latency:

1. Command/address parity blocking should be disabled (MR15 OP1 CAPARBLK)
2. Command/Address parity must be enabled (MR15 OP0 CAPAR)
3. Wait tMOD
4. Issue NOP with CAPAR bit intentionally set incorrectly and monitor timing on CA parity error returned on ERR signal

5.4 Data Training

5.4.1 Data Training Common Features

5.4.1.1 Overview

GDDR7 SGRAMs provide a means for training the data bus (DQE and DQ[9:0]) interface training . Data training encompasses both Read training and Write training and may be required to enable high-speed data transmission between a host and DRAM device. New for GDDR7, data training can be executed with one of two different data training modes enabled: FIFO data training mode or LFSR data training mode. Both data training modes enable interface training without needing to access the memory core.

FIFO data training mode enables the host to write custom patterns to and read custom patterns from the Read FIFO for data training. While FIFO data training mode enables useful custom training patterns it cannot be used to create a long continuous PRBS pattern (due to FIFO depth limitations) which can be useful for some types of trainings to improve training accuracy and/or reduce training time. Thus, LFSR data training mode is introduced to supplement FIFO data training mode. FIFO or LFSR data training mode can be set via MR23 OP0 (DT_LFSR).

Data training utilizes the following commands:

- LDFF – Load data into Read FIFO from CA bus. A LDFF command is not allowed when MR23 OP0 (DT_LFSR) is set to 1_B to allow data training using the LFSR.
- WRTR – Write training data from DQ into FIFO, or to compare with LFSR
- RDTR – Read training data on DQ from FIFO, or from LFSR
- RDWTEC – Read LFSR write training burst error count values. A RDWTEC command is not allowed when MR23 OP0 (DT_LFSR) is set to 0_B to allow data training using the FIFO.

Data training may be performed in the following states:

- Bank Active
- Following a REFab with the TR flag set to H
- In Self-refresh with the TR flag set to H
- In Self-refresh following an exit from Self-refresh-sleep

The host must not trigger a transition to all banks Idle using a PREab or a PREpb to the last open bank before completing any ongoing data training operation.

Eye Masking

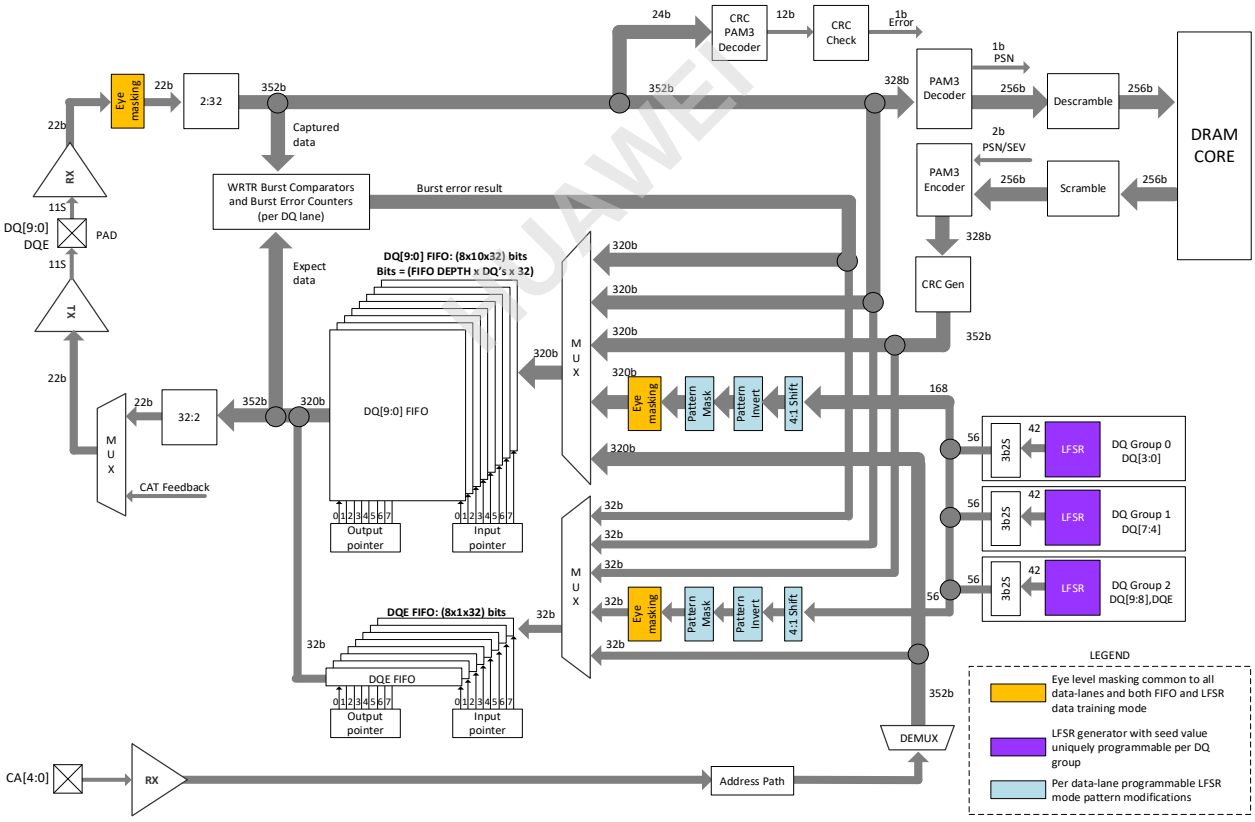
In PAM3 mode, in either FIFO or LFSR data training mode, eye masking can be applied which enables the host to train either just the upper eye, lower data eye or both depending on MR23 OP[5:4] (DT_EYE_MASK) programmed value. The programmed eye mask setting is common to all DQs and DQE within the channel. See MR23 OP[5:4] for programming details. The table below shows programming options.

5.4 Data Training (cont'd)

Table 57 — Data Training Eye Mask

Eye Mask Mode	PAM3 Input Symbol	RX Upper Output	RX Lower Output	PAM3 Output Symbol (post masking)
No Eye Mask (Normal Mode)	+1	1	1	+1
	0	0	1	0
	-1	0	0	-1
Mask Lower Eye (Lower eye is a don't care)	+1	1	X (masked)	+1
	0	0	X (masked)	0
	-1	0	X (masked)	0
Mask Upper Eye (Upper eye is a don't care)	+1	X (masked)	1	0
	0	X (masked)	1	0
	-1	X (masked)	0	-1

DRAM data path diagram is shown below with eye masking locations highlighted.



NOTES:

1. FIFO depth is vendor specific. A FIFO depth of 8 shown for illustration purposes.
2. PAM3 mode shown.

Figure 32 — Data Paths Used for Read and Write Trainings

5.4.1.2 Data Training AC Timings

Table 58 — Data Training AC Timings [1]

PARAMETER	SYMBOL	VALUES		UNIT	NOTES
		MIN	MAX		
Activate to LDFF command delay	t_{RCDLTR}		–	ns	
Activate to RDTR command delay	t_{RCDRTR}		–	ns	
Activate to WRTR command delay	t_{RCDWTR}		–	ns	
LDFF to LDFF command delay	t_{LTLTR}		–	nCK4	
LDFF31 to RDTR command delay	t_{LTRTR}		–	nCK4	
Read to LDFF command delay	t_{RDTLT}		–	nCK4	
Write to LDFF command delay	t_{WRTLT}		–	nCK4	
RDTR or Info Read to LDFF command delay	t_{RTRTLT}		–	nCK4	
RDTR to RDTR command delay	t_{CCD}		–	nCK4	
RDTR to RDWTEC to Info Read command delay	$t_{RTR2IRD}$		–	nCK4	
RDTR or RDWTEC to Mode Register Set command delay	t_{RTRMRS}		–	nCK4	
RDTR or RDWTEC to Read command delay	t_{RTRRD}		–	nCK4	
RDTR to RDWTEC or RDWTEC to RDTR command delay	t_{RTRRTR}		–	nCK4	
RDTR, RDWTEC or Info Read to Power-Down Entry command delay	t_{RTRPDE}		–	nCK4	
RDTR, RDWTEC or Info Read to Sleep Mode Entry command delay	t_{RTRSLE}		–	nCK4	
Read to RDTR or RDWTEC command delay	t_{RDRTR}		–	nCK4	
Write to RDTR or RDWTEC command delay	t_{WRRTR}		–	nCK4	
Info Read to Read, RDTR or RDWTEC command delay	t_{IRD2RD}		–	nCK4	
RDTR, RDWTEC or Info Read to Write or WRTR command delay	t_{RTRWR}		–	nCK4	
WRTR to WRTR command delay	t_{CCD}		–	nCK4	
WRTR to RDTR, RDWTEC, Info Read or Read command delay	t_{WTRTR}		–	nCK4	
Write or WRTR to Mode Register Set command delay	t_{WRMRS}		–	nCK4	
Write or WRTR to Power-Down Entry command delay	t_{WRPDE}		–	nCK4	
WRTR to Sleep Mode Entry command delay	t_{WTSLE}		–	nCK4	
WRTR to Write command delay	t_{WTRWR}		–	nCK4	
Write to WRTR command delay	t_{WRWTR}		–	nCK4	
REFab to RDTR or WRTR command delay	t_{KO}		–	ns	
NOTE 1 See AC Timings section for more details.					

5.4.1.3 FIFO Data Training Mode

FIFO data training mode enables the host to train the data link using custom programmed patterns by writing to or reading from the Read FIFO using LDFF, WRTR and RDTR commands. The Read FIFO allows each data lane to support a unique training pattern with the pattern length limited to the Read FIFO depth. The Read FIFO depth is vendor specific and should be read out by the host via IRD command to IRA1 (PAM3 FIFO Depth and NRZ FIFO Depth). RDWTEC commands are not allowed when MR23 OP0 (DT_LFSR) is set to 0_B to allow data training using the FIFO. The following additional command restrictions must be followed when MR23 OP0 (DT_LFSR) is set to 0_B to allow data training using the FIFO.

The total number of LDFF commands to burst 31 plus the total number of WRTR commands modulo FIFO depth must be equal to the total number of RDTR commands modulo FIFO depth. That is, the condition show in Equation 1 must be satisfied. If this condition is violated the Read FIFO pointers may need to be reset. The interleaving of LDFF and WRTR without RDTR is not allowed.

Equation 1:

(Total LDFF31 commands + total WRTR commands) mod(FIFO depth) = Total RDTR commands mod(FIFO depth)

After loading the Read FIFO via LDFF or WRTR commands its contents are invalidated if any commands are issued other than LDFF, RDTR, WRTR, NOP, RCKSTRT, RCKSTOP, REFpb, REFab, RFMpb, RFMab, and MRS to MR[26:23, 21:16, 7:5].

To ensure FIFO data training pointers are correctly set, the following condition will reset the Read FIFO pointers:

- Read FIFO pointers will be reset before the expiration of tCSP_POST after a SGRAM Reset, Sleep Exit, or CA Training Exit
- Issuing an MRS command to program MR23 OP8 (FIFO_PTR_RST) to 1_B. MR23 OP8 is a self-clearing Mode Register.

When the Read FIFO pointers are reset, the host can assume all modulo FIFO depth commands are satisfied. A Read FIFO pointer reset invalidates the Read FIFO contents. Subsequently, LDFF or WRTR commands will be required to reload the Read FIFO.

5.4.1.4 LFSR Data Training Mode

LFSR (Linear Feedback Shift Register) data training mode enables the host to train the data link using a PRBS15 or PRBS11 (Pseudo Random Binary Sequence) training pattern. Training may encompass lane-to-lane phase de-skewing, frame alignment, VREF training and IO equalization value training. LFSR training mode allows the host to utilize multiple Fibonacci LFSRs for PRBS pattern generation along with associated LSFR data training mode features such as lane-to-lane pattern shifting, pattern inversion and pattern masking to further improve link training. Note that LDFF commands are not allowed when MR23 OP0 (DT_LFSR) is configured to enable LFSR training mode.

To use LFSR pattern generator with Read or Write training MR23 OP0 (DT_LFSR) must be set to 1_B. There are three logical DQ groupings. Each group sources a common LFSR for that group. The groups are split as follows and illustrated in *FIGURE 33*:

- Group 0: DQ[3:0]
- Group 1: DQ[7:4]
- Group 2: DQ[9:8] and DQE

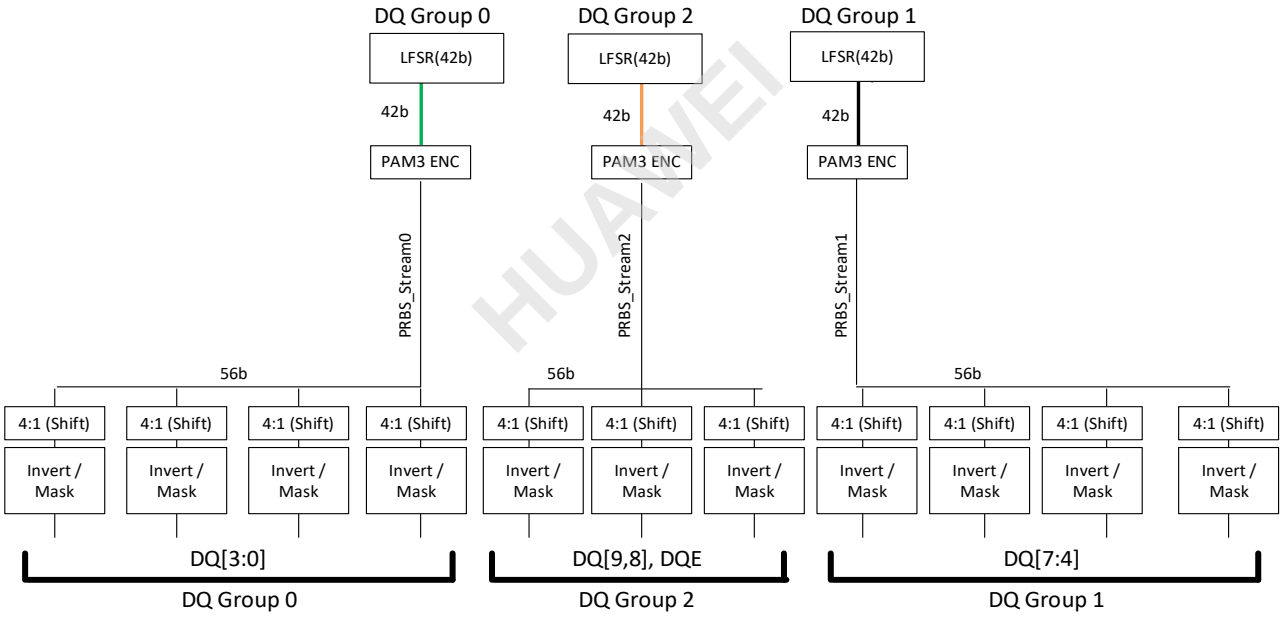


Figure 33 — Write Training LFSR DQ Groups

MR23 OP1 (LFSR_TYPE) can be set to configure all Fibonacci LFSRs to generate a PRBS15 or PRBS11 pattern. The following polynomials are supported corresponding to each PRBS type selected via MRS.

- PRBS15 = x^15 + x^14 + 1
- PRBS11 = x^11 + x^9 + 1

5.4.1.4 LFSR Data Training Mode (cont'd)

Whenever MR23 OP1 (LFSR_TYPE) is updated to change the selected polynomial (PRBS15 or PRBS11), the programmed seed value is no longer valid and must be reprogrammed via MR24 and MR25.

Table 59 — Encoder for CRC, Data Remainder, and LFSR Training

3b2S PAM3 Encoder/Decoder Truth Table								
3 Bits			Internal Binary Representation				2 Trits	
MSB		LSB	S1 MSB	S1 LSB	S0 MSB	S0 LSB		
b2	b1	b0	b3	b2	b1	b0	S1	S0
0	0	0	0	1	0	0	0	-1
0	0	1	0	0	1	1	-1	+1
0	1	0	1	1	0	0	+1	-1
0	1	1	0	1	1	1	0	+1
Invalid			0	1	0	1	0	0
1	0	0	0	0	0	0	-1	-1
1	0	1	0	0	0	1	-1	0
1	1	0	1	1	0	1	+1	0
1	1	1	1	1	1	1	+1	+1

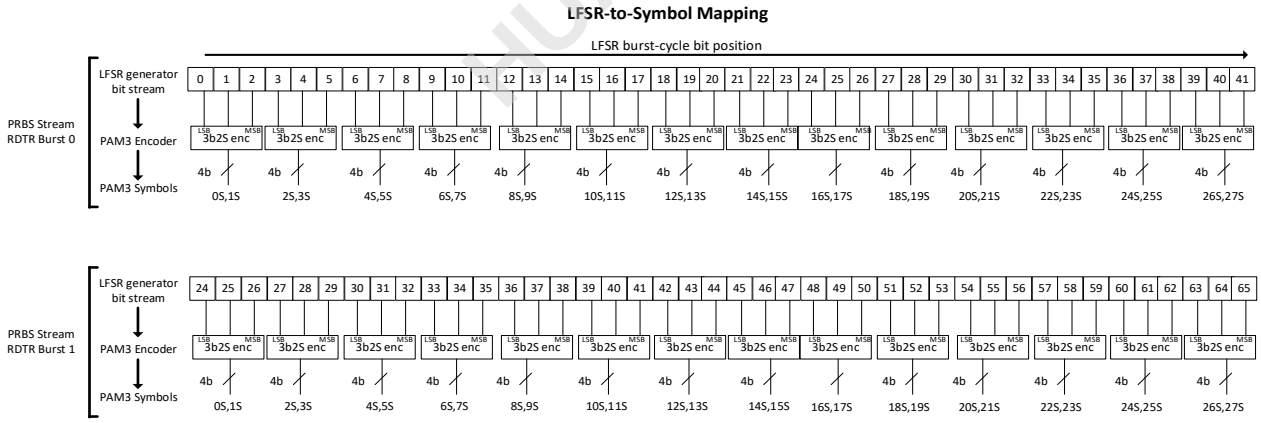


Figure 34 — LFSR to 3b2S Encoder Mapping

In *FIGURE 34*, bit "0" is considered the LSB binary input to the 3b2S encoder and bit "2" is the MSB binary input to the 3b2S encoder. This ordering is crucial as both the host and DRAM must have matching pattern generators which include LFSR, 3b2S encoder, pattern shift, pattern masking and pattern inversion architecture. From burst 0 to burst 1 the LFSR bit stream advances by 24bits. This is because 24bits from the LFSR pattern generator post 3b2S encoder creates 16 symbols worth of data which is equivalent to one full burst. The remaining bits [24:41] are lookahead bits used enable up to a 12-symbol pattern shift selectable per DQ lane. The symbol shift can be programmed per data lane via MR26 OP[1:0].

5.4.1.4 LFSR Data Training Mode (cont'd)

LFSR Pattern Generator Seeding and Advancement

The LFSR seed values must be programmed by the host prior to data training in LFSR mode in order to guarantee the host and DRAM LFSR pattern generators are aligned. The LFSR seed values are programmed via MR24 and MR25. A unique seed value can be programmed for each of the three LFSR generators. The seed value can be programmed up to 15bits. When the LFSR type is configured to be PRBS15 (via MR23 OP1) the entire 15bits of the programmed seed value are used to seed the LFSR. When the LFSR type is configured to PRBS11 (via MR23 OP1) only the lower 11bits of the LFSR Seed Bits [14:0] bits will be used.

The LFSR pattern generator automatically advances upon every RDTR or WRTR command while in LFSR training mode. READ or WRITE commands are allowed between RDTR and WRTR commands. The LFSR seed values do not need to be re-programmed when interleaving RDTR, WRTR, READ, and WRITE commands. The LFSR seed values are required to be programmed only after initialization, device reset or LFSR type change (MR23 OP1).

Example LFSR training sequence without needing to re-seed the LFSR generators between Read and Write training:

1) Issue MRS commands to enable LFSR data training mode and configure LFSR type, shift, inversion, and masking Mode Register values
2) Issue MRS commands to set LFSR seed values in preparation for Read training
3) Issue RDTR commands
4) Read training complete
5) Issue MRS to reset/configure burst error counters
6) Issue WRTR commands
7) Issue RDWTEC command
8) Repeat steps 6 through 7 for desired number of steps. Assumes MR23 OP3 is programmed to automatic reset after RDWTEC command.
9) Write training complete

5.4.1.4 LFSR Data Training Mode (cont'd)

LFSR pattern generated can be inverted or masked per data lane and eye masking can be applied per channel. Eye masking is applied post pattern inversion (LFSR_INV) and pattern masking (DT_LANE_MASK). Pattern masking is applied post pattern inversion (refer to *FIGURE 32*).

Table 60 — LFSR Pattern Inversion and Pattern Masking

PAM3 Input Symbol	MR26 OP2 (LFSR_INV)	MR26 OP3 (DT_LANE_MASK)	MR23 OP[5:4] (DT_EYE_MASK)	PAM3 Output Symbol
+1	0B	0B	00B	+1
0	0B	0B	00B	0
-1	0B	0B	00B	-1
+1	1B	0B	00B	-1
0	1B	0B	00B	0
-1	1B	0B	00B	+1
+1	Don't Care	1B	00B	+1
0	Don't Care	1B	00B	+1
-1	Don't Care	1B	00B	+1
+1	0B	0B	01B	+1
0	0B	0B	01B	0
-1	0B	0B	01B	0
+1	1B	0B	01B	0
0	1B	0B	01B	0
-1	1B	0B	01B	+1
+1	0B	0B	10B	0
0	0B	0B	10B	0
-1	0B	0B	10B	-1
+1	1B	0B	10B	-1
0	1B	0B	10B	0
-1	1B	0B	10B	0
+1	Don't Care	1B	01B	+1
0	Don't Care	1B	01B	+1
-1	Don't Care	1B	01B	+1
+1	Don't Care	1B	10B	0
0	Don't Care	1B	10B	0
-1	Don't Care	1B	10B	0

5.4.1.5 Example LFSR Implementation

FIGURE 35 is an example of the LFSR configured as a PRBS11 generator with 42b lookahead. Please note that the data feedback to the LFSR advances the PRBS11 to generate a proper PRBS11 sequence advanced in 24b increments every burst cycle (2nCK4) the generator is run. The 24bits are encoded through a later 3b2S encoder to generate 16 PAM3 symbols in PAM3 mode. The remaining bits (25 to 42) are pattern lookahead bits used to shift/decorrelate data lane patterns from each other. The per DQ pattern shift can be programmed via MR26 OP[1:0] (LFSR_SHIFT_SEL). See *MODE REGISTERS* section for programming details.

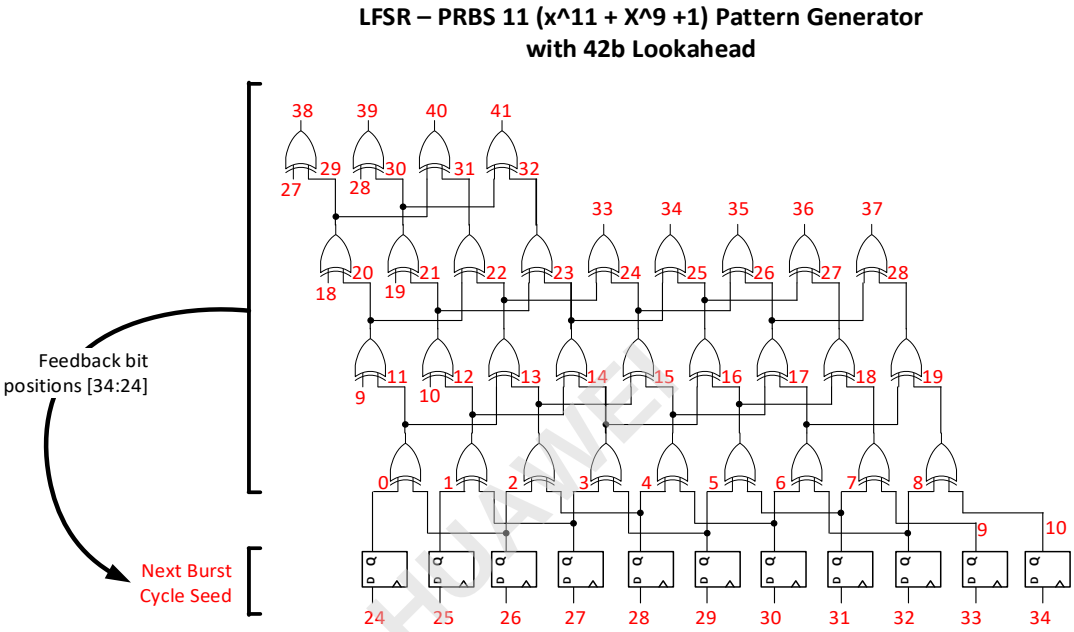


Figure 35 — Example LFSR PRBS11 Pattern Generator

The remaining LFSR example section is an example of the LFSR configured as a PRBS15 generator with 42b lookahead. Please note that the data feedback to the LFSR advances the PRBS15 to generate a proper PRBS15 sequence advanced in 24b increments every burst cycle (2nCK4) the generator is run. The 24bits are encoded through a later 3b2S encoder to generate 16 PAM3 symbols in PAM3 mode. The remaining bits (25 to 42) are pattern lookahead bits used to shift/decorrelate data lane patterns from each other. The per DQ pattern shift can be programmed via MR26 OP[1:0] (LFSR_SHIFT_SEL). See *MODE REGISTERS* section for programming details.

5.4.1.5 Example LFSR Implementation (cont'd)

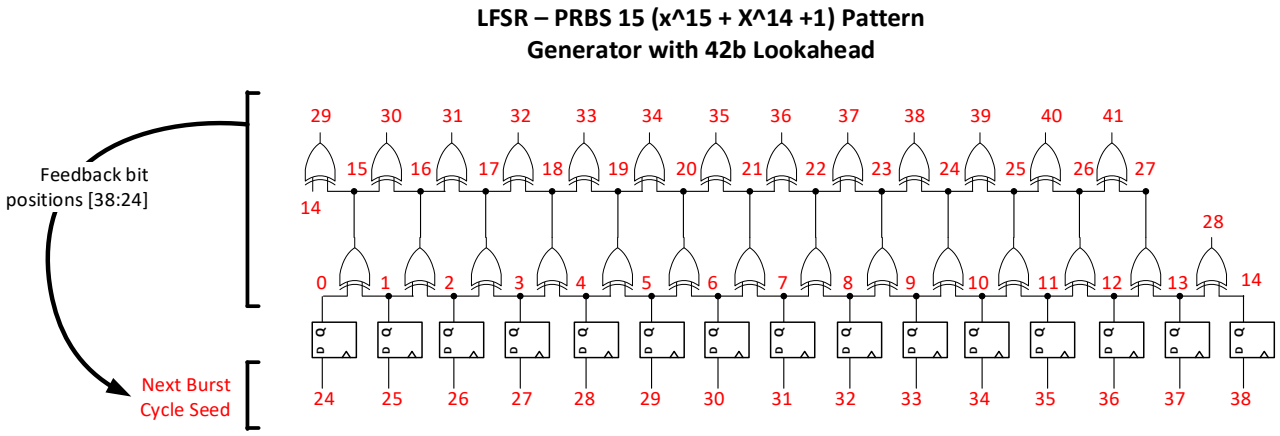


Figure 36 — Example LFSR PRBS15 Pattern Generator

Data burst output example for LFSR set to PRBS15

Example load seed values:

- DQ Group 2 = 0x3A73 (15b seed value programmed via MR24 and MR25)
- DQ Group 1 = 0x2D17 (15b seed value programmed via MR24 and MR25)
- DQ Group 0 = 0x21FF (15b seed value programmed via MR24 and MR25)

Table 61 — Example LFSR PRBS15 Pattern Generator Output (before 3b2S Encoder)

LFSR DQ Group	Burst-Cycle Number	LFSR output binary bit position	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Notes
2	0	[14:0]	0	1	1	1	0	1	0	0	1	1	1	0	0	1	1	Initial MRS Seed = 0x3A73
2	0	[29:15]	0	1	0	0	1	1	1	0	1	0	0	1	0	1	0	
2	0	[41:30]				0	1	0	0	1	1	1	0	1	1	1	1	
2	1	[14:0]	0	1	1	1	0	1	1	1	1	0	1	0	0	1	1	From previous cycle bit position [38:24]
2	1	[29:15]	0	1	0	0	1	1	0	0	0	1	1	1	0	1	0	
2	1	[41:30]				0	1	0	1	0	0	1	0	0	1	1	1	
2	2	[14:0]	1	0	0	1	0	0	1	1	1	0	1	0	0	1	1	From previous cycle bit position [38:24]
2	2	[29:15]	1	1	0	1	1	0	1	0	0	1	1	1	0	1	0	
2	2	[41:30]				1	0	1	1	1	0	1	0	0	1	1	1	
1	0	[14:0]	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1	Initial MRS Seed = 0x2D17
1	0	[29:15]	0	1	1	1	0	1	1	1	0	0	1	1	1	0	0	
1	0	[41:30]				0	1	1	0	0	1	0	1	0	0	1	0	
1	1	[14:0]	0	0	1	0	1	0	0	1	0	0	1	1	1	0	1	From previous cycle bit position [38:24]
1	1	[29:15]	1	0	1	1	1	1	0	1	1	0	1	0	0	1	1	
1	1	[41:30]				0	0	0	1	1	0	1	1	1	0	1	0	
1	2	[14:0]	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	From previous cycle bit position [38:24]
1	2	[29:15]	1	0	1	1	0	0	1	1	1	1	1	1	0	0	0	
1	2	[41:30]				0	1	0	1	0	0	0	0	0	1	0	0	
0	0	[14:0]	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	Initial MRS Seed = 0x21FF
0	0	[29:15]	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	
0	0	[41:30]				1	0	0	1	1	0	0	0	0	0	0	0	
0	1	[14:0]	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	From previous cycle bit position [38:24]
0	1	[29:15]	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	
0	1	[41:30]				1	0	0	0	0	0	0	1	1	1	1	0	
0	2	[14:0]	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	From previous cycle bit position [38:24]
0	2	[29:15]	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	
0	2	[41:30]				0	1	1	0	0	1	0	0	0	0	1	0	

5.4.1.5 Example LFSR Implementation (cont'd)

The initial seed value loaded into the LFSR is set via MR24 and MR25. See *MODE REGISTERS* section for programming details. In PAM3 mode the output pattern generated from the LFSR then goes through a DC balanced 3b2S PAM3 encoding. See 3b2S encoding table within the *PAM3* section for more details.

TABLE 62 shows an example output of the first three PAM3 encoded bursts for the stated seed value programmed into DQ group 2 LFSR generator.

Table 62 — Example WRTR Output Burt Stream for DQ[9] (DQ Group 2), where LFSR_SHIFT_SEL = 0x0, LFSR_INV = 0x0, DT_MASK = 0x0

Group 2 MRS Seed = 0x3A73, 4:1 LFSR_SHIFT_SEL = 0x0, LFSR_INV = 0x0, DT_MASK = 0x0																	
LFSR DQ Group	Pattern Stream Location	Burst Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
2	DQ[9] PAD (PAM3)	0	-1	0	-1	+1	+1	-1	0	+1	-1	0	-1	+1	+1	0	0
2	DQ[9] PAD (PAM3)	1	0	-1	+1	+1	+1	-1	0	+1	-1	0	+1	+1	+1	-1	0
2	DQ[9] PAD (PAM3)	2	-1	-1	+1	+1	+1	-1	-1	-1	-1	-1	+1	+1	+1	-1	0

After the LFSR pattern streams are PAM3 encoded and eye masking is applied the pattern streams are sent to their corresponding DQs within each DQ group. Within each DQ lane further pattern manipulation is possible. On a per DQ basis the LFSR pattern can be shifted (MR26 OP[1:0] → LFSR_SHIFT_SEL), inverted (MR26 OP2 → LFSR_INV), and masked (MR26 OP3 → DT_LANE_MASK). See MR26 for programming details and usage.

TABLE 63 shows example PAM3 encoded data training pattern output for various lane pattern configurations.

Table 63 — Example WRTR Output Burst Stream for Single Burst from DQE and DQ[9:0]

PRBS Group 0 Seed = 0x21FF, Group 1 Seed = 0x2D17, Group 2 Seed = 0x3A73

LFSR DQ Group	Pattern Stream Location	MR26 OP[1:0] (LFSR_SHIFT_SEL)	MR26 OP2 (LFSR_INV)	MR26 OP3 (DT_LANE_MASK)	Burst Number	bits or symbols	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28
2	LFSR output	NA	NA	NA	0	[41:0]															0	1	0	0	1	1	1	0	1	1	1	1	0	1
2	PAM3 Encoded (Bin)	NA	NA	NA	0	[55:0]	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	1
2	PAM3 Encoded (Trit)	NA	NA	NA	0	S[15:0]	+1		-1		0		+1		-1		0		+1		+1		+1		-1		0		+1		-1		0	
2	Post eye mask = 0x00	NA	NA	NA	0	[55:0]	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	1
2	DQE post FIFO	0	0	0	0	[31:0]																									0	0	0	1
2	DQ[9] post FIFO	2	0	0	0	[31:0]																									0	0	0	1
2	DQ[8] post FIFO	3	0	0	0	[31:0]																									1	1	0	0
1	LFSR output	NA	NA	NA	0	[41:0]															0	1	1	0	0	1	0	1	0	0	1	0	0	1
1	PAM3 Encoded (Bin)	NA	NA	NA	0	[55:0]	0	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
1	PAM3 Encoded (Trit)	NA	NA	NA	0	S[15:0]	0		+1		-1		+1		+1		-1		+1		-1		0		+1		-1		0		+1		0	
1	Post eye mask = 0x00	NA	NA	NA	0	[55:0]	0	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
1	DQ[7] post FIFO	3	0	0	0	[31:0]																									0	1	1	1
1	DQ[6] post FIFO	2	0	0	0	[31:0]																									1	1	0	0
1	DQ[5] post FIFO	1	0	0	0	[31:0]																									0	1	1	1
1	DQ[4] post FIFO	0	0	0	0	[31:0]																									1	1	0	1
0	LFSR output	NA	NA	NA	0	[41:0]															1	0	0	1	1	0	0	0	0	0	0	0	0	1
0	PAM3 Encoded (Bin)	NA	NA	NA	0	[55:0]	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0
0	PAM3 Encoded (Trit)	NA	NA	NA	0	S[15:0]	-1		-1		+1		0		0		-1		0		-1		0		+1		0		-1		-1		-1	
0	Post eye mask = 0x00	NA	NA	NA	0	[55:0]	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0
0	DQ[3] post FIFO	0	0	0	0	[31:0]																									0	0	0	0
0	DQ[2] post FIFO	0	1	0	0	[31:0]																									1	1	1	1
0	DQ[1] post FIFO	0	0	1	0	[31:0]																									1	1	1	1

LFSR DQ Group	Pattern Stream Location	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	LFSR output	0	0	1	1	1	0	1	0	0	1	0	1	0	0	1	1	1	0	1	0	0	1	1	1	0	0	1	1
2	PAM3 Encoded (Bin)	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1	1
2	PAM3 Encoded (Trit)	-1		+1		+1		-1		0		+1		-1		0		-1		+1		+1		0		0		+1	
2	Post eye mask = 0x00	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1	1
2	DQE post FIFO	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1	1
2	DQ[9] post FIFO	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1	0	0	0	1	1	1
2	DQ[8] post FIFO	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	1	1
1	LFSR output	1	1	0	1	1	1	0	0	1	1	1	0	0	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1
1	PAM3 Encoded (Bin)	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1
1	PAM3 Encoded (Trit)	0		+1		-1		-1		+1		-1		+1		0		-1		-1		+1		-1		+1		+1	
1	Post eye mask = 0x00	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1
1	DQ[7] post FIFO	0	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	0	1	1	1
1	DQ[6] post FIFO	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	0	1	1	1	0	0	0	0	1	1	0	0
1	DQ[5] post FIFO	0	0	0	1	1	1	0	1	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0
1	DQ[4] post FIFO	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1
0	LFSR output	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1
0	PAM3 Encoded (Bin)	0	1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	PAM3 Encoded (Trit)	0		-1		0		-1		+1		-1		0		-1		+1		+1		+1		+1		+1		+1	
0	Post eye mask = 0x00	0	1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	DQ[3] post FIFO	0	1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	DQ[2] post FIFO	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	DQ[1] post FIFO	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

5.4.1.5 Example LFSR Implementation (cont'd)

Example Verilog code for a LFSR generator (PRBS15 and PRBS11) with pattern shift, inversion, and masking logic:

```
module g7_prbs_lookahead(input clk, input reset, input advance, input [14:0] seed, input prbs_mode, output [55:0] symbols);
	wire [41:0] bitstream;
	reg [14:0] current_seed;
	wire [41:0] prbs15_bitstream;
	wire [13:0] prbs15_28_15;
	wire [12:0] prbs15_41_29;
	wire [41:0] prbs11_bitstream;
	wire [8:0] prbs11_19_11;
	wire [8:0] prbs11_28_20;
	wire [8:0] prbs11_37_29;
	wire [3:0] prbs11_41_38;

	always @(posedge clk) begin
		if (reset) begin
			current_seed[14:0] <= seed[14:0];
		end
		else if (advance) begin
			current_seed[14:0] <= bitstream[38:24];
		end
		else begin
			current_seed[14:0] <= current_seed[14:0];
		end
	end
	// current_seed[14:11] is ignored and unused in prbs11 mode
	assign bitstream = prbs_mode ? prbs11_bitstream : prbs15_bitstream;
	assign prbs15_bitstream[41:0] = {prbs15_41_29, prbs15_28_15, current_seed[14:0]};
	assign prbs15_28_15[13:0] = {prbs15_bitstream[14]^prbs15_bitstream[13],
					prbs15_bitstream[13]^prbs15_bitstream[12],
					prbs15_bitstream[12]^prbs15_bitstream[11],
					prbs15_bitstream[11]^prbs15_bitstream[10],
					prbs15_bitstream[10]^prbs15_bitstream[9],
					prbs15_bitstream[9]^prbs15_bitstream[8],
					prbs15_bitstream[8]^prbs15_bitstream[7],
					prbs15_bitstream[7]^prbs15_bitstream[6],
					prbs15_bitstream[6]^prbs15_bitstream[5],
					prbs15_bitstream[5]^prbs15_bitstream[4],
					prbs15_bitstream[4]^prbs15_bitstream[3],
					prbs15_bitstream[3]^prbs15_bitstream[2],
					prbs15_bitstream[2]^prbs15_bitstream[1],
					prbs15_bitstream[1]^prbs15_bitstream[0]};
	assign prbs15_41_29[12:0] = {prbs15_bitstream[27]^prbs15_bitstream[26],
					prbs15_bitstream[26]^prbs15_bitstream[25],
					prbs15_bitstream[25]^prbs15_bitstream[24],
					prbs15_bitstream[24]^prbs15_bitstream[23],
					prbs15_bitstream[23]^prbs15_bitstream[22],
					prbs15_bitstream[22]^prbs15_bitstream[21],
					prbs15_bitstream[21]^prbs15_bitstream[20],
					prbs15_bitstream[20]^prbs15_bitstream[19],
					prbs15_bitstream[19]^prbs15_bitstream[18],
					prbs15_bitstream[18]^prbs15_bitstream[17],
					prbs15_bitstream[17]^prbs15_bitstream[16],
					prbs15_bitstream[16]^prbs15_bitstream[15],
					prbs15_bitstream[15]^prbs15_bitstream[14]};
```

5.4.1.5 Example LFSR Implementation (cont'd)

```
        assign prbs11_bitstream[41:0] = {prbs11_41_38, prbs11_37_29, prbs11_28_20, prbs11_19_11, current_seed[10:0]};
        assign prbs11_19_11[8:0] = {prbs11_bitstream[10]^prbs11_bitstream[8],
                                    prbs11_bitstream[9]^prbs11_bitstream[7],
                                    prbs11_bitstream[8]^prbs11_bitstream[6],
                                    prbs11_bitstream[7]^prbs11_bitstream[5],
                                    prbs11_bitstream[6]^prbs11_bitstream[4],
                                    prbs11_bitstream[5]^prbs11_bitstream[3],
                                    prbs11_bitstream[4]^prbs11_bitstream[2],
                                    prbs11_bitstream[3]^prbs11_bitstream[1],
                                    prbs11_bitstream[2]^prbs11_bitstream[0]};
        assign prbs11_28_20[8:0] = {prbs11_bitstream[19]^prbs11_bitstream[17],
                                    prbs11_bitstream[18]^prbs11_bitstream[16],
                                    prbs11_bitstream[17]^prbs11_bitstream[15],
                                    prbs11_bitstream[16]^prbs11_bitstream[14],
                                    prbs11_bitstream[15]^prbs11_bitstream[13],
                                    prbs11_bitstream[14]^prbs11_bitstream[12],
                                    prbs11_bitstream[13]^prbs11_bitstream[11],
                                    prbs11_bitstream[12]^prbs11_bitstream[10],
                                    prbs11_bitstream[11]^prbs11_bitstream[9]};
        assign prbs11_37_29[8:0] = {prbs11_bitstream[28]^prbs11_bitstream[26],
                                    prbs11_bitstream[27]^prbs11_bitstream[25],
                                    prbs11_bitstream[26]^prbs11_bitstream[24],
                                    prbs11_bitstream[25]^prbs11_bitstream[23],
                                    prbs11_bitstream[24]^prbs11_bitstream[22],
                                    prbs11_bitstream[23]^prbs11_bitstream[21],
                                    prbs11_bitstream[22]^prbs11_bitstream[20],
                                    prbs11_bitstream[21]^prbs11_bitstream[19],
                                    prbs11_bitstream[20]^prbs11_bitstream[18]};
        assign prbs11_41_38[3:0] = {prbs11_bitstream[32]^prbs11_bitstream[30],
                                    prbs11_bitstream[31]^prbs11_bitstream[29],
                                    prbs11_bitstream[30]^prbs11_bitstream[28],
                                    prbs11_bitstream[29]^prbs11_bitstream[27]};
        genvar i;
        generate
                for (i = 0; i < 14; i = i + 1) begin
                        g7_3b2s_enc enc (.binary_data(bitstream[(2+(i*3)) : (i*3)]),
                                .binary_symbols(symbols[(3+(i*4)) : (i*4)]));
                end
        endgenerate
endmodule // g7_prbs_lookahead

module g7_3b2s_enc(input [2:0] binary_data, output [3:0] binary_symbols);
        assign binary_symbols = (binary_data == 3'b000) ? 4'b0100 : //  0, -1
                                (binary_data == 3'b001) ? 4'b0011 : // -1,  1
                                (binary_data == 3'b010) ? 4'b1100 : //  1, -1
                                (binary_data == 3'b011) ? 4'b0111 : //  0,  1
                                (binary_data == 3'b100) ? 4'b0000 : // -1, -1
                                (binary_data == 3'b101) ? 4'b0001 : // -1,  0
                                (binary_data == 3'b110) ? 4'b1101 : //  1,  0
                                                          4'b1111;  //  1,  1
endmodule // g7_3b2s_enc

module g7_eye_mask(input [1:0] dt_eye_mask, input [1:0] raw_symbol, output [1:0] masked_symbol);
        assign masked_symbol[1:0] = {raw_symbol[1] & ~dt_eye_mask[1], raw_symbol[0] | dt_eye_mask[0]};
endmodule // g7_eye_mask

module g7_lane_mask(input dt_mask, input [1:0] raw_symbol, output [1:0] masked_symbol);
        assign masked_symbol[1:0] = {raw_symbol[1] | dt_mask, raw_symbol[0] | dt_mask};
endmodule // g7_lane_mask
```

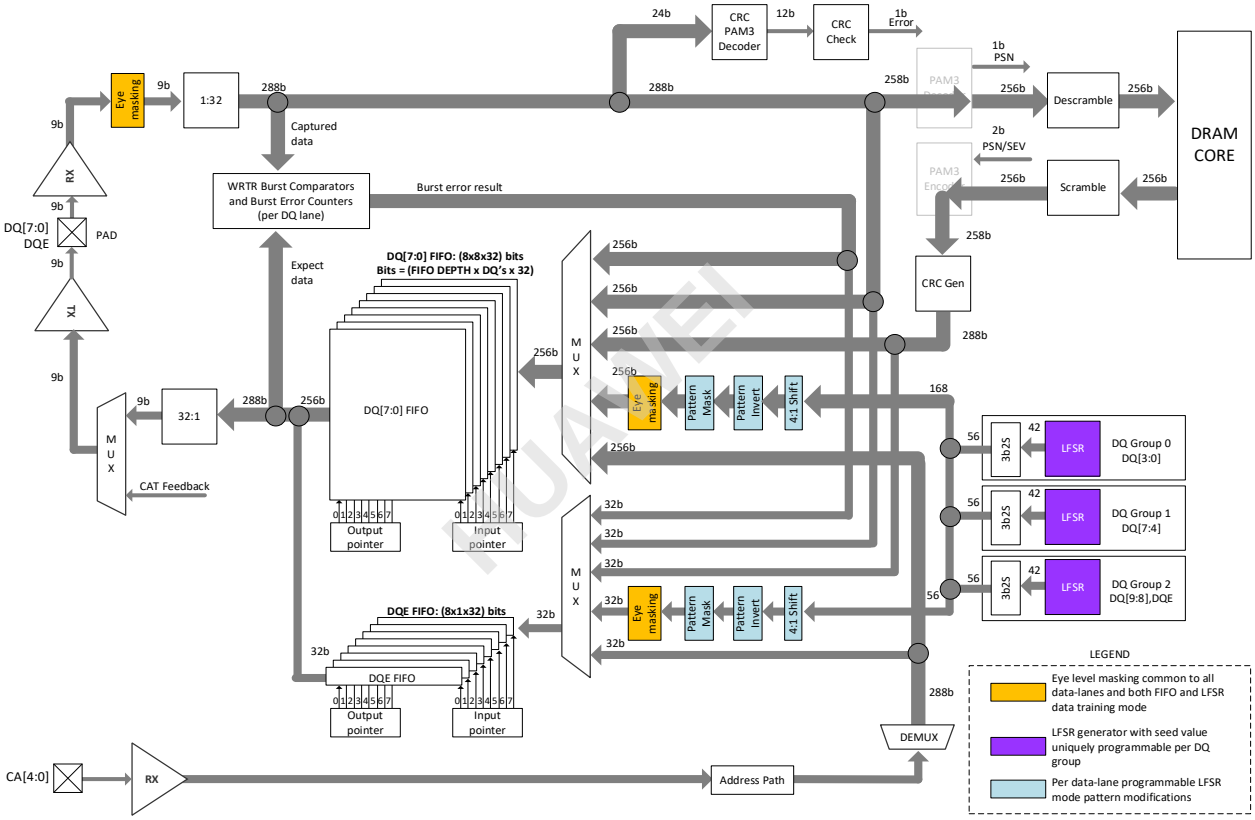
5.4.1.5 Example LFSR Implementation (cont'd)

```
module g7_symbol_invert(input lfsr_inv, input [1:0] raw_symbol, output [1:0] inverted_symbol);
        assign inverted_symbol = (~lfsr_inv || (raw_symbol == 2'b01)) ? raw_symbol : ~raw_symbol;
endmodule // g7_symbol_invert
module g7_data_lane(input [55:0] lfsr_symbols, input [1:0] dt_symbol_offset, input dt_symbol_invert, input dt_lane_mask,
                    input [1:0] dt_eye_mask, output [31:0] burst_symbols);
        wire [31:0] selected_symbols;
        wire [31:0] inverted_symbols;
        wire [31:0] lane_masked_symbols;
        assign selected_symbols = (dt_symbol_offset == 2'b00) ? lfsr_symbols[31:0] : // No shift
                    (dt_symbol_offset == 2'b01) ? lfsr_symbols[39:8] : // Shift 4 symbols
                    (dt_symbol_offset == 2'b10) ? lfsr_symbols[47:16] : // Shift 8 symbols
                    lfsr_symbols[55:24]; // Shift 12 symbols
        genvar i;
        generate
                for (i = 0; i < 16; i = i + 1) begin
                    g7_symbol_invert invert (.lfsr_inv(dt_symbol_invert),
                            .raw_symbol(selected_symbols[(1+(i*2)):(i*2)]),
                            .inverted_symbol(inverted_symbols[(1+(i*2)):(i*2)]));
                    g7_lane_mask lane_mask (.dt_mask(dt_lane_mask),
                            .raw_symbol(inverted_symbols[(1+(i*2)):(i*2)]),
                            .masked_symbol(lane_masked_symbols[(1+(i*2)):(i*2)]));
                    g7_eye_mask eye_mask (.dt_eye_mask(dt_eye_mask),
                        .raw_symbol(lane_masked_symbols[(1+(i*2)):(i*2)]),
                        .masked_symbol(burst_symbols[(1+(i*2)):(i*2)]));
                end
        endgenerate
endmodule // g7_data_lane

module g7_data_group0(input clk, input reset, input advance, input [14:0] seed, input prbs_mode, input [1:0] dt_eye_mask,
                    input dq0_dt_lane_mask, input dq0_dt_symbol_invert, input [1:0] dq0_symbol_offset,
                    input dq1_dt_lane_mask, input dq1_dt_symbol_invert, input [1:0] dq1_symbol_offset,
                    input dq2_dt_lane_mask, input dq2_dt_symbol_invert, input [1:0] dq2_symbol_offset,
                    input dq3_dt_lane_mask, input dq3_dt_symbol_invert, input [1:0] dq3_symbol_offset,
                    output [31:0] dq0_burst, output [31:0] dq1_burst,
                    output [31:0] dq2_burst, output [31:0] dq3_burst);
        wire [55:0] lfsr_symbols;

        g7_prbs_lookahead lfsr (.clk(clk), .reset(reset), .advance(advance),
                .seed(seed), .prbs_mode(prbs_mode), .symbols(lfsr_symbols));
        g7_data_lane lane0 (.lfsr_symbols(lfsr_symbols), .dt_symbol_offset(dq0_symbol_offset),
                .dt_symbol_invert(dq0_dt_symbol_invert), .dt_lane_mask(dq0_dt_lane_mask),
                .dt_eye_mask(dt_eye_mask), .burst_symbols(dq0_burst));
        g7_data_lane lane1 (.lfsr_symbols(lfsr_symbols), .dt_symbol_offset(dq1_symbol_offset),
                .dt_symbol_invert(dq1_dt_symbol_invert), .dt_lane_mask(dq1_dt_lane_mask),
                .dt_eye_mask(dt_eye_mask), .burst_symbols(dq1_burst));
        g7_data_lane lane2 (.lfsr_symbols(lfsr_symbols), .dt_symbol_offset(dq2_symbol_offset),
                .dt_symbol_invert(dq2_dt_symbol_invert), .dt_lane_mask(dq2_dt_lane_mask),
                .dt_eye_mask(dt_eye_mask), .burst_symbols(dq2_burst));
        g7_data_lane lane3 (.lfsr_symbols(lfsr_symbols), .dt_symbol_offset(dq3_symbol_offset),
                .dt_symbol_invert(dq3_dt_symbol_invert), .dt_lane_mask(dq3_dt_lane_mask),
                .dt_eye_mask(dt_eye_mask), .burst_symbols(dq3_burst));

endmodule // g7_data_group0
```

5.4.1.6 Data Training in NRZ Mode

In NRZ mode both FIFO data training and LFSR data training modes are supported for DQ[7:0] and DQE. DQ[9:8] signals will remain disabled in a high-z state. Additionally, when MR5 OP10 (DQE_HZ) is programmed to high-z, the DQE signal will remain in high-z state during data training.

LFSR data training mode is supported in NRZ mode. In NRZ mode when MR23 OP0 (DT_LFSR) is set to 1_B to allow data training using the LFSR, the pattern generated and compared is equivalent to the binary representation of a pattern generated in PAM3 mode with equivalent seed, shift, inversion, and mask settings. That is, in NRZ mode each RDTR and WRTR 32-bit burst pattern is equivalent to the 32-bit pattern (16-symbol) generated in PAM3 mode which is PAM3 encoded using the 3b2S encoding as show in *TABLE 59*. See sections *LFSR DATA TRAINING MODE* and *EXAMPLE LFSR IMPLEMENTATION* for further details.



NOTES:

1. FIFO depth is vendor specific. A FIFO depth of 8 shown for illustration purposes.
2. NRZ mode shown.

Figure 37 — NRZ Mode Data Paths Used for Read and Write Trainings

When performing data training in FIFO mode, if the host switches between PAM3 and NRZ mode any content within the DRAM Read FIFO is not guaranteed to be maintained.

5.4.2 Read Training

5.4.2.1 Overview

Read training allows the host to find the data-eye optimal position (symbol training) and burst frame location (frame training) for each high-speed output of the device. Each signal (DQE and DQ[9:0]) can be individually trained during this sequence.

For Read Training the following conditions must be satisfied prior to training:

- DRAM must be in one of the following three states with satisfied condition as stated:
  - Bank Active
  - REFab with TR flag set to H
  - Self-refresh with TR flag set to H
  - In Self-refresh following an exit from Self-refresh-sleep

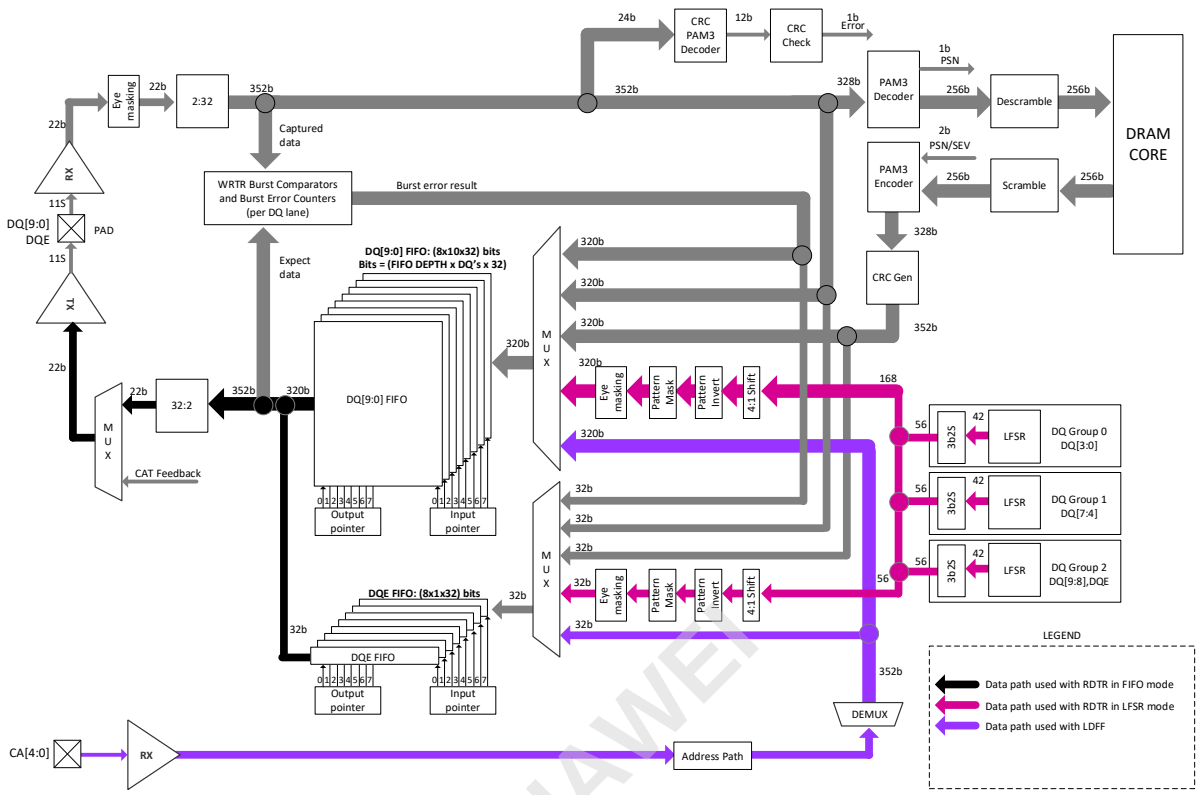
The following commands are associated with Read Training:

- LDFF to preload the Read FIFO
- RDTR to read a burst of data directly out of the Read FIFO
- RDTR to read a burst of data directly out of the Read FIFO
- RDTR with LFSR mode enabled to read a burst of data from the LFSR pattern generator.

Neither LDFF nor RDTR access the memory core. No MRS is required to enter Read Training. However, MR23 OP0 (DT_LFSR) must be set to properly enter the desired data training mode.

FIGURE 38 shows an example of the internal data paths used with LDFF and RDTR. *TABLE 58* lists AC timing parameters associated with Read Training.

5.4.2.1 Overview (cont'd)



NOTES:

1. FIFO depth is vendor specific. A FIFO depth of 8 shown for illustration purposes.
2. PAM3 mode shown.

Figure 38 — Data Paths used for Read Training

5.4.2.2 LDFF Command

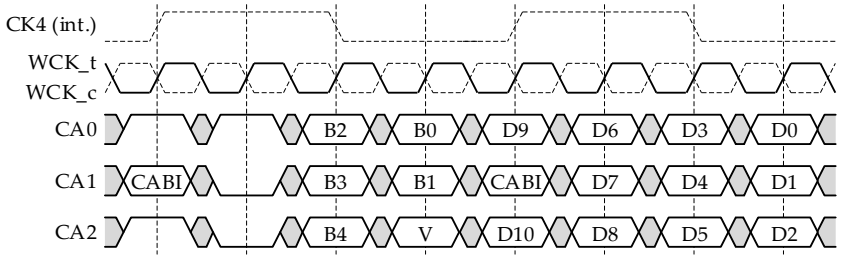
The LDFF command (*FIGURE 39*) is used to securely load data to the device's Read FIFOs via the CA bus. The READ FIFO has a vendor specific depth. For this example, a FIFO depth of 8 will be referenced (32b x 8 = 256b depth. i.e., 128 symbol depth). The LDFF command enables a unique bit pattern load for every DQ, DQE signal within the channel.

Thirty-two LDFF commands are required to fill one FIFO stage. Each LDFF command loads one burst bit position, and B[4:0] conveyed on CA[2:0] select the burst position from 0 to 31. In PAM3 mode, two LDFF commands are required for every symbol. In PAM3 mode a "10" is an invalid state which has no defined symbol representation and must be avoided.

The data pattern D[10:0] for DQE and DQ[9:0] is conveyed on CA[2:0] as shown in *FIGURE 40*.

LDFF loads the DQE and DQ[9:0] FIFO regardless of being in PAM3 or NRZ mode. It also loads the DQE FIFO regardless of the WRCRC and RDCRC Mode Register bits, and no CRC is calculated.

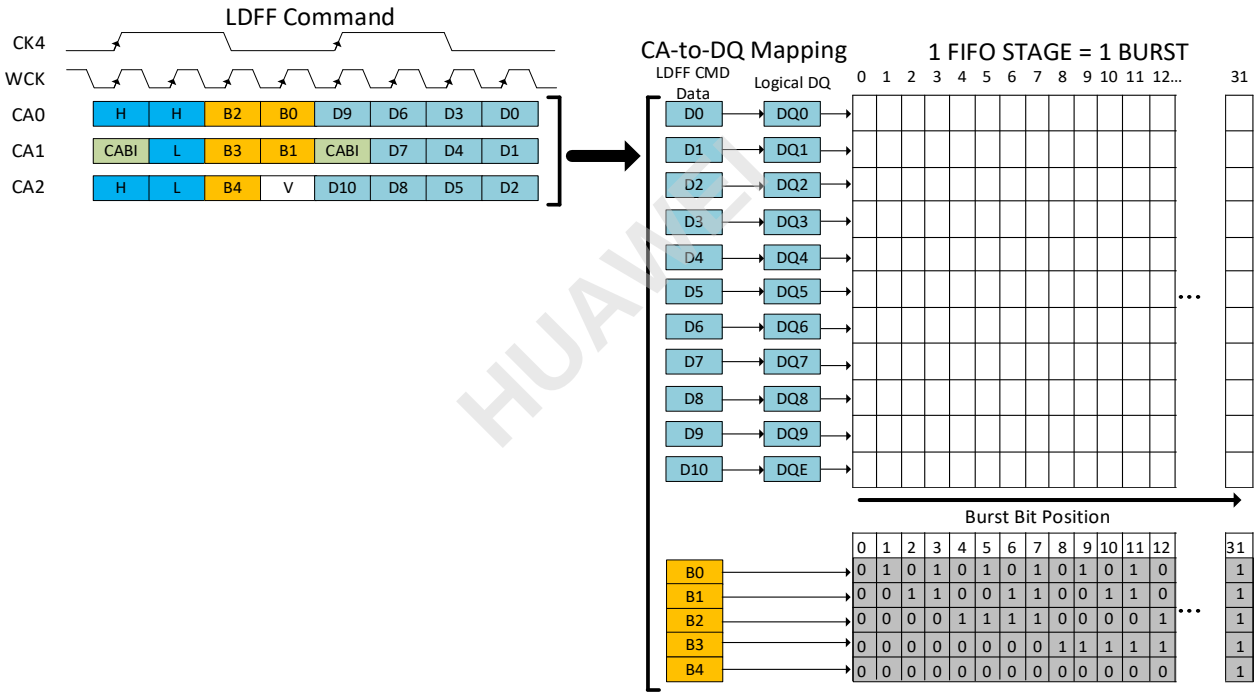
5.4.2.2 LDFF Command (cont'd)



NOTES:

1. In NRZ mode, D8 and D9 LDFF commands inputs are a don't care.

Figure 39 — LDFF Command



NOTES:

1. In NRZ mode, D8 and D9 LDFF commands inputs are a don't care.

Figure 40 — LDFF Command Address to DQE and DQ[9:0] Mapping

All bursts bit positions (0 to 31) must be loaded. LDFF commands to burst 0 to 30 may be issued in random order. The LDFF command to burst 31 (LDFF31) must be the last of 32 consecutive LDFF commands, as it effectively loads the data into the FIFO and results in a FIFO pointer increment. Consecutive LDFF commands must be spaced by at least tLTLTR.

5.4.2.2 LDFF Command (cont'd)

LDFF pattern may efficiently be replicated to the next FIFO stages by issuing consecutive LDFF commands to burst 31 (with identical data pattern). The data pattern in the scratch memory for LDFF will be available until the first RDTR command.

The DQ/DQE output buffers remain in ODT state during LDFF.

An amount of LDFF commands to burst 31 greater than the FIFO depth is allowed and shall result in a looping of the FIFO's data input.

Only LDFF, ACT, REFab, REFpb, PREab, PREpb, and MRS to MR[26:23, 21:16, 7:5] are allowed between LDFF commands until the total number of LDFF commands to burst 31 is equal to modulo FIFO depth. If this condition is violated the Read FIFO pointers may need to be reset.

In PAM3 mode, one symbol is represented by two bits of data per DQ. The two bits correspond to the symbol mapping as shown in *TABLE 64*.

Table 64 — Binary to PAM3 Symbol Mapping

Binary		PAM3 Symbol
MSB	LSB	
1	1	+1
1	0	INVALID
0	1	0
0	0	-1

A "10" binary representation of a symbol is invalid in PAM3 mode. In PAM3 mode, if a "10" is programmed during LDFF command sequences, it will have no predetermined symbol mapping. As such when the hosts issues RDTR command with a "10" symbol loaded into the Read FIFO the DRAM output symbol for that PAM3 symbol position will not be determined.

5.4.2.3 RDTR Command

A RDTR burst is initiated with a RDTR command as shown in *FIGURE 41*. No bank or column addresses are used as the data is read from the internal READ FIFO or LFSR pattern generator, not the array. The length of the burst initiated with a RDTR command is 32bits (sixteen symbols in PAM3 mode). There is no interruption nor truncation of RDTR bursts.

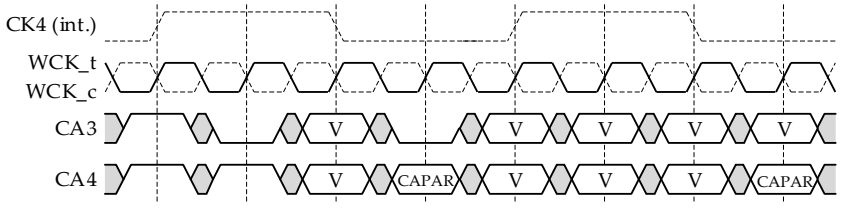


Figure 41 — RDTR Command

A RDTR command may only be issued when a bank is open, or a refresh is in progress with TR flag set to H or in self-refresh with TR flag set to H.

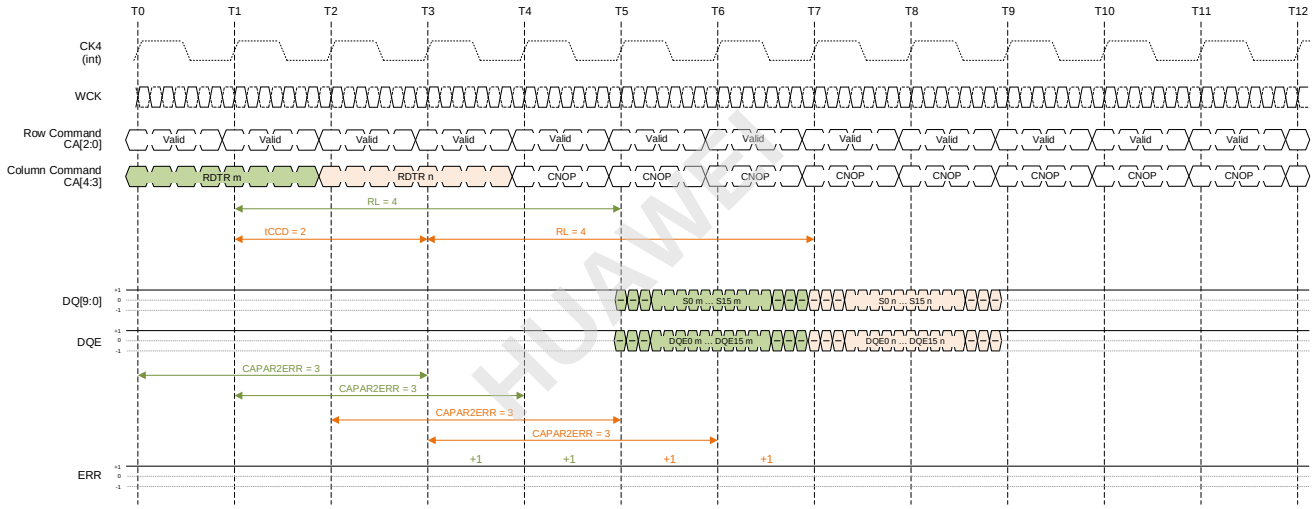
5.4.2.3 RDTR Command (cont'd)

Upon completion of a burst, assuming no other RDTR command has been initiated, all DQs and DQE will drive a value of '1' and the ODT will be enabled at a maximum of tODT_on time later. The drive value and termination value may be different due to separately defined calibration offsets. If the ODT is disabled, the signals will drive High-Z.

Data from any RDTR burst may be concatenated with data from a subsequent RDTR command. A continuous flow of data can be maintained. The first data element from the new burst follows the last element of a completed burst. The new RDTR command should be issued after the first RDTR command according to the tCCD timing.

During RDTR bursts, the first valid data-out element will be available after the Read latency (RL). RDTR valid data on DQE is issued concurrently with DQ[9:0] as DQERL does not apply.

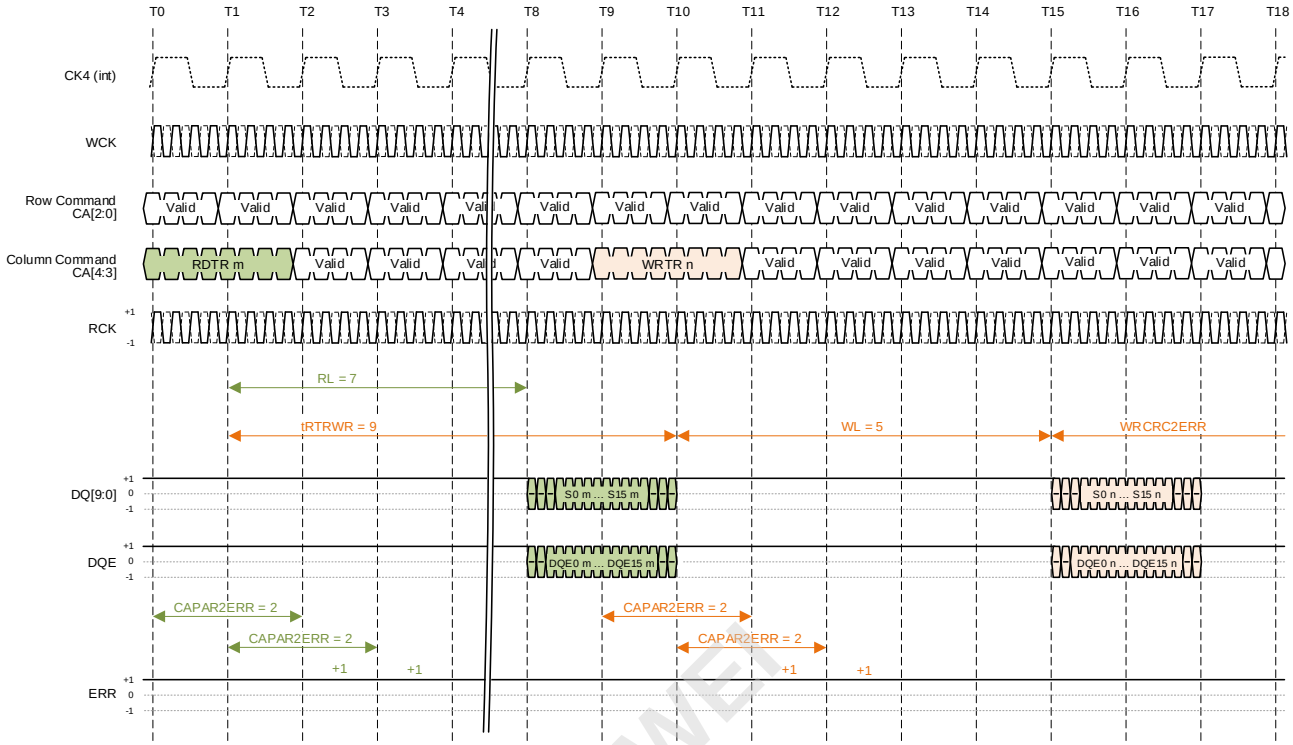
A WRTR can be issued any time after a RDTR command as long as the bus turnaround time tRTRWR is met.



NOTES:
1. RL = 4 is shown for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
3. No error (+1) for the RDTR command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.
4. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 42 — Gapless Read Training Commands

5.4.2.3 RDTR Command (cont'd)



NOTES:
1. RL = 7, tRTRWR = 9, and WL = 5 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met. If the Write is to a different bank, then an Activate (ACT) command is required to be issued before the Write and tRCDWR must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA, tWCK2DQI and tWCK2DQO = 0 are shown for illustration purposes.
4. No error (+1) for the RDTR and WRTR commands are shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings and toggle modes.

Figure 43 — RDTR to WRTR Command

5.4.2.4 FIFO Enabled RDTR

The total number of LDFF commands to burst 31 plus the total number of WRTR commands modulo FIFO depth must be equal to the total number of RDTR commands modulo FIFO depth. That is, the condition shown in Equation 1 must be satisfied. If this condition is violated the Read FIFO pointers may need to be reset.

An amount of RDTR commands greater than the FIFO depth is allowed and shall result in a looping of the FIFO's data output.

After loading the Read FIFO via LDFF or WRTR commands its contents are invalidated if any commands are issued other than LDFF, RDTR, WRTR, NOP, RCKSTRT, RCKSTOP, REFpb, REFab, RFMpb, RFMab, and MRS to MR[26:23, 21:16, 7:5].

5.4.2.5 LFSR Enabled RDTR

As a response to high data rates and PAM3 signaling of GDDR7 the LFSR data training mode has been introduced. For Read training LFSR data training mode is intended to bring the following intended benefits:

- reduce training error
    - increase training pattern complexity
    - lane training pattern decorrelation
- reduce training time
    - eliminates need to load Read FIFO with data during Read training

The following new key features are introduced to support LFSR Read training:

- three DQ groups with associated independent LFSR pattern generators (one for each group)
- LFSR pattern generators supporting PRBS15 or PRBS11 and lookahead symbol generation
- PAM3 3b2S encoder (PAM3 mode and NRZ)
- eye level masking
- per lane (DQE and DQ[9:0]) pattern symbol shift (0, 4, 8, or 12 symbols)
- per lane (DQE and DQ[9:0]) pattern mask and pattern inversion

In contrast to Read training in FIFO mode, Read training in LFSR mode is intended to allow the host to issue RDTR commands without the overhead of preloading the Read FIFO with data patterns. There is no effectiveness in using LDFF commands to preload a data pattern as RDTR with LFSR enabled will generate data patterns sourcing LFSR pattern generators. In contrast to FIFO mode, for data training in LFSR mode WRITE and READ commands are allowed between RDTR commands without needing to re-load LFSR seed values.

Typical LFSR Enabled Read Training Sequence:

- issue MRS (MR23 and MR24) commands to set LFSR type (PRBS15 or PRBS11) load LFSR seed values for each LFSR
- issue desired number of RDTR commands

As detailed in the data training common feature section a per DQ LFSR shift, LFSR inversion, and data training masking can be applied. See *DATA TRAINING COMMON FEATURES* section for details.

5.4.3 Write Training

5.4.3.1 Overview

Write training enables the host to train DQE and DQ[9:0] and to independently optimize eye margin for each data lane. The host may independently optimize DQ voltage reference levels for and equalization settings.

For Write Training, the following conditions must be satisfied prior to training:

- DRAM must be in one of the following three states with satisfied condition as stated:
  - Bank Active
  - REFab with TR flag set to H
  - Self-refresh with TR flag set to H
  - In Self-refresh following an exit from Self-refresh-sleep

The following commands are associated with Write Training:

- WRTR with FIFO mode enabled to write a burst of data directly into the Read FIFO
- WRTR with LFSR mode enabled to write a burst of data for comparison against a pre-configured pattern generator and results stored in a burst error counter
- RDTR to read a burst of data directly out of the Read FIFO
- RDWTEC to read a burst of burst error counter data (corresponding to LFSR training)

Neither WRTR nor RDTR access the memory core. No MRS is required to enter Write Training. However, MRS commands are recommended to ensure LFSR type trainings are properly initialed to synchronize expect data within the DRAM with pattern data issue by the host.

FIGURE 47 shows an example of the internal data paths used with WRTR and RDTR. *FIGURE 45* shows a typical Write training command sequence using WRTR and RDTR. *TABLE 58* lists AC timing parameters associated with WRITE Training.

5.4.3.2 WRTR Command

A WRTR burst is initiated with a WRTR command as shown in *FIGURE 44*. No bank or column addresses are used as the data is either written to the internal READ FIFO or compared against a LFRS expect pattern (depending on MR setting) and is not written to the array. The length of the burst initiated with a WRTR command is 32bits (sixteen symbols in PAM3 mode). There is no interruption nor truncation of WRTR bursts.

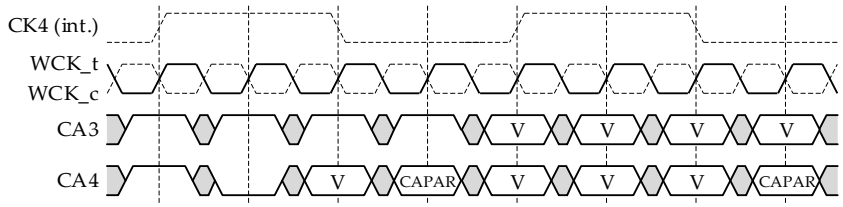


Figure 44 — WRTR Command

A WRTR command may only be issued when a bank is open or a refresh is in progress with TR flag set or in self-refresh with TR flag set to H.

PAM3 operating mode (MR0 OP8) must be enabled to write DQ[9:8] with the WRTR command. In contrast to a normal WRITE, no WRCRC is calculated and checked by the WRTR command, and no WRCRC error can be generated. However, the ERR signals will drive CAPAR error information normally as configured.

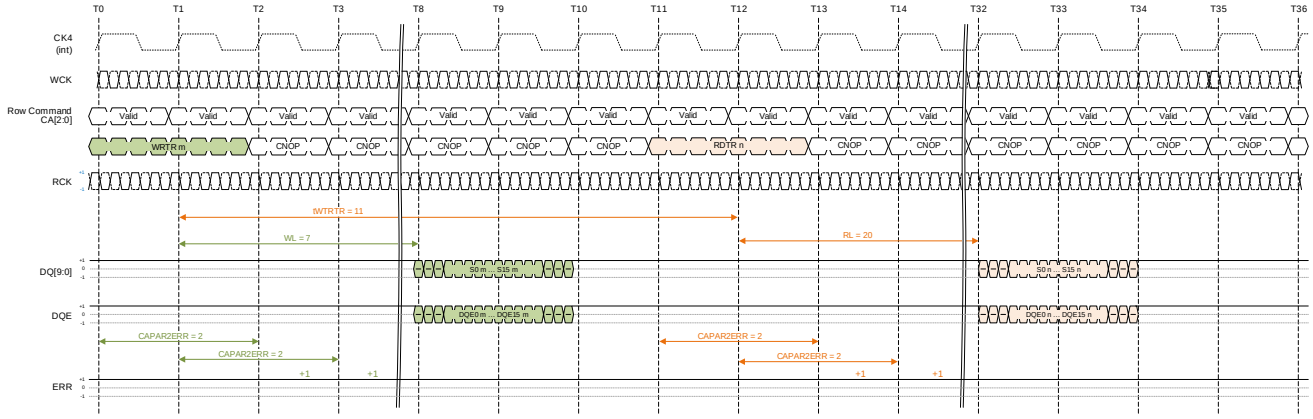
During WRTR bursts, the first valid data-in element must be available at the input latch after the Write Latency (WL). The Write Latency for WRTR is the same as for WRITE.

Upon completion of a burst, assuming no other WRTR data is expected on the bus, the DQs and DQE will be driven according to the ODT state. Any additional input data will be ignored.

Data from any WRTR burst may be concatenated with data from a subsequent WRTR command. A continuous flow of data can be maintained. The first data element from the new burst follows the last element of a completed burst. The new WRTR command should be issued after the previous WRTR command according to the tCCD timing.

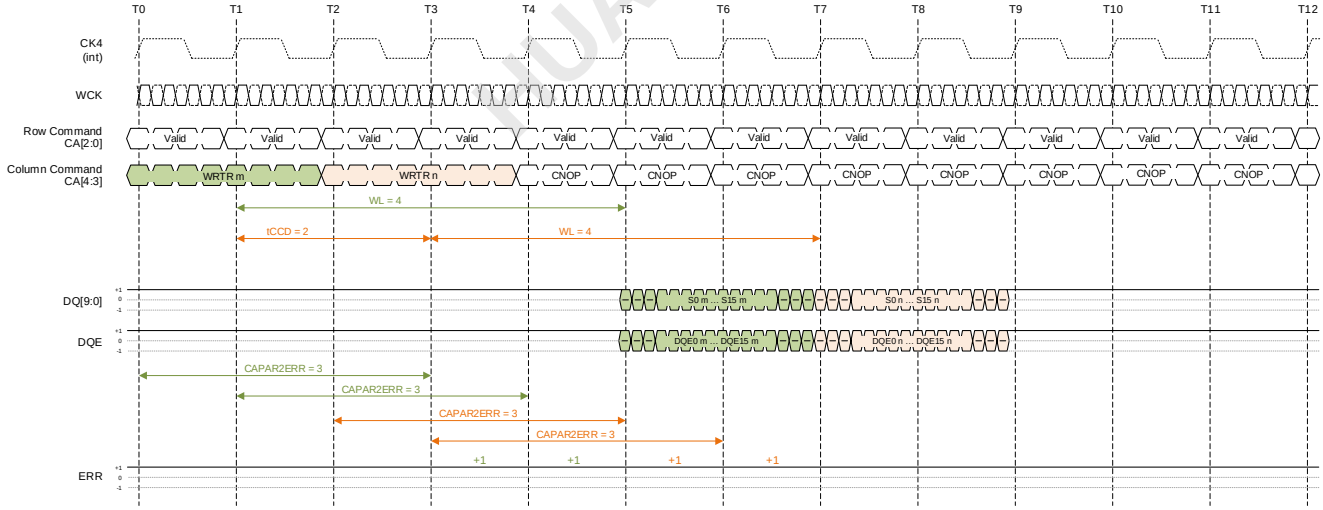
A RDTR command can be issued any time after a WRTR command if the internal bus turnaround time tWTRTR is met.

5.4.3.2 WRTR Command (cont'd)



NOTES:
1. RL = 20, tWTRTR = 11, WL = 7 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Write Latency = WL * tCK4 + tWCK2DQI. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA, tWCK2DQI, tWCK2DQO = 0 are shown for illustration purposes.
3. No error (+1) for the RDTR and WRTR commands is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.
4. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.

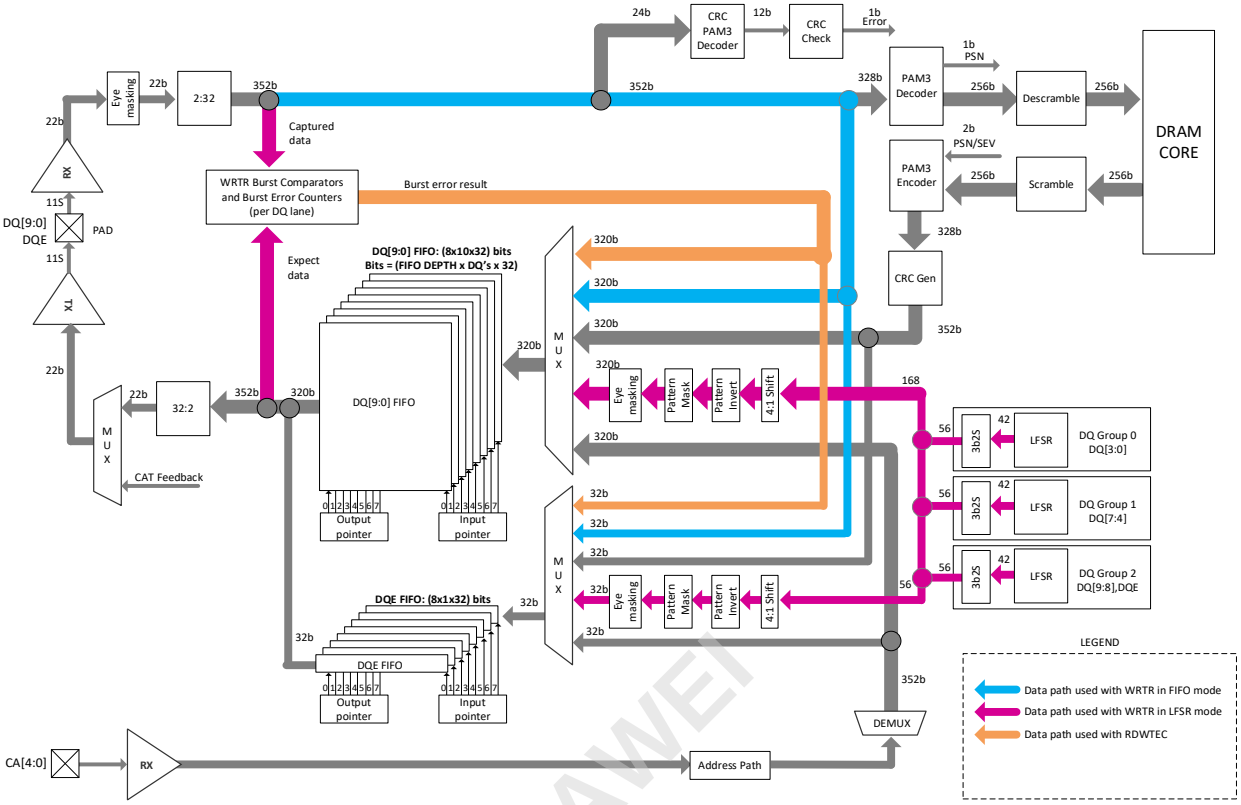
Figure 45 — WRTR to RDTR Command



NOTES:
1. WL = 4 is shown for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
3. No error (+1) for the WRTR command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.
4. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 46 — Gapless Write Training Commands

5.4.3.2 WRTR Command (cont'd)



NOTES:

1. FIFO depth is vendor specific. A FIFO depth of 8 shown for illustration purposes.
2. PAM3 mode shown.

Figure 47 — Data Paths used for Write Training

5.4.3.3 FIFO Enabled WRTR

The total number of LDFF commands to burst 31 plus the total number of WRTR commands modulo FIFO depth must be equal to the total number of RDTR commands modulo FIFO depth. That is, the condition show in Equation 1 must be satisfied. If this condition is violated the Read FIFO pointers may need to be reset.

An amount of WRTR commands equal to the FIFO depth is required to fully load the FIFO.

An amount of WRTR commands greater than the FIFO depth is allowed and shall result in a looping of the FIFO's data input.

A RDWTEC command is not allowed when MR23 OP0 (DT_LFSR) is set to 0_B to allow data training using the FIFO.

After loading the Read FIFO via LDFF or WRTR commands its contents are invalidated if any commands are issued other than LDFF, RDTR, WRTR, NOP, RCKSTRT, RCKSTOP, REFpb, REFab, RFMpb, RFMab, and MRS to MR[26:23, 21:16, 7:5].

5.4.3.4 LFSR Enabled WRTR

Due to the high data rates and PAM3 signaling of GDDR7 a LFSR training method has been introduced for Write to gain the following intended benefits:

- Reduce training error
  - increase training pattern complexity
  - lane training pattern decorrelation
- Reduce training time
  - eliminates need to Read FIFO for captured Write data during Write training

The following new key features are introduced to support LFSR Write training:

- Three DQ groups with associated independent LFSR pattern generators (one for each group)
- LFSR pattern generators supporting PRBS15 or PRBS11 and lookahead symbol generation
- PAM3 3b2S encoder (PAM3 mode and NRZ)
- Eye level masking
- Per lane (DQE and DQ[9:0]) 12bit (or dual 6 bit) burst error counter
- Per lane (DQE and DQ[9:0]) pattern symbol shift (0, 4, 8, or 12 symbols)
- Per lane (DQE and DQ[9:0]) pattern mask and pattern inversion

In contrast to Write training with FIFO, Write training with LFSR is intended to allow the host to issue any number of WRTR command bursts. There is no concept of filling a FIFO with WRTR command when LFSR type training is enabled. The host may provide any number of WRTR commands as desired. Only a single RDWTEC command is needed to read back error counter results after any desired number of WRTR commands have been issued. In contrast to FIFO mode, for data training in LFSR mode WRITE and READ commands are allowed between WRTR commands without needing to re-load LFSR seed values. Any data which may have been previously loaded into the Read FIFO will no longer be valid for use in RDTR or WRTR commands with FIFO enabled once a RDTR or WRTR command is issued with LFSR enabled; the contents of the FIFO would have been overwritten.

To use LFSR pattern generator with WRTR the DT_FIFO_LFSR MR (MR23 OP0) must be set to 1_B. There are three logical DQ groupings. Each groups sources a common LFSR for that group. The groups are split as shown in *FIGURE 33*.

5.4.3.4 LFSR Enabled WRTR (cont'd)

Typical LFSR enabled Write Training Sequence:

- Reset burst error counter
- Issue MRS commands to set LFSR type (PRBS15 or PRBS11) load LFSR seed values for each LFSR
- Issue desired WRTR commands
- Issue RDWTEC to read burst error count
    - Note if MR23 OP3 (LFSR_RST_MODE) is set to 0_B, the burst error count will automatically reset. Otherwise, a MRS must be issue to explicitly clear burst error count if desired.

The Write training shares the same LFSR generators and data path as RDTR with LFSR mode enabled. The same pattern shift, pattern inversion, pattern masking, 3b2S PAM3 encoder, and eye masking MR settings apply to both RDTR and WRTR with LFSR mode enabled.

5.4.3.5 Burst Error Counter Overview

The Burst Error Counter is used only for Write training with LFSR mode enabled. The Burst Error Counter can be configured to one of the following modes via MR23 OP7 (LFSR_CNT_MODE):

- Single 12-bit Burst Error Counter mode (Single eye per data lane)
- Dual 6-bit Burst Error Counter mode (Dual eye per data lane)

NOTE: When MR23 OP7 (LFSR_CNT_MODE) is programmed to Dual 6-bit Burst Error Counter mode the user must also program MR23 OP[5:4] (DT_EYE_MASK) to 0b00 (No data receiver eye masking). No other eye mask setting is valid in Dual 6-bit Burst Error Counter Mode.

In 12-bit burst error counter mode there is a dedicated 12-bit Burst Error Counter for each data lane (DQ[9:0] and DQE). The maximum number of burst errors which can be captured is 2^12-1. The Burst Error Counter will saturate after the maximum count value is reached and does not overflow or reset upon subsequent detected errors.

In dual 6-bit Burst Error Counter mode there are two 6-bit Burst Error Counters per data lane. One 6-bit Burst Error Counter dedicated to counting errors from the upper data eye and the other 6-bit Burst Error Counter dedicated to counting errors from the lower data eye. The maximum number of burst errors which can be captured is 2^6-1 for each counter respectively. The Burst Error Counter value will saturate for each counter independently after the maximum count value is reached for that counter. The counter does not overflow or automatically reset upon subsequent detected errors.

The host may configure MR23 OP3 (LFSR_RST_MODE) to automatically reset the error counter data after issuing RDWTEC command. Alternatively, LFSR_RST_MODE Mode Register can be configured to manually reset the Burst Error Counter. This allows the host to read out the Burst Error Counter values multiple times (issuing multiple RDWTEC commands) before issuing a manual Burst Error Counter reset via MR23 OP2 (LFSR_CNT_RESET). If LFSR_CNT_MODE (MR23 OP7) is modified, the error counter mode be reset using MR23 OP2 (LFSR_CNT_RESET).

The Burst Error Counter performs a symbol-wise data comparison and will only increase its count value by a maximum of one for every WRTR burst (16 symbols) when one or more symbol errors are detected within that WRTR burst. If the WRTR burst contains zero errors, then the counter value does not increase.

5.4.3.6 RDWTEC Command

Burst Error Counter values are read back via RDWTEC command as shown in *FIGURE 48*. The RDWTEC command is only valid when MR23 OP0 (DT_LFSR) is set to 1_B to allow data training to occur via LFSR.

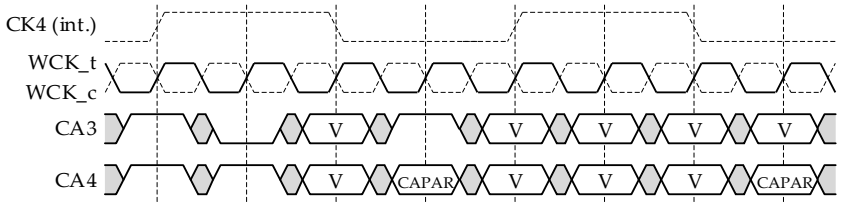


Figure 48 — RDWTEC Command

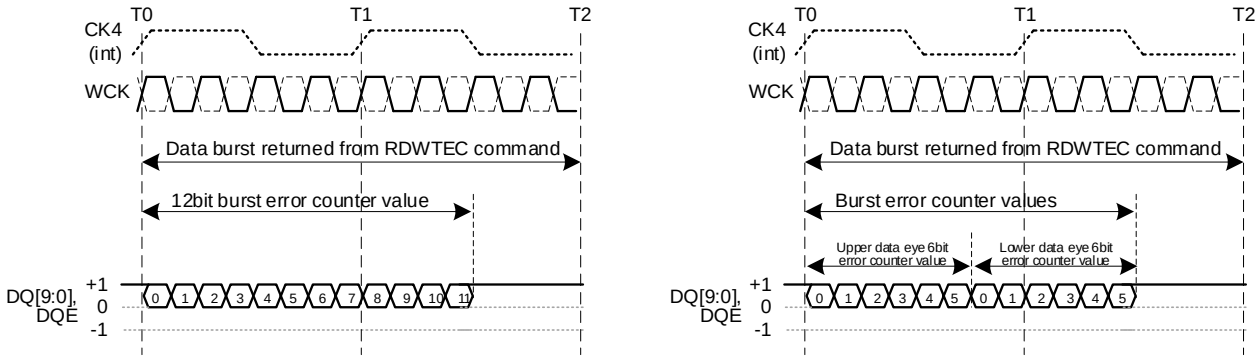
The count values are mapped to +1 and 0 levels in PAM3 mode and H / L in NRZ mode as shown below.

Table 65 — Burst Error Counter Data Encoding

RDWTEC Command Error Counter Data Encoding		
Error counter binary value	Data output symbol value	
	PAM3	NRZ
1	+1 (2b11)	H
0	0 (2b01)	L

The Burst Error counter value is returned on each data lane corresponding to the error counter of that lane. In Single 12-bit Burst Error Counter mode the RDWTEC command returns all 12 bits of the Burst Error Counter value consecutively for each data lane as shown below in *FIGURE 49*. The remainder of the burst is driven to a +1 (PAM3) or H (NRZ) level after the 12 bits of data have been driven.

In Dual 6-bit Burst Error Counter mode the RDWTEC command also returns 12 bits consecutively per data lane as shown below in *FIGURE 49*. The first 6 bits represent the Burst Error Counter value associated with the upper data eye, the last 6 bits represents the Burst Error Counter value associated with the lower data eye. The remainder of the burst is driven to a +1 (PAM3) or H (NRZ) level after the 12 bits of data have been driven.

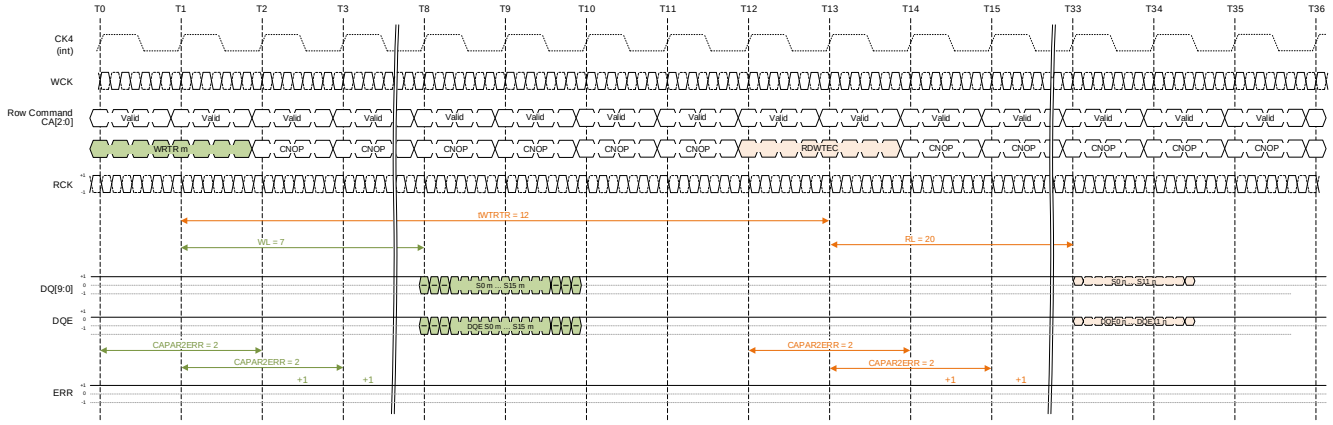


NOTE 1 Each data lane (DQ[9:0] and DQE) returns a unique error counter value associated with the error count value for that lane.

NOTE 2 PAM3 mode shown for example. See *READ* section for more details such as ODT on/off and RCK for RDWTEC transactions.

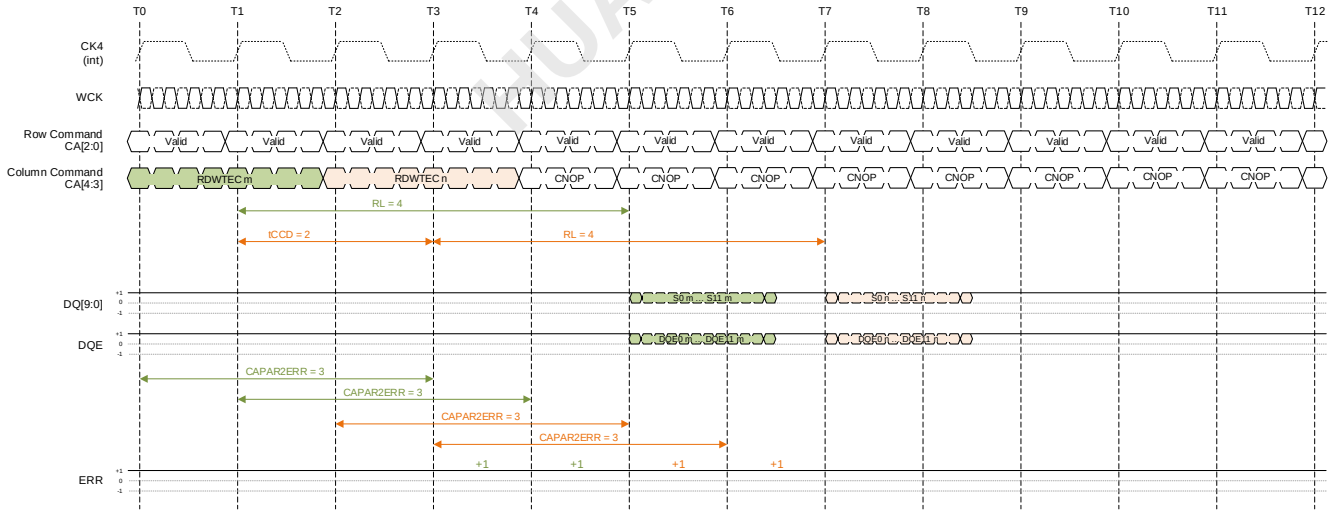
Figure 49 — Burst Error Counter Data Returned using RDWTEC Command in Single 12-bit Burst Error Counter Mode (Left) and Dual 6-bit Burst Error Counter Mode (Right)

5.4.3.6 RDWTEC Command (cont'd)



NOTES:
1. RL = 20, tWTRTR = 12, WL = 7 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Write Latency = WL * tCK4 + tWCK2DQI. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA, tWCK2DQI, tWCK2DQO = 0 are shown for illustration purposes.
3. No error (+1) for the WRTR and RDWTEC commands is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.
4. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.

Figure 50 — Write Training using WRTR and RDWTEC Commands in LFSR Training Mode



NOTES:
1. RL = 4 is shown for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
3. No error (+1) for the RDWTEC command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details.
4. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.
5. RDCRC is not calculated for the Write Training Burst Error Counter value which is returned to the host upon issuing a RDWTEC command.

Figure 51 — Gapless RDWTEC Commands in LFSR Training Mode

5.5 Command Address Oscillator (CAOSC)

As voltage and temperature change on the GDDR7 DRAM, the WCK clock tree delay will shift relative to CA and may require re-training. The WCK-to-CA clock-tree timing is denoted as tWCK2CA. The GDDR7 DRAM includes an internal WCK-to-CA interval oscillator (CAOSC) allowing the controller to measure the amount of change in tWCK2CA timing over a controller determined time interval. Thus, allowing the controller to optionally make appropriate timing adjustments to the CA path and or execute a CA training procedure.

There is a CAOSC associated with each channel and operates fully independent of any channel's operating frequency or state (e.g. bank active, bank idle, sleep, power-down, or self-refresh). Also, no WCK clock is required while the oscillator is counting.

The CAOSC Mode Register (MR23 OP6) and Info Read CAOSC COUNT registers (IRA Address [42:40]) are associated with the oscillator. Setting the CAOSC to 1 will start an internal ring oscillator that counts the number of times a signal propagates through a copy of the WCK-to-CA clock tree. The oscillator is stopped by setting the CAOSC bit back to 0.

The tWCK2CA timing parameter refers to the real WCK-to-CA relative insertion delay timing. tCAOSC refers to the CA oscillator delay which mimics the delay of tWCK2CA over voltage and temperature drift.

- tWCK2CA = Actual WCK-to-CA clock path
- tCAOSC = Interval ring oscillator delay which mimics tWCK2CA timing and VT sensitivity
- tCAOSC = tWCK2CA + Matching Error (where matching error is intended to ideally be zero)

After started, every time the CAOSC circuitry sees 2 * tCAOSC, its counter will increase by a value of 1.

The maximum count is 2^{23} - 1, and the longest run time for the oscillator to not overflow the counter can be calculated as follows:

Longest Run Time Interval = 2^{23} * (2 * tCAOSC(min))

Changing the operation frequency may also change tWCK2CA and tCAOSC timing. Therefore, controller must re-run CAOSC to measure tCAOSC after a frequency change to maximize the accuracy of measuring tWCK2CA.

The use of DCC may change tWCK2CA timing. Any timing impact from DCC will be capture by tCAOSC. The controller must re-run CAOSC to measure tCAOSC after running DCC to maximize the accuracy of measuring tWCK2CA.

The validity of the clock count is indicated by the CAOSC_COUNT_VALID bit in the CAOSC_COUNT Info Read register. The default state of 0 indicates an invalid count. The state is also set to 0 when the oscillator is started. When the oscillator stops, the CAOSC_COUNT_VALID bit is set to 1 to indicate a valid count, and the result of the counter is stored in the CAOSC_COUNT_VALUE field of the CAOSC_COUNT Info mode registers. The CAOSC_COUNT_VALID bit will remain 0 (invalid) if the counter overflows (2^{23} or more cycles) or if the oscillator is interrupted by pulling RESET_n to LOW. After the oscillator stops the controller may issue Info Read commands to read out the count after tCAOSCO timing is satisfied.

CAOSC_COUNT_VALUE = Floor(Run Time / (2 * tCAOSC)),
not accounting for measurement error terms

There are several key error components contributing to the accuracy of the CAOSC to be reliably used by the host to measure tWCK2CA VT drift behavior. The following CAOSC matching error terms are introduced to define its matching behavior:

- tCAOSCmatch(V) = Voltage sensitivity matching between tWCK2CA and tCAOSC
- tCAOSCmatch(T) = Temperature sensitivity matching between tWCK2CA and tCAOSC

5.5 Command Address Oscillator (CAOSC) (cont'd)

Additionally, the system may incur further error from CAOSC granularity error (quantization error). To minimize CAOSC granularity error the controller may adjust the accuracy of the result by running the oscillator for shorter (less accurate) or longer (more accurate) duration. This CAOSC granularity error is determined by the following equation:

$$\textbf{CA WCK Oscillator Granularity Error} = \frac{\textbf{2 * tCAOSC}}{\textbf{Run Time}}$$

Where: Run Time = Total time between the oscillator starting and stopping MRS commands

In addition to granularity error, the difference in delay between tCAOSC and the actual WCK-to-CA clock tree timing across voltage and temperature must be accounted for.

Therefore, the total accuracy of the CAOSC is given by:

$$\textbf{CAOSC Accuracy} = 1 - (\textbf{Granularity Error + Matching Error Terms})$$

The CAOSC matching error term (tCAOSCMatch) is defined as the delta min and max difference between tWCK2CA and tCAOSC over voltage and temperature. The difference between tWCK2CA and tCAOSC is denoted as tCAOSCoffset. The difference between max(tCAOSCoffset) and min(tCAOSCoffset) is denoted as tCAOSCMatch as shown in the equations below:

$$\textbf{tCAOSCMatch(V)} = \textbf{max(tCAOSCoffset(V))} - \textbf{min(tCAOSCoffset(V))}$$

$$\textbf{tCAOSCMatch(T)} = \textbf{max(tCAOSCoffset(T))} - \textbf{min(tCAOSCoffset(T))}$$

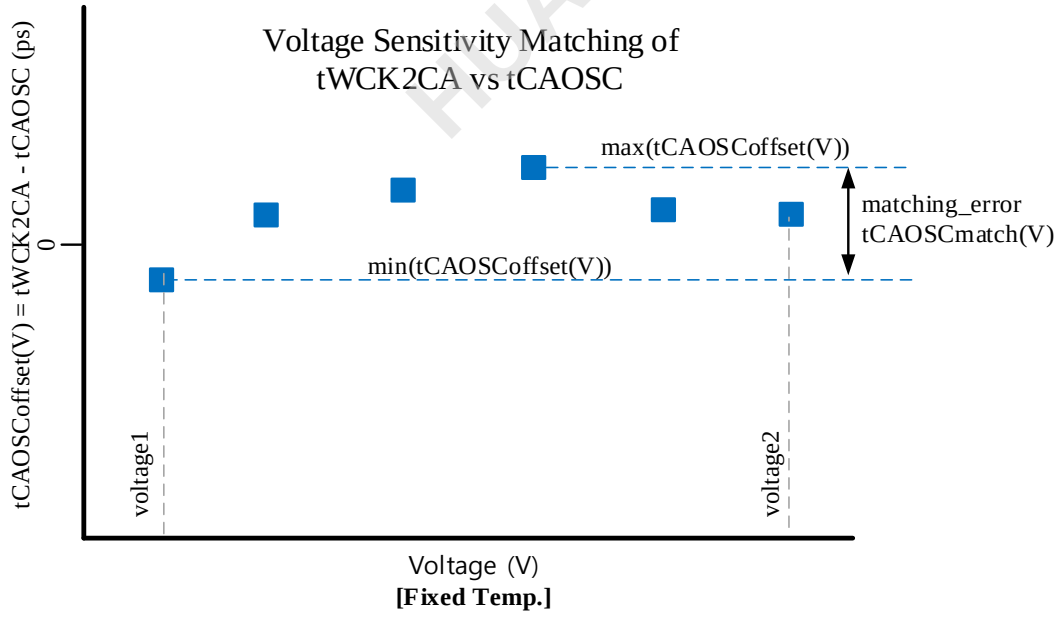


Figure 52 — CA Oscillator Offset (tCAOSCoffset(V))

5.5 Command Address Oscillator (CAOSC) (cont'd)

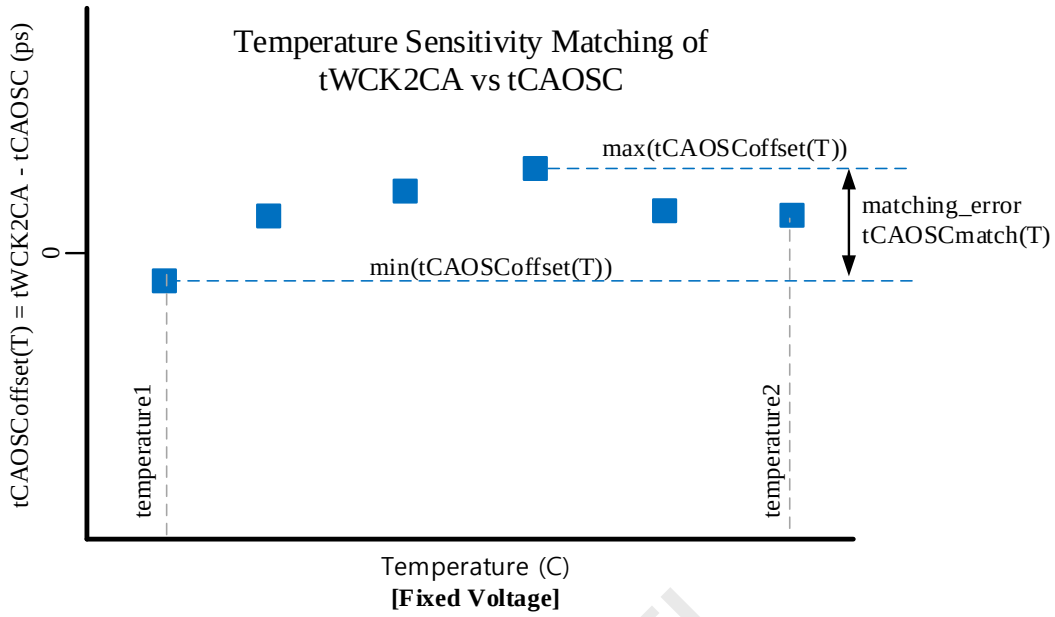


Figure 53 — Oscillator Offset (tCAOSCoffset(T))

Table 66 — WCK Oscillator Matching Error Specification

Parameter	Symbol	Min	Max	Units	Notes
CA WCK Oscillator Offset Error: voltage variation	tCAOSCoffset(V)			ps	1, 2, 3, 4, 5, 6
CA WCK Oscillator Offset Error: temperature variation	tCAOSCoffset(T)			ps	1, 2, 3, 4, 5, 6, 7

NOTE 1 The tCAOSCoffset is the difference between the actual WCK-to-CA path and the interval oscillator delay (tCAOSC) over voltage or temp.

NOTE 2 This parameter will be characterized or guaranteed by design at the highest operating data rate.

NOTE 3 The input stimulus for tWCK2CA will be consistent over voltage and temp conditions.

NOTE 4 The matching error and offset of the oscillator came from CA oscillator.

NOTE 5 These parameters are defined per channel.

NOTE 6 Both min and max values must be provided. The Host may calculate the total matching error by, for example, tCAOSCmatch(T)= max(tCAOSCooffset(T)) - min(tCAOSCoffset(T))

NOTE 7 Temperature1 = Min device supported. Temperature2 = Max device supported.

5.5.1 tWCK2CA Offset due to Temperature and Voltage Variation

As temperature and voltage change on the GDDR7 DRAM, the WCK clock tree will shift and may require retraining. The oscillator is usually used to measure the amount of delay over a given time interval (determined by the controller), allowing the controller to compare the trained delay value to the delay value seen at a later time. The tWCK2CA offset due to temperature and voltage variation specification can be used for instances when the oscillator cannot be used to control the tWCK2CA.

Table 67 — tWCK2CA Offset Due to Temperature and Voltage Variation

Parameter	Symbol	Min	Max	Units	Notes
WCK to Command Address (CA) offset temperature variation	tWCK2CA_TEMP	-		ps/°C	
WCK to Command Address (CA) offset voltage variation	tWCK2CA_VOLT	-		ps/mV	1
NOTE 1 tWCK2CA max delay variation as a function of the DC voltage variation for VDDC. It includes VDDC AC noise impact for frequencies TBD MHz and max voltage of TBD Vpk-pk from DC TBD MHz at a fixed temperature on the package.					

5.5.2 CAOSC_Run and CAOSC_Count Registers

The CAOSC Mode Register and CAOSC COUNT Info Read registers are associated with the CA WCK Interval Oscillator in the GDDR7 DRAM.

Table 68 — CAOSC Mode Register

Function	Operand	Data	Notes
CAOSC	OP[x]	Start/Stop CAOSC 0_B = Stop CAOSC (default) 1_B = Start CAOSC	1, 2
NOTE 1 CAOSC COUNT value is not valid while CAOSC is in running state (after CA OSC Start and before Stop). NOTE 2 When CAOSC is set to 1 from 0, the CAOSC counter always start from 0, even in the case that info read command to readout CAOSC counter value was not issued in the previous CAOSC operation cycle.			

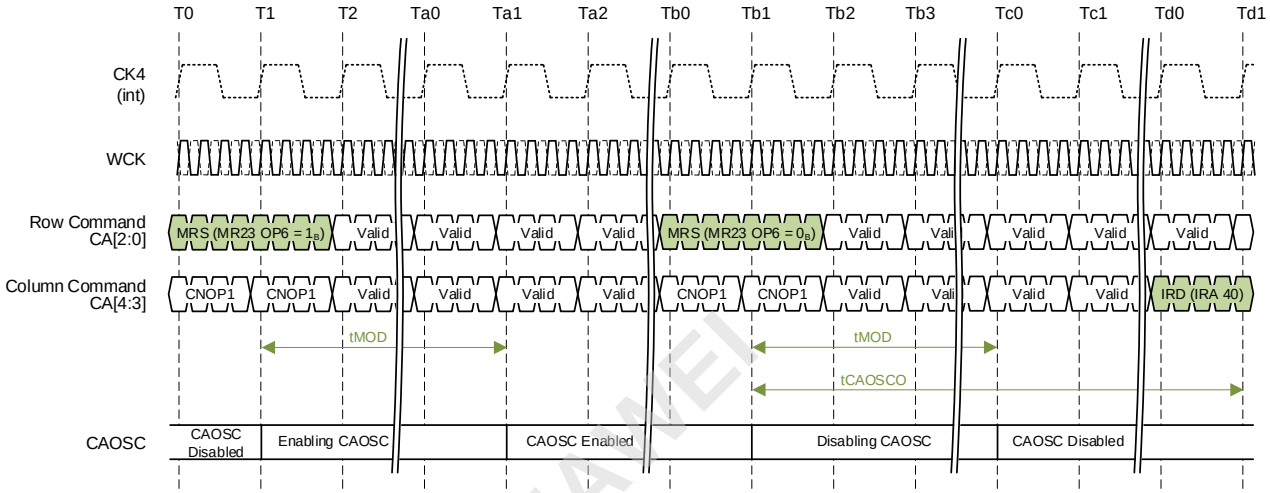
Table 69 — Info Read Data

IRA	Data	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0	Notes
40	CAOSC COUNT 1	CAOSC_COUNT_VALUE [7:0]								1, 2
41	CAOSC COUNT 2	CAOSC_COUNT_VALUE [15:8]								1, 2
42	CAOSC COUNT 3	CAOSC_ COUNT_ VALID	CAOSC_COUNT_VALUE [22:16]							1, 2
43-47	RFU	RFU								
NOTE 1 The CAOSC count value is used to compensate for VT drift on Command Address (CA) lanes to maximize CA timing margin. The count can be used by the memory controller to periodically adjust the phase of WCK relative to CA data. NOTE 2 The contents of bits CAOSC_COUNT_VALUE [22:0] are reset by starting the oscillator.										

5.5.2 CAOSC_Run and CAOSC_Count Registers (cont'd)

Table 70 — CAOSC (WCK2CA) Interval Oscillator AC Timing

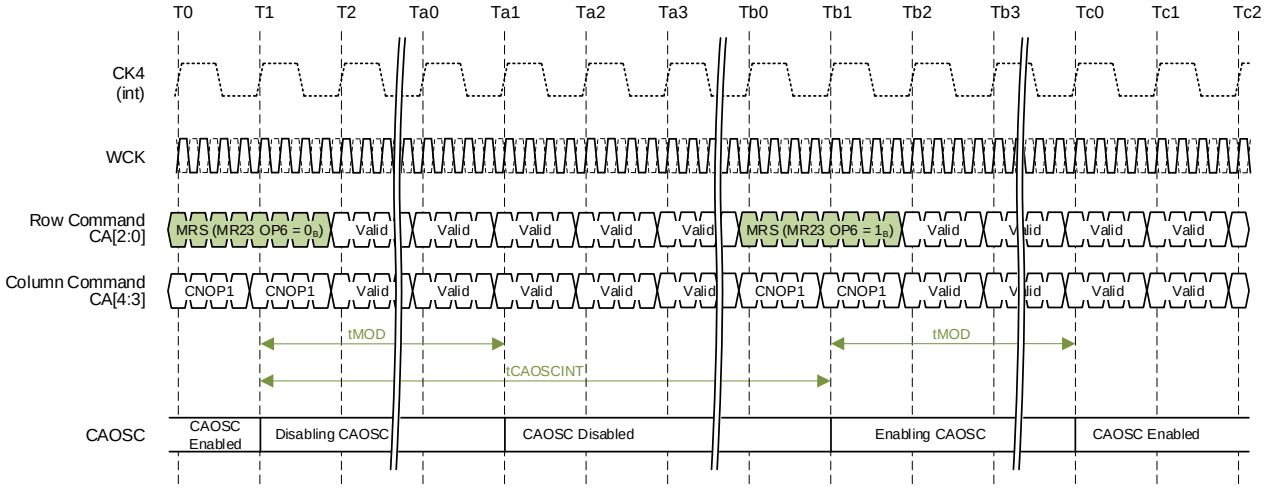
Parameter	Symbol	Min	Max	Units
Delay time from Stop CAOSC MRS to CAOSC_COUT_VALUE IRD	t_{CAOSCO}	Max(40ns, 8nCK4)	-	ns
Delay time from Stop CAOSC MRS command to Start CAOSC MRS command	$t_{CAOSCINT}$	Max(40ns, 8nCK4)	-	ns



NOTE 1 IRA 40 is shown as an example. The host must read out IRA [42:40] to read out the full CAOSC_COUNT_VALUE

NOTE 2 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 54 — CAOSC Stop to IRD (tCAOSCO)



NOTE 1 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 55 — CAOSC Stop to CAOSC Start (tCAOSCINT)

5.6 Duty Cycle Corrector (DCC)

As GDDR7 SGRAMs do not include a PLL/DLL, the use of an optional Duty Cycle Corrector (DCC) can correct for the duty cycle error of the WCK clocks, resulting in improved timing margins for all WCK-to-DQ related timings. Since the WCK is shared between DQ, RCK, and CA, RCK duty cycle and WCK-to-CA timings can be impacted by the DCC. The DCC can be enabled at any time before the interface data training process. Best results for interface data training cannot be guaranteed unless duty cycle correction has been completed prior to finalizing the relevant training steps. Read, Write and CA related trainings may need to be rerun after running the DCC to maximize timing margins.

DCC operation is controlled by MR8 OP[6:5]. The duty cycle correction is started by setting bit OP5. The DCC must be held in this state for a minimum duration of tDCC. During tDCC, MRS commands may be issued to speed up the training process, however the issuing of commands during tDCC carries the risk of impacting the DCC operation.

After tDCC is met, bit OP6 shall be set to terminate the duty cycle correction and hold the correction code. The DCC may be disabled at any time by resetting bits OP6 and OP5. *TABLE 71* shows the DCC timing parameter and *TABLE 72* the control signals.

When in the “hold” state, i.e., MR8 OP[6:5] is set to ‘11”, the DCC correction code will be valid until either the device is taken out of the “hold” state by programming MR8 OP[6:5] to ‘00’ or ‘01’ or a reset of the device. The correction code is valid for all frequencies less than or equal to the WCK frequency at which the DCC correction code was determined. The optimal DCC correction is only guaranteed at the temperature and voltage at which the DCC code was determined.

Table 71 — DCC Timing

Parameter	Symbol	Min	Max	Unit
Duty cycle corrector (DCC) start to hold MRS command delay	tDCC		-	nCK4 (or ns)

Table 72 — DCC Control Signals

MR8 OP6	MR8 OP5	DCC
0	0	DCC off
0	1	DCC start
1	0	Reserved
1	1	DCC hold

6 Operation

6.1 Row And Column Commands

GDDR7 introduces a new command protocol that allows row and column commands to be issued in parallel, provided the conditions for issuing the command and the respective timings are met. CA[2:0] inputs are associated with row commands, and CA[4:3] inputs are associated with column commands. A few commands are defined across all CA[4:0] inputs. The command truth table also includes restrictions for certain row commands that must be paired with CNOP1 or CNOP2 commands on the column bus. Refer to the *COMMAND TRUTH TABLE* for details.

The GDDR7 command/address (CA) interface supports CA parity (CAPAR) and CA bus inversion (CABI) as shown in *TABLE 73*:

- The CABI bit is received on the CA1 input and in subsequent figures shown as part of row commands.
- The CAPAR bit is received on the CA4 input and in subsequent figures shown as part of column commands.

However, CA parity (CAPAR) and CA bus inversion (CABI) are calculated and checked over all CA[4:0] inputs combined when these features are enabled in the respective Mode Registers.

Table 73 — CABI and CAPAR Bit Positions

<table>
<tr><th>Clock Cycle</th><th>CA0</th><th>CA1</th><th>CA2</th><th>CA3</th><th>CA4</th></tr>
<tr><td>0</td><td rowspan="4">Row Command
Address 11-bit</td><td>CABI</td><td rowspan="4"></td><td rowspan="3">Column Command
Address 7-bit</td><td rowspan="4"></td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td>CAPAR</td></tr>
</table>

6.2 Command Truth Table

Table 74 — Truth Table – Commands

FUNCTION	SYMBOL	WCK Clock Cycle	CA0	CA1	CA2	CA3	CA4 [10]	NOTES
Row No Operation 1	**RNOP1**	0	H	CABI	H	Valid Column Command		1, 2, 3
		1	H	H	H			
		2	V	V	V			
		3	V	V	V			
Row No Operation 2	**RNOP2**	0	H	H	H	CNOP2		1, 3
		1	H	H	H			
		2	H	H	H			
		3	H	H	H			
Column No Operation 1	**CNOP1**	0	Valid Row Command			H	H	1, 3
		1				H	H	
		2				V	V	
		3				V	CAPAR	
Column No Operation 2	**CNOP2**	0	RNOP2			H	H	1, 3
		1				H	H	
		2				H	H	
		3				H	H	
Activate (Select bank and activate row)	**ACT**	0	L	CABI	V (R16)	Valid Column Command(s)		1, 2, 4
		1	R13	V / (R14)	V / (R15)			
		2	BA2	BA3	R12			
		3	BA0	BA1	R11			
		4	R9	CABI	R10			
		5	R6	R7	R8			
		6	R3	R4	R5			
		7	R0	R1	R2			
Mode Register Set	**MRS**	0	H	CABI	H	CNOP1		1, 2, 5
		1	L	H	M5			
		2	M2	M3	M4			
		3	M0	M1	OP11			
		4	OP9	CABI	OP10			
		5	OP6	OP7	OP8			
		6	OP3	OP4	OP5			
		7	OP0	OP1	OP2			
Load FIFO	**LDFF**	0	H	CABI	H	CNOP1		1, 2, 6
		1	H	L	L			
		2	B2	B3	B4			
		3	B0	B1	V			
		4	D9	CABI	D10			
		5	D6	D7	D8			
		6	D3	D4	D5			
		7	D0	D1	D2			

Table 74 — Truth Table – Commands (cont'd)

FUNCTION	SYMBOL	WCK Clock Cycle	CA0	CA1	CA2	CA3	CA4 [10]	NOTES
Info Read	**IRD**	0	Valid Row Command(s)			H	H	1, 2
		1				L	L	
		2				V	V	
		3				V	CAPAR	
		4				OP5	OP6	
		5				OP3	OP4	
		6				OP1	OP2	
		7				OP0	CAPAR	
Read Training	**RDTR**	0	Valid Row Command(s)			H	H	1, 2
		1				L	H	
		2				V	V	
		3				L	CAPAR	
		4				V	V	
		5				V	V	
		6				V	V	
		7				V	CAPAR	
Read the Write Training Error Counter	**RDWTEC**	0	Valid Row Command(s)			H	H	1, 2
		1				L	H	
		2				V	V	
		3				H	CAPAR	
		4				V	V	
		5				V	V	
		6				V	V	
		7				V	CAPAR	
Write Training	**WRTR**	0	Valid Row Command(s)			H	H	1, 2
		1				H	L	
		2				H	V	
		3				H	CAPAR	
		4				V	V	
		5				V	V	
		6				V	V	
		7				V	CAPAR	
RCK Start	**RCKSTRT**	0	Valid Row Command			H	H	1, 2
		1				H	L	
		2				L	V	
		3				L	CAPAR	
		4				V	V	
		5				V	V	
		6				V	V	
		7				V	CAPAR	

Table 74 — Truth Table – Commands (cont'd)

FUNCTION	SYMBOL	WCK Clock Cycle	CA0	CA1	CA2	CA3	CA4 [10]	NOTES
RCK Stop	**RCKSTOP**	0	Valid Row Command			H	H	1, 2
		1				H	L	
		2				H	V	
		3				L	CAPAR	
		4				V	V	
		5				V	V	
		6				V	V	
		7				V	CAPAR	
Read (Select bank, column, and start burst)	**RD**	0	Valid Row Command(s)			L	H	1, 2, 7
		1				BA2	BA3	
		2				BA0	BA1	
		3				H	CAPAR	
		4				C5	V (C6)	
		5				C3	C4	
		6				C1	C2	
		7				C0	CAPAR	
Read with Autoprecharge	**RDA**	0	Valid Row Command(s)			L	H	1, 2, 7
		1				BA2	BA3	
		2				BA0	BA1	
		3				L	CAPAR	
		4				C5	V (C6)	
		5				C3	C4	
		6				C1	C2	
		7				C0	CAPAR	
Write (Select bank, column, and start burst)	**WR**	0	Valid Row Command(s)			H	L	1, 2, 7
		1				BA2	BA3	
		2				BA0	BA1	
		3				H	CAPAR	
		4				C5	V (C6)	
		5				C3	C4	
		6				C1	C2	
		7				C0	CAPAR	
Write with Autoprecharge	**WRA**	0	Valid Row Command(s)			H	L	1, 2, 7
		1				BA2	BA3	
		2				BA0	BA1	
		3				L	CAPAR	
		4				C5	V (C6)	
		5				C3	C4	
		6				C1	C2	
		7				C0	CAPAR	

Table 74 — Truth Table – Commands (cont'd)

FUNCTION	SYMBOL	WCK Clock Cycle	CA0	CA1	CA2	CA3	CA4 [10]	NOTES
Precharge Per-Bank (Deactivate row in a bank)	**PREpb**	0	H	CABI	H	Valid Column Command		1, 2, 8
		1	L	L	H			
		2	BA2	BA3	DRFM			
		3	BA0	BA1	V			
Precharge All-Bank (Deactivate row in all banks)	**PREab**	0	H	CABI	H	Valid Column Command		1, 2
		1	L	L	L			
		2	V	V	V			
		3	V	V	V			
Refresh Per-Bank	**REFpb**	0	H	CABI	L	Valid Column Command		1, 2, 8
		1	L	H	H			
		2	BA2	BA3	V			
		3	BA0	BA1	V			
Refresh All-Bank	**REFab**	0	H	CABI	L	Valid Column Command		1, 2
		1	L	H	L			
		2	V	V	V			
		3	V	TR	V			
Refresh Management Per-Bank	**RFMpb**	0	H	CABI	L	Valid Column Command		1, 2, 8
		1	H	L	H			
		2	BA2	BA3	DRFM			
		3	BA0	BA1	V			
Refresh Management All-Bank	**RFMab**	0	H	CABI	L	Valid Column Command		1, 2
		1	H	L	L			
		2	V	V	V			
		3	V	V	V			
Power-Down Entry	**PDE**	0	H	H	L	CNOP1		1, 9
		1	L	L	L			
		2	L	H	H			
		3	H	H	H			
Sleep Entry	**SLE**	0	H	H	L	CNOP1		1, 9
		1	L	L	L			
		2	H	H	H			
		3	H	H	H			
Self Refresh Entry	**SRE**	0	H	H	L	CNOP1		1, 9
		1	L	L	L			
		2	L	L	H			
		3	H	TR	H			
Self Refresh Sleep Entry	**SRSE**	0	H	H	L	CNOP1		1, 9
		1	L	L	L			
		2	H	L	H			
		3	H	H	H			

Table 74 — Truth Table – Commands (cont'd)

FUNCTION	SYMBOL	WCK Clock Cycle	CA0	CA1	CA2	CA3	CA4 [10]	NOTES
Power-Down Exit / Self Refresh Exit	**PDX / SRX**	0	H	H	L	CNOP1		1, 9
		1	L	L	L			
		2	H	H	L			
		3	H	H	L			
CA Training Entry	**CATE**	0	H	H	L	CNOP1		1, 9
		1	L	L	H			
		2	V	L	L			
		3	V	V	V			
Command Start Point	**CSP**	0	H	L	H	L	H	1
		1	H	H	H	H	H	
		2	H	H	H	H	H	
		3	H	H	H	H	H	
CA Training Exit	**CATX**	n	L	L	L	L	L	1, 11
		n+1	L	L	L	L	L	
		n+2	L	L	L	L	L	
		n+3	L	L	L	L	L	
		n+4	L	L	L	L	L	
		n+5	L	L	L	L	L	
		n+6	L	L	L	L	L	
		n+7	L	L	L	L	L	
		n+8	L	L	L	L	L	
		n+9	L	L	L	L	L	
		n+10	L	L	L	L	L	
		n+11	L	L	L	L	L	
		n+12	L	L	L	L	L	
		n+13	L	L	L	L	L	
		n+14	L	L	L	L	L	
		n+15	L	L	L	L	L	

NOTE 1 H = Logic High Level; L = Logic Low Level; V = Valid, signal may be H or L, but not floating.

NOTE 2 Values shown for CA[4:0] are logical values; the physical values are inverted when Command/Address Bus Inversion (CABI) is enabled and CABI = L.

NOTE 3 Both encodings perform the same NOP. NOP2 encodings are used when the state of the CA bus is required to be High.

NOTE 4 BA[3:0] provide the bank address, R[(16), (15), (14), 13:0] provide the row address. R14 is only used on 16 Gbit 2 channel mode and 24/32 Gbit 4 channel mode. R15 is only used on the 24/32 Gbit 2 channel mode. R16 is reserved for future use. See *ADDRESSING* section for more details.

NOTE 5 M[5:0] provide the Mode Register address (MRA), OP[11:0] the opcode to be loaded.

NOTE 6 B[4:0] select the burst position, and D[10:0] provide the data.

NOTE 7 BA[3:0] provide the bank address, C[(6), 5:0] provide the column address; no sub-word addressing within a burst of 16 symbols. C6 is reserved for future use. See *ADDRESSING* section for more details.

NOTE 8 BA[3:0] provide the bank address.

NOTE 9 CNOP1 required on the column bus (CA[4:3]).

NOTE 10 CABI (Command Address Bus Inversion) and CAPAR (CA Parity) are calculated on CA[4:0].

NOTE 11 CATX is 16 consecutive WCK cycles of L (n to n+15) on all CA inputs as the internal CK4 alignment is not guaranteed until CSP command issued after exiting CA Training. See the *COMMAND ADDRESS BUS TRAINING* section for more details.

6.3 NOP

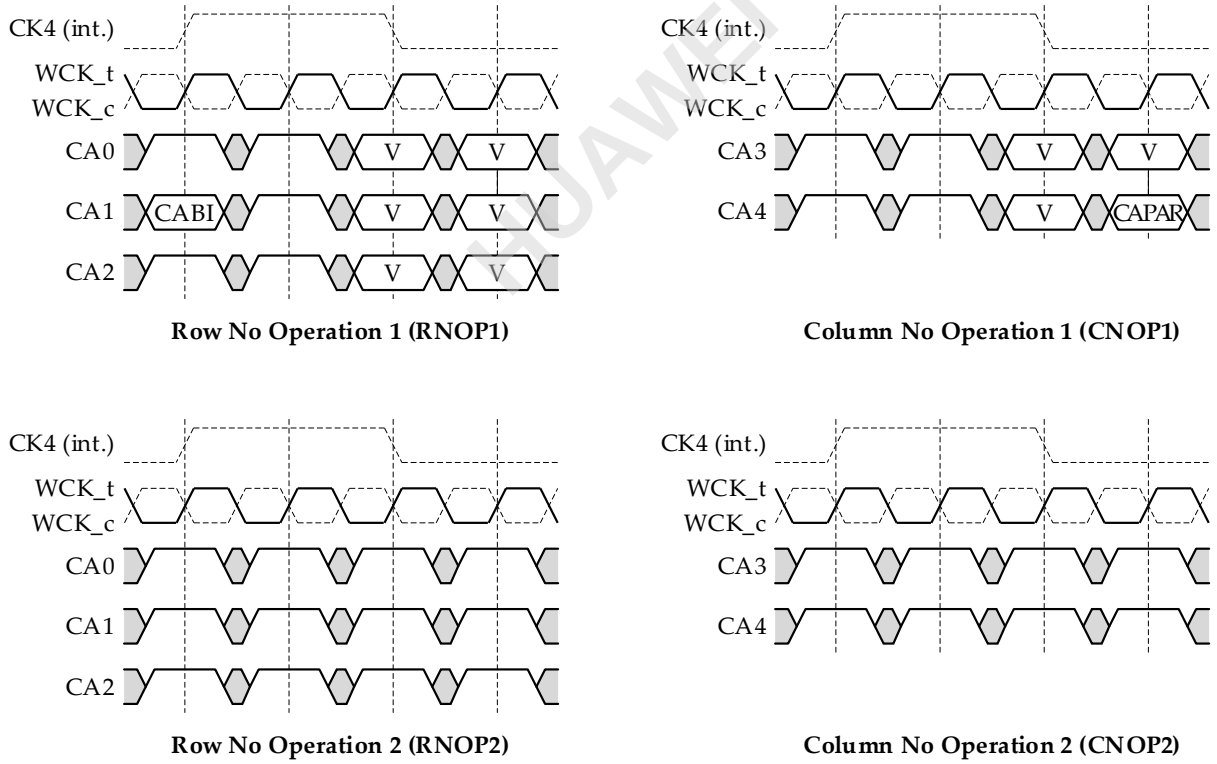
The Row No Operation 1 (RNOP1) command is used to instruct a channel to perform a NOP as row command; this prevents unwanted row commands from being registered during idle or wait states. Operations already in progress are not affected.

The Column No Operation 1 (CNOP1) command is used to instruct a channel to perform a NOP as column command; this prevents unwanted column commands from being registered during idle or wait states. Operations already in progress are not affected.

During certain state transitions, Row No Operation 2 (RNOP2) and Column No Operation 2 (CNOP2) commands are required as pairs instead of RNOP1 and CNOP1. These state transitions include

- Exit from CA bus training;
- Transitions from asynchronous to synchronous operation that require a CSP command;
- Power-Down entry and exit;
- Self Refresh entry and exit;
- Sleep mode entry and exit.

CA parity is evaluated with the RNOP1 and CNOP1 commands when the parity calculation is enabled in MR15 OP0. RNOP2 and CNOP2 commands are defined with all inputs driven High. This encoding meets the condition for (even) CA parity by definition.



NOTE 1 CABI = Command/Address Bus Inversion; CAPAR = CA Parity; V = Valid (H or L but not floating).

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 56 — Row and Column No Operation Commands

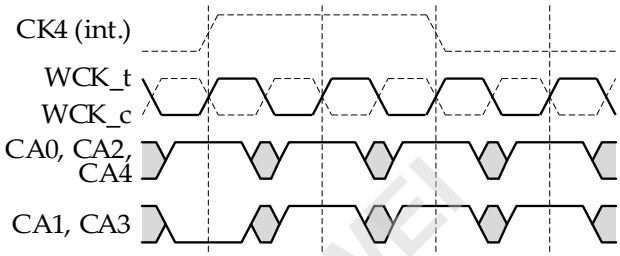
6.4 Command Start Point (CSP)

The Command Start Point (CSP) command is provided to align the DRAM's internal CK4 clock with the 4 U.I. command burst boundary as assumed by the host. The command is required to be issued exactly once in the following state transitions to normal operation (see the *SIMPLIFIED STATE DIAGRAM*):

- Exit from Sleep mode or Self Refresh Sleep mode (SLX), where the DRAM's WCK receiver was disabled and the DRAM's internal CK4 clock had lost its synchronization with the host.
- Exit from CA bus training (CATX) (see *COMMAND ADDRESS BUS TRAINING*).

Issuing a CSP command in all other cases is illegal and may lead to unspecified operation.

The CSP command encoding (see *FIGURE 57*) is like a combination of RNOP2 and CNOP2 except for the two Low bits on the first U.I. of the command on CA1 and CA3 inputs. RNOP2 and CNOP2 commands are required in the tCSP_PRE or tCATX period prior to a CSP command, and in the tCSP_POST period following a CSP command. This makes the two Low bits to stick out of a series of command cycles with all CA inputs constantly driven High.



NOTE 1 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 57 — Command Start Point Command

FIGURE 58 illustrates how the CSP command will allow the DRAM to align its internal CK4 clock with the host and be able to correctly decode subsequent access commands. Depending on the initial state of the internal CK4 clock there will be 4 cases with the two Low bits sampled either on the first, second, third or fourth U.I. of a CK4 cycle.

In case 1, the rising edge of the internal CK4 clock occurs aligned with the command boundary as assumed by the host. The CSP command will be correctly sampled as **L**HHH which indicates alignment between host and DRAM. No change to the internal CK4 clock is required following the CSP command.

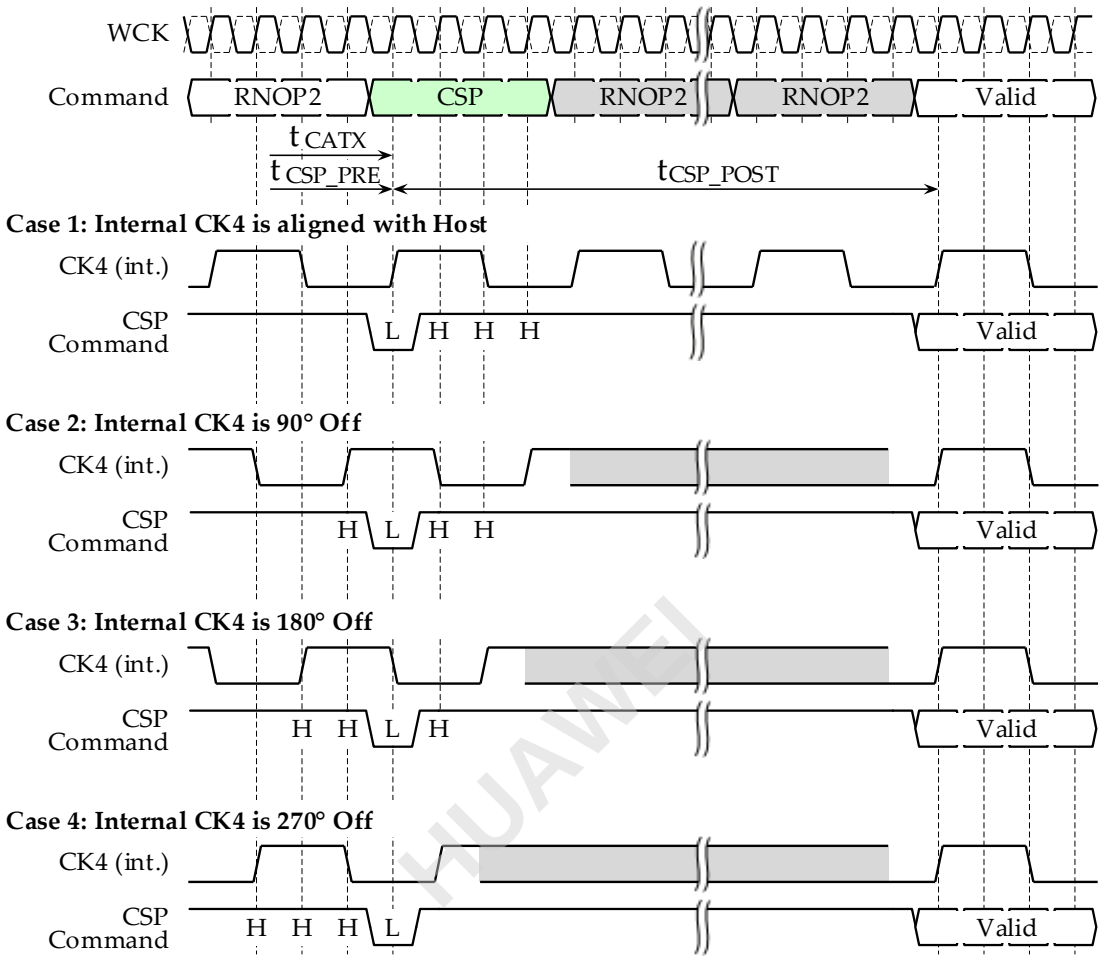
In cases 2, 3, and 4, the rising edge of the internal CK4 clock is shifted by 90°, 180° or 270° with respect to the command boundary as assumed by the host. The CSP command will be sampled as H**L**HH, HH**L**H or HHH**L**, respectively. The DRAM will use this information to correct the phase of the internal CK4 clock during the tCSP_POST period.

The channel is ready to receive valid commands once the tCSP_POST timing has elapsed.

CA parity is evaluated with the CSP command when the parity calculation is enabled in MR15 OP0. CA parity calculation may temporarily be suspended in the tCSP_POST period.

When enabled in MR0 OP0, CA bus inversion is suspended in device states that require a CSP command (e.g., chip reset, Sleep modes, CA bus training), and will be resumed within the tCSP_POST period following the CSP command. See the *COMMAND ADDRESS BUS INVERSION (CABI)* section for details on CABI.

6.4 Command Start Point (CSP) (cont'd)

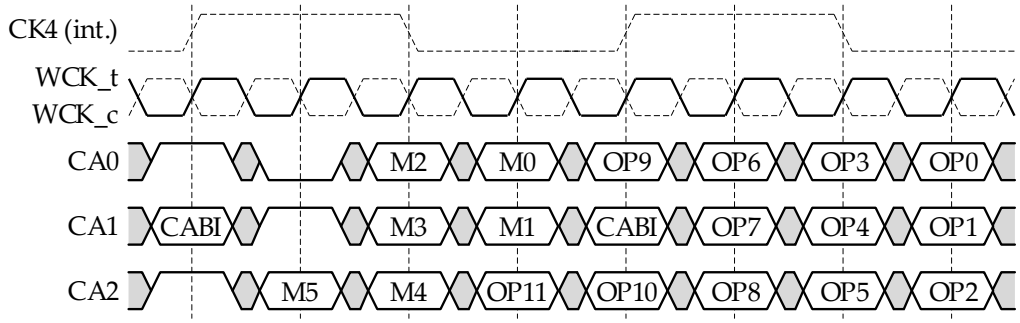


NOTE 1 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 58 — CSP Command Operation

6.5 Mode Register Set (MRS)

The Mode Register Set (MRS) command is used to load the Mode Registers of the device. M[5:0] select the Mode Register, and OP[11:0] determine the opcode to be loaded. See *MODE REGISTERS* section for the register definition.



NOTE 1 M = Mode Register Address; OP = Opcode to be loaded; CABI = Command/Address Bus Inversion.

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

NOTE 3 See *MODE REGISTERS* for the register definition.

Figure 59 — Mode Register Set Command

The Mode Register Set command can be issued when the device is in bank idle state or in Self Refresh mode, and when all previous Read or Write operations have completed. For Reads, a burst completion is defined as when the last data element including CRC has been transmitted on the data outputs (tRDMRS timing); for Writes, a burst completion is defined as when the last data element has been written to the memory array and the result of the write CRC check has been returned to the host (tWRMRS timing).

The Mode Register Set command may also be issued in bank active state for specific mode registers that define data interface characteristics or control data training. Refer to the *MODE REGISTERS* section for a list of mode registers that are supported in bank active state.

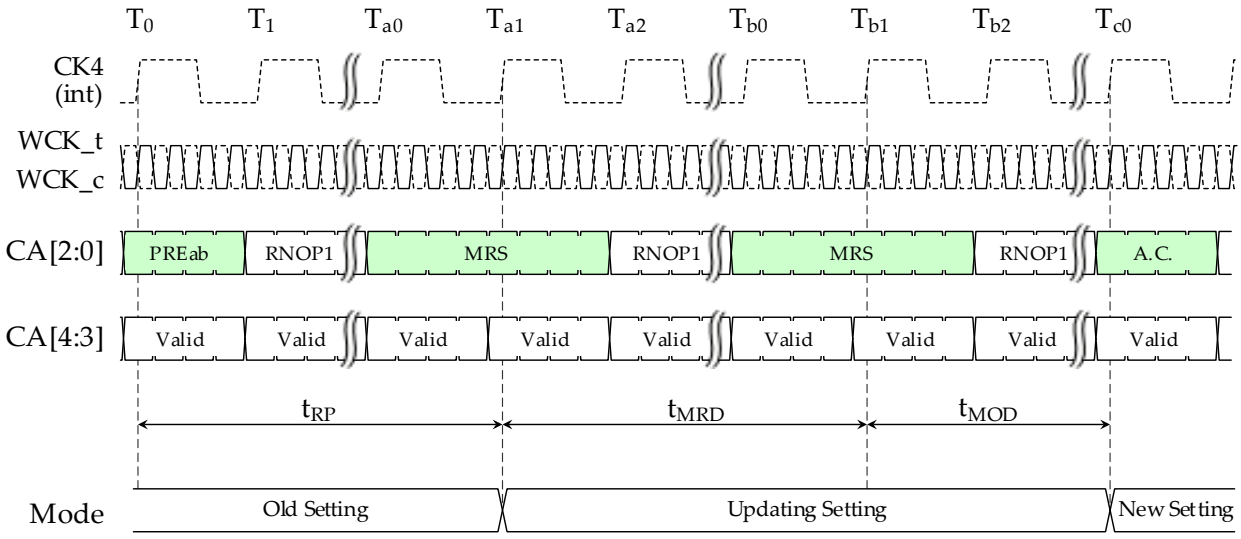
The MRS command to non-MRS command delay, tMOD, is required by the device to update the features, and is the minimum time required from an MRS command to a non-MRS command excluding RNOP and CNOP. The MRS command cycle time, tMRD, is required to complete the write operation to the Mode Register, and is the minimum time required between two MRS commands.

The value of tMRD is less than or equal to tMOD, and the vendor's datasheet should be consulted for details.

The actual Mode Register update is initiated with the second CK4 cycle of the MRS command. All relevant timing parameters therefore refer to this second rising CK4 clock edge as shown in subsequent timing diagrams.

CA parity is evaluated with the Mode Register Set command when CA parity has already been enabled in MR15 OP0 prior to this Mode Register Set command. When CA parity is enabled by a Mode Register Set command, the channel requires all subsequent commands including NOP commands to be issued with correct parity until tMOD has expired for the Mode Register Set command that disables CA parity.

6.5 Mode Register Set (MRS) (cont'd)



NOTE 1 A Precharge command as shown in the figure shall not be issued when the channel is in Self Refresh state.

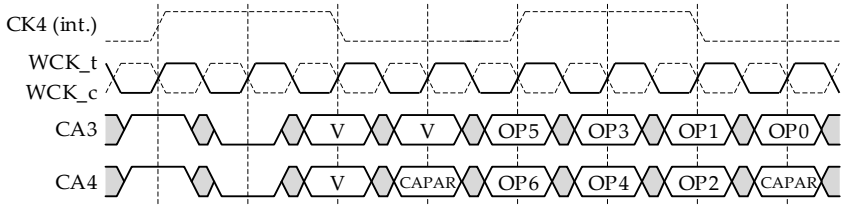
NOTE 2 A.C. = Any row command allowed in bank idle or Self Refresh state.

NOTE 3 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 60 — Mode Register Set Timings

6.6 Info Read

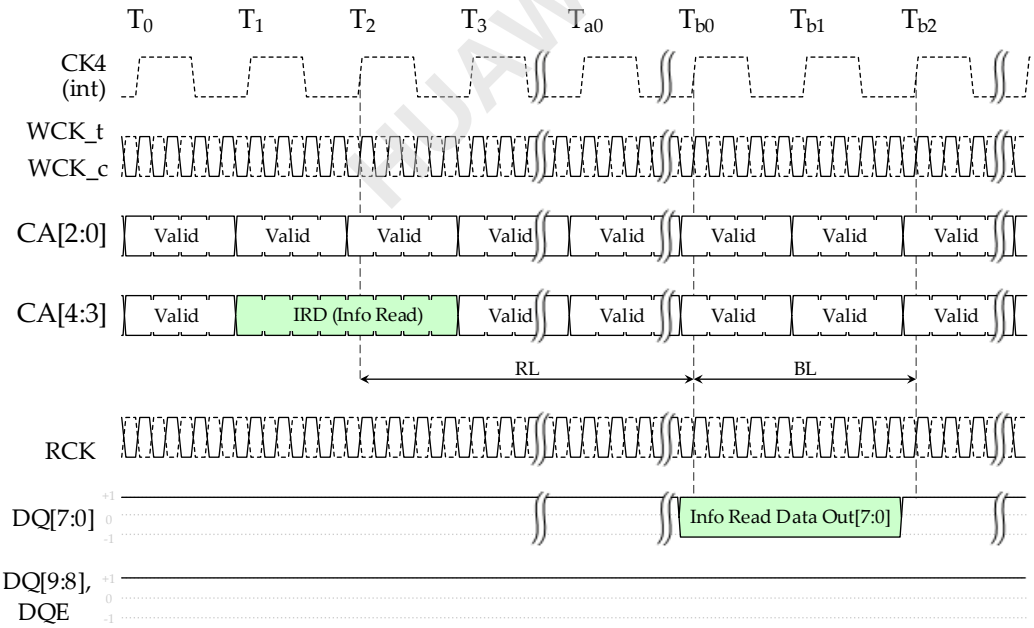
GDDR7 SGRAMs Info Read (IRD) command provides access to a variety of information about the device operation and status. The IRD data readout cycle is initiated with the IRD command as shown in *FIGURE 61*, where the OP[6:0] bits select one of 128 Info Register Addresses (IRA), as summarized in *TABLE 75*, with 32 IRA reserved for vendor specific data.



NOTE 1 OP = Info Read Register Address (IRA); CAPAR = CA Parity; V = Valid (H or L but not floating).

Figure 61 — Info Read Command (IRD)

The timing of the IRD data return matches that of a normal Read operation; see *FIGURE 62* for an overview of the IRD timing diagram. The DRAM will not send CRC data back to host regardless of whether RDCRC is enabled or not. During IRD readout DQ[9:8] and DQE will drive +1 level in PAM3 mode. In NRZ mode, DQ[9:8] are High-Z and DQE will drive either High-Z or H level depending on the MR5 OP10 bit (if DQE High-Z State is supported).



NOTE 1 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

NOTE 2 The shown RCK is just for illustration, IRD will support the same RCK protocol as in normal read operations, refer to *READ CLOCK (RCK)* Section for details.

Figure 62 — Info Read Operation

6.6 Info Read (cont'd)

Each IRD command will drive out 8 bits of data, one bit per DQ in DQ[7:0] in a static High/Low level encoding scheme (+1/-1 in PAM3 mode or H/L in NRZ mode). The data will be available on the DQ[7:0] after the read latency RL, and it will be held for the duration of one read burst BL, i.e., 16 U.I. in PAM3 mode or 32 U.I. in NRZ mode.

Table 75 — Info Read Data

IRA	Data	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0	Notes
0	Vendor ID1, see *TABLE 77*	Revision ID				Manufacturer ID				
1	Vendor ID2,see *TABLE 78*	PAM3 FIFO Depth		NRZ FIFO Depth	UE Count Support	UE COL Support	Density			2
2	Vendor ID3, see *TABLE 79*	ERR Driver Offset	CTLE Rx Sub-Add.	DQE_HZ	DCC	SCRAMBLE	ECC_ENG _CE	DFE Rx Sub-Add	SEV2ERR	
3	Vendor ID4, See *TABLE 80*	RFU		HALF_CK4_ ERR_TRAIN	ECC_ENG _2CH	CAPARBLK_LAT_TYPE		CSP feedback	BRC Support	
4	Refresh Management	ARFM	RFM	RAADEC	RAAMMT		RAAIMT			
5	Adaptive Refresh Management	RFU		RAADEC_A	RAAMMT_A		RAAIMT_A			
6		RFU		RAADEC_B	RAAMMT_B		RAAIMT_B			
7		RFU		RAADEC_C	RAAMMT_C		RAAIMT_C			
8-15	RFU	RFU								
16	Temp., ECS Error flag, see *TABLE 81*	ECS Error	Temperature							
17	Max. Temp. see *TABLE 82*	RFU	Max. Temperature Log							
18	ECS Error Log 1	Max CE per ROW Address BA[3:0]				UE Address BA[3:0]				2
19	ECS Error Log 2	Max CE per Row Address R[15:8]								
20	ECS Error Log 3	Max CE per Row Address R[7:0]								
21	ECS Error Log 4	UE Address R[15:8]								
22	ECS Error Log 5	UE Address R[7:0]								
23	ECS Error Log 6	RFU	UE Address C[6:0] or {0, C[5:0]}							
24	ECS Error Log 7	ECS CE FLAG	ECS UE FLAG	ECS UE COUNT FLAG	RFU	UE Count [3:0]				

Table 75 — Info Read Data (cont'd)

IRA	Data	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0	Notes
25	hPPR Resources 1, see *TABLE 83*	Bank 7	Bank 6	Bank 5	Bank 4	Bank 3	Bank 2	Bank 1	Bank 0	
26	hPPR Resources 2, see *TABLE 83*	Bank 15	Bank 14	Bank 13	Bank 12	Bank 11	Bank 10	Bank 9	Bank 8	
27-31	RFU	RFU								
32	Serial ID, see *TABLE 84*	Serial ID-1								
33		Serial ID-2								
34		Serial ID-3								
35		Serial ID-4								
36		Serial ID-5								
37		Serial ID-6								
38		Serial ID-7								
39		Serial ID-8								
40	CAOSC COUNT 1	CAOSC_COUNT_VALUE [7:0]								3
41	CAOSC COUNT 2	CAOSC_COUNT_VALUE [15:8]								3
42	CAOSC COUNT 3	CAOSC_COUNT_VALID	CAOSC_COUNT_VALUE [22:16]							3
43-95	RFU	RFU								
96-127	Vendor Specific	Data								

NOTE 1 Default value for RFU bits is 1_B.
NOTE 2 Refer to the *ERROR CHECK AND SCRUB (AUTO ECS)* section for details about the ECS related fields.
NOTE 3 Refer to the *COMMAND ADDRESS OSCILLATOR (CAOSC)* section for details.

6.6 Info Read (cont'd)

The IRD command is allowed provided that the following conditions are met:

- Bank idle state.
- Bank active state when timing conditions in *TABLE 76* are met.
- Refresh All-Bank when the TR bit is set with the REFab command and the timing conditions in *TABLE 76* are met.
- Self Refresh mode when the TR bit is set with the SRE command. Please refer to the *SELF REFRESH* section for details.

Table 76 — Info Read Timings[1]

From Command	To Command	Minimum Delay Between "From Command" to "To Command"	Unit	Notes
RD and RDA	IRD	tRDRTR	nCK4	
RDTR and RDWTEC	IRD	tRTRRTR	nCK4	
WR and WRA	IRD	tWRRTR	nCK4	
WRTR	IRD	tWTRTR	nCK4	
IRD	RD and RDA	tRTRRD	nCK4	
IRD	RDTR and RDWTEC	tRTRRTR	nCK4	
IRD	WR, WRA and WRTR	tIRD2WR	nCK4	
IRD	IRD	tIRD2IRD	nCK4	
IRD Info Register Address x	IRD Info Register Address x	tIRD2IRDSA	nCK4	2
RCKSTRT	IRD	tRCKSTRT2RD	nCK4	3
IRD	RCKSTOP	tRD2RCKSTOP	nCK4	4
RCKSTOP	IRD	tRCKSP2ST	nCK4	4

NOTE 1 This table summarizes the labels for the command-to-command delays related to Info Read, for the specific values please check the — AC Timings section.

NOTE 2 The tIRD2IRDSA delay for back-to-back IRD commands to the same Info Register Address (IRA), permits to keep the data steady at the output for longer than a burst. Back-to-back IRD commands to the same Info Register Address (IRA) are only supported for IRAs 0 to 7 and 32 to 39.

NOTE 3 This delay applies when RCKMODE is set to Start with RCKSTRT command (MR9 OP[1:0]=10_B).

NOTE 4 This delay applies when RCKMODE is set to Start with Read (i.e., IRD) command (MR9 OP[1:0]=01_B) .

Info Read operation can be carried out during normal device operation with a limited impact in performance. *FIGURE 63* shows that the minimum delay between two consecutive reads with an IRD readout in between is tRD2IRD + tIRD2IRD.

6.6 Info Read (cont'd)

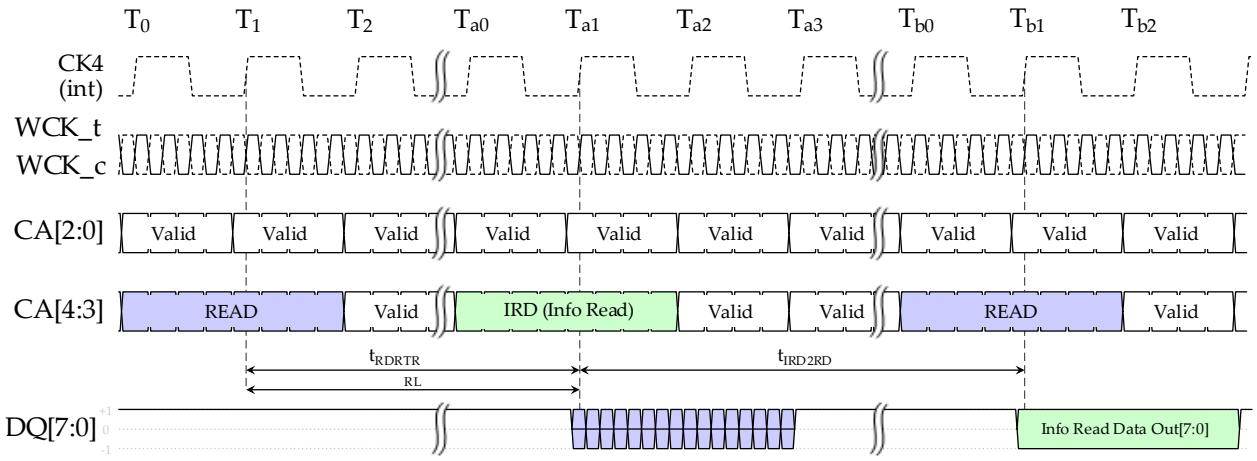


Figure 63 — Example of Info Read Operation Between Consecutive Reads

FIGURE 64 shows that the minimum delay between two consecutive writes with an IRD in between is tWRRTR + tRTRWR.

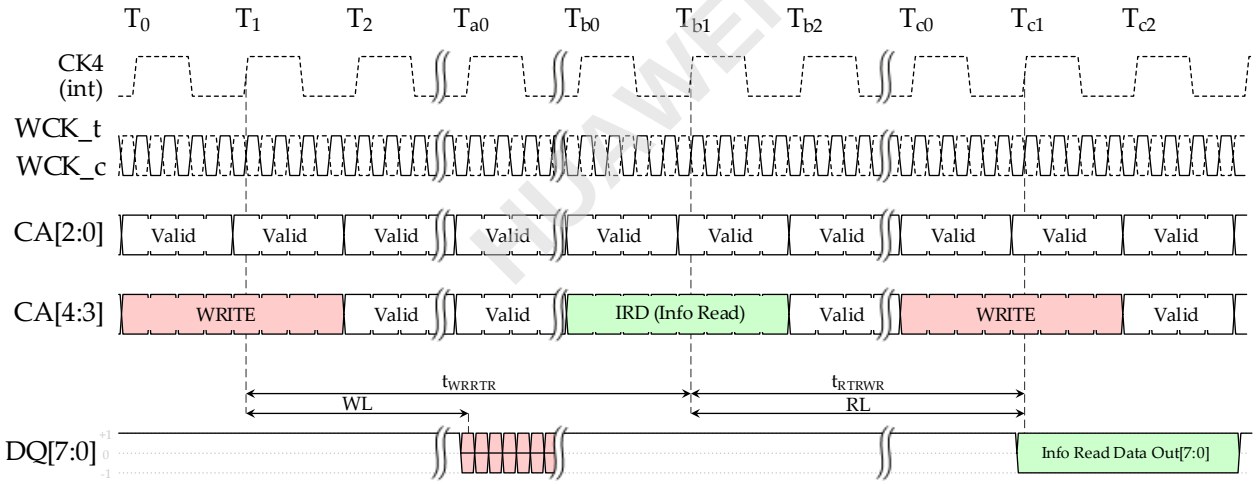


Figure 64 — Example of Info Read Operation Between Consecutive Writes

Table 77 — Info Register Address 0 – Vendor ID1

DQs	Field	Description
[7:4]	Revision ID	The device revision ID is a vendor specific assignation 0000_B 0001_B … 1111_B
[3:0]	Manufacturer ID	Manufacturer's Vendor Code and Manufacturer's Name 0001_B – Samsung 0110_B – SK Hynix 1111_B – Micron All other values are reserved.

6.6 Info Read (cont'd)

Table 78 — Info Register Address 1 – Vendor ID2

DQ	Field	Description
[7:6]	PAM3 FIFO Depth	Supported FIFO Depth in PAM3 mode 00_B – 8 Bursts 01_B – 10 Bursts 10_B – 12 Bursts 11_B – RFU
[5]	NRZ FIFO Depth	Supported FIFO Depth in NRZ mode 0_B – Same as in PAM3 mode 1_B – Half of PAM3 mode
[4]	ECS UE Count Support	ECS UE Count logging 0_B – Supported 1_B – Not Supported
[3]	ECS COL Support	ECS UE Column Address logging 0_B – Supported 1_B – Not Supported
[2:0]	Density	Memory density per device 001_B – 16 Gb 010_B – 24 Gb 011_B – 32 Gb 100_B – 48 Gb 101_B – 64 Gb All other values are reserved.

Table 79 — Info Register Address 2 – Vendor ID3

DQs	Field	Description
[7]	ERR Driver Offset	The ERR signal pull-up and pull-down driver impedances offsets can be adjusted (see MR27 OP[7:0]) 0_B – Supported 1_B – Not Supported
[6]	CTLE Rx Sub-Address	CTLE values can be programmed for the upper and lower data receiver separately (see MR18 OP[7:6]) 0_B – Supported 1_B – Not Supported
[5]	DQE_HZ	Set DQE signal to High-Z (see MR5 OP10) 0_B – Supported 1_B – Not Supported
[4]	DCC	Duty Cycle Corrector (see MR8 OP[6:5]) 0_B – Supported 1_B – Not Supported
[3]	SCRAMBLE	Data Scramble (see MR20) 0_B – 32-bit 1_B – 8-bit
[2]	ECC_ENG_CE	ECC Engine Test mode reports CE (refer to the ECC Test Mode section for details) 0_B – Supported 1_B – Not Supported
[1]	DFE Rx Sub-Address	DFE values can be programmed for the upper and lower data receivers separately (MR19 OP[7:6]) 0_B – Supported 1_B – Not Supported
[0]	SEV2ERR	SEV flag on the ERR signal (see MR5 OP9) 0_B – Supported 1_B – Not Supported

6.6 Info Read (cont'd)

Table 80 — Info Register Address 3 – Vendor ID4

DQs	Field	Description
[7:6]	RFU	
5	HALF_CK4_ERR_TRAIN	Half CK4 rate clock ERR signal training pattern 0_B – Supported 1_B – Not Supported
[4]	ECC_ENG_2CH	Additional ECC engine test required at 2 Channel mode 0_B – Required 1_B – Not required
[3:2]	CAPARBLK_LAT_TYPE	CAPARBLK_LAT support 00_B – Both Explicit and Implicit CAPARBLK_LAT supported 01_B – RFU 10_B – Explicit CAPARBLK_LAT only supported 11_B – Implicit CAPARBLK_LAT only supported
[1]	CSP Feedback	CSP Feedback on ERR signal 0_B – Supported 1_B – Not Supported
[0]	SEV2ERR	BRC Support Level (see MR8 OP[4:3]) 0_B – BRC 2 only (default) 1_B – BRC 2, 3, 4 Supported

6.6.1 Temperature Sensor Readout

The GDDR7 SGRAM features a temperature sensor with digital readout function. This function allows the controller to monitor the die's junction temperature and use this information to confirm that the device is operated within the specified temperature range or to adjust interface timings relative to temperature changes over time.

The temperature sensor is permanently enabled. The temperature readout uses the DRAM Info Read feature. The temperature is internally sampled with the IRD command to Info Register Address 16 and the corresponding digital value will be driven out on the DQ[7:0] after RL. The device's junction temperature is linearly encoded as shown in *TABLE 81*. The sensor's accuracy is vendor specific.

Table 81 — Info Register Address 16 – Temperature Sensor and ECS Error Flag

DQ	Field	Description	Notes
7	ECS Error Flag	The ECS Error Flag indicates if the ECS has logged an error, the flag is cleared upon a successful ECS log IRD readout. Check the ECS Section for more details. Flag Status: 0_B – No new ECS error logged 1_B – New ECS error logged	
[6:0]	Temperature Sensor	The 7 bits digital readout of the temperature sensor translates to temperature in degrees Celsius as follows: $000\ 0000_B$ 0_D -40 C $000\ 0001_B$ 1_D -38 C … $001\ 0100_B$ 20_D 0 C $001\ 0101_B$ 21_D 2 C $001\ 0110_B$ 22_D 4 C … $101\ 0000_B$ 80_D 120 C …	1
NOTE 1 Temperature sensor operation is only guaranteed for temperatures within the operating range.			

6.6.2 Maximum Temperature Log

The IRA17 logs the maximum temperature value as measured by the internal temperature sensor. The default value at device initialization, after device reset or after each Info Read to Info Register Address 17 readout operation is 0 decimal (-40°C). The stored value is compared at regular intervals with an internal readout of the temperature sensor, and it is updated if the last readout value is bigger than the stored one. The output is encoded in the same way as described for the Temperature Sensor readout in IRA16, please see *TABLE 81* for details.

Table 82 — Info Register Address 17 – Maximum Temperature Log

DQ	Field	Description
7	RFU	
[6:0]	Max. Temperature	Stores the maximum temperature value as it is internally logged at regular intervals. Same encoding as for Info Register Address 16.

6.6.3 hPPR Resources

IRA25 and 26 indicate the availability of a hPPR resource in each of the 16 banks. Refer to the *HARD POST PACKAGE REPAIR (HPPR)* section for details on how to use these IRAs.

Table 83 — Info Register Addresses 25 and 26 – hPPR Resources

DQ	Field	Description
any	hPPR Resource 1 hPPR Resource 2	Availability of a hPPR resource in a bank: 0_B – No hPPR resource available 1_B – hPPR resource available

6.6.4 Serial ID

The Info Registers Addresses 32 to 39 provide up to 64 bits for a device serial identification code. The contents, format and length of the Serial ID code is vendor specific, but all the 8 Info Registers Addresses must provide valid data output. The 8 IRA can be read in any order.

Table 84 — Info Register Addresses 32 to 39 – Serial ID

DQ	Field	Description
[7:0]	Serial ID-x	Vendor specific

6.7 Row Activation

Before any Read or Write command can be issued to a bank, a row in that bank must be opened (activated). This is accomplished by the Activate command (see *FIGURE 65*): BA[3:0] select the bank, and R[16:0], R[15:0], R[14:0] or R[13:0], depending on the channel density, select the row to be activated. Once a row is open, a Read or Write command could be issued to that row, subject to the tRCD specification.

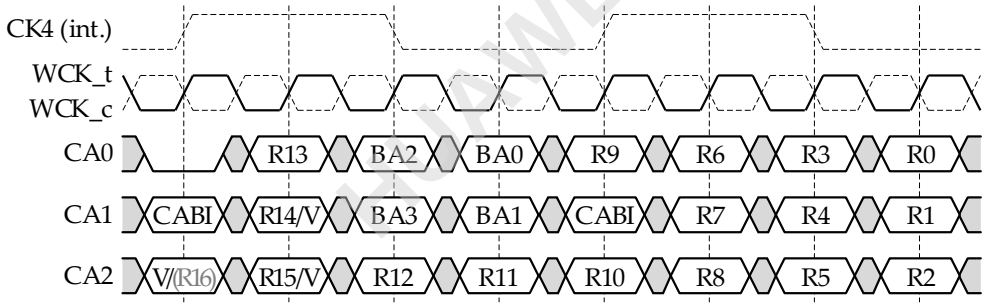
A subsequent Activate command to another row in the same bank can only be issued after the previous row has been closed (precharged). The minimum time interval between two successive Activate commands on the same bank is defined by tRC. A minimum time, tRAS, must have elapsed between opening and closing a row.

A subsequent Activate command to another bank can be issued while the first bank is being accessed, which results in a reduction of total row access overhead. The minimum time interval between two successive Activate commands on different banks is defined by tRRD. *FIGURE 66* illustrates the tRCD and tRRD definition.

The actual row activation is initiated with the second CK4 cycle of the Activate command. All relevant timing parameters therefore refer to this second rising CK4 clock edge as shown in subsequent timing diagrams.

The row remains active until a Precharge command (or Read or Write command with Auto Precharge) is issued to the bank.

CA parity is evaluated with the Activate command when the parity calculation is enabled in MR15 OP0.



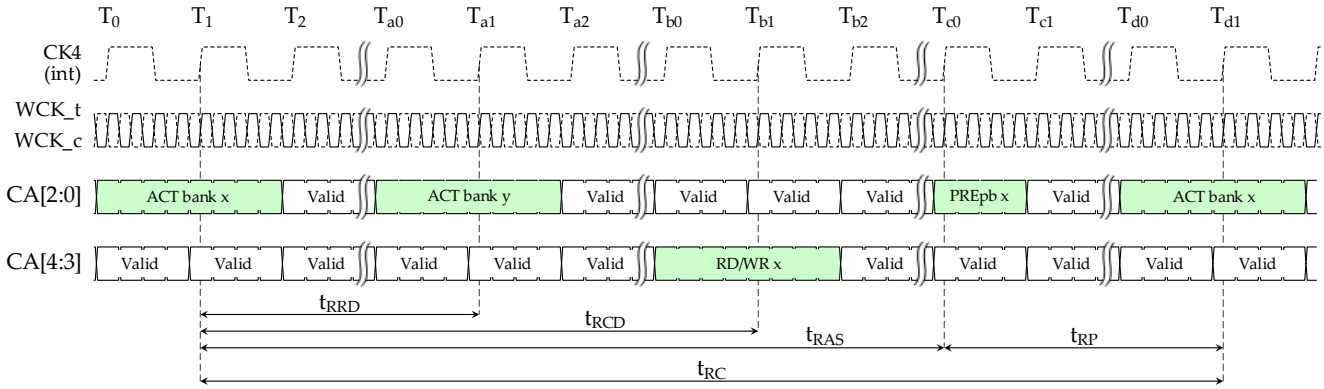
NOTE 1 BA = Bank Address; R = Row Address; CABI = Command/Address Bus Inversion; V = Valid (H or L but not floating).

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

NOTE 3 R15 and R14 are either valid row address bits or Don't Care ("V"), depending on channel density. R16 is a placeholder and not yet allocated for any density. Refer to the *ADDRESSING* table for the specific row address range.

Figure 65 — Activate Command

6.7 Row Activation (cont'd)



NOTE 1 tRCD is tRCDRD or tRCDWR depending on command. In case of a Write command, the tWR timing must be met as well before closing the bank.

Figure 66 — Row Activation Command Cycle

6.7.1 Additional Requirements Concerning a Non-Existing Row Address Range

The row address range with the MSB and MSB-1 row address bits both as 1 does not exist for channel densities of 6 Gb, 12 Gb, and 24 Gb, and ACT commands with row addresses to that address range are illegal. However, ACT commands that accidently address a row in this illegal row address range shall follow all timing and protocol rules as though the ACT were valid.

Any RD or RDA command following an ACT command to the invalid row address range shall drive the RCK with normal timing but shall not output data on the DQs that can be used to learn about data stored in cells in any bank with valid row addresses. DQ data that coincidentally matches cell array data is permissible, for example, always sending all 1s and cell data sometimes being all 1s. Consistently exposing data from a previous Read or previous Activate is not permissible.

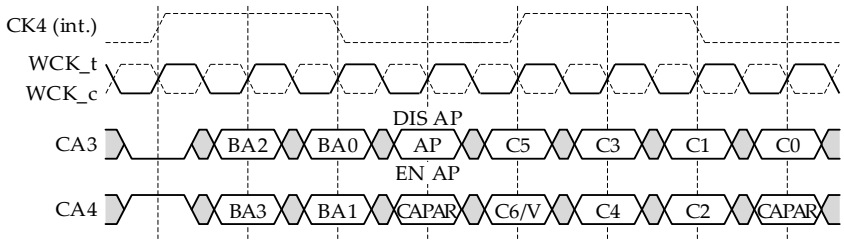
Any WR or WRA command following an ACT to the invalid row address range shall not result in new data being written anywhere within the device.

The device will operate normally for Reads and Writes to banks which have pages open for valid rows.

6.8 Write

Write bursts are initiated with a Write command as shown in *FIGURE 67*. The bank and column addresses are provided with the Write command and auto precharge is either enabled or disabled for that access with the AP bit (CA3-3). If auto precharge is enabled, the row being accessed is precharged after tWR(min) has been met or after the number of CK4 cycles programmed in the WR field of MR4 OP[6:0], depending on the implementation choice per DRAM vendor.

The length of the burst initiated with a Write command is sixteen symbols in PAM3 mode and thirty-two bits if in NRZ mode and the column address is unique for this burst of sixteen or thirty-two. There is no interruption nor truncation of Write bursts.



NOTES:

1) BA = Bank address; C = Column Address; CAPAR = CA Parity, V = Valid (H or L but not floating), AP = Auto Precharge, EN AP = Enable Auto Precharge, DIS AP = Disable Auto Precharge

2) WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 67 — Write Command

Write timings are shown for PAM3 and NRZ modes in *FIGURE 68*. During Write bursts, the first valid data-in element must be available at the input latch after the Write Latency (WL).

The Write Latency is defined as

WL (MR2 OP[4:0]) * tCK4 + tWCK2DQI

when CAPARBLK OFF (MR15 OP1 = 0) or CAPARBLK ON (MR15 OP1 = 1) and implicit CAPARBLK latency.

The Write Latency is defined as

WL (MR2 OP[4:0]) * tCK4 + CAPARBLK_LAT (MR15 OP[5:3]) * tCK4 + tWCK2DQI

when CAPARBLK ON (MR15 OP1 = 1) and explicit CAPARBLK latency.

Where WL is the number of CK4 cycles programed in MR2 OP[3:0], CAPARBLK_LAT is the number of CK4 cycles programmed in MR15 OP[5:3] and tWCK2DQI is the WCK to DQ/DQE offset as measured at the DRAM balls to ensure concurrent arrival at the latch. The total delay is relative to the data eye center averaged over DQ/DQE. The maximum skew within a DQ/DQE is defined by tDQ2DQI.

GDDR7 devices support either implicit (IRA3 DQ[3:2] = 11_B); or explicit (IRA3 DQ[3:2] = 10_B); or both implicit and explicit CAPARBLK_LAT (IRA3 DQ[3:2] = 00_B). If the DRAM supports both, the host has the flexibility to program the CAPARBLK_LAT control register (MR15 OP6) to support either implicit or explicit. Refer to the *COMMAND ADDRESS PARITY (CAPAR) PROTOCOL* section for more details and vendor datasheets to see which method is supported.

6.8 Write (cont'd)

The data input valid window, tDIVW, defines the time region when input data must be valid for reliable data capture at the receiver for any one worst-case channel. It accounts for jitter between data and WCK at the latching point introduced in the path between the DRAM pads and the latching point. Any additional jitter introduced into the source signals (i.e., within the system before the DRAM pad) must be accounted for in the final timing budget. tDIVW is for DRAM design only and valid on the silicon die. It is not intended to be measured. In general, tDIVW is smaller than tDIPW.

The data input pulse width, tDIPW, defines the minimum positive or negative input pulse width for any one worst-case channel required for proper propagation of an external signal to the receiver. tDIPW is for DRAM design only and valid on the silicon die. It is not intended to be measured. In general, tDIPW is larger than tDIVW.

Upon completion of a burst, assuming no other Write data is expected on the bus the DQ and DQE signals will be driven according to the ODT state. Any additional input data will be ignored. Data for any Write burst may not be truncated with a subsequent Write command.

Data from any Write burst may be concatenated with data from a subsequent Write command. A continuous flow of data can be maintained. The first data element from the new burst follows the last element of a completed burst. The new Write command should be issued after the previous Write command according to the tCCD or tCCDSB timing. If that Write command is to another bank, then an Activate command must precede the Write command and tRCDWR also must be met.

A Read can be issued any time after a Write command if the internal turnaround time tWTR is met. If that Read command is to another bank, then an Activate command must precede the Read command and tRCDRD also must be met.

A Precharge can also be issued after tWR has been met. After the Precharge command, a subsequent command to the same bank cannot be issued until tRP is met.

A MRS command can be issued while a bank is open after tWR2MRS to program some registers. See the *MODE REGISTER SET (MRS)* command section for details on what registers are allowed to be programmed while a bank is active.

FIGURE 69 through *FIGURE 76* illustrate Write operations in PAM3 mode including:

Figure 69 — Single PAM3 Write with WRCRC and CA Parity Enabled

Figure 70 — Single PAM3 Write with WRCRC Disabled

Figure 71 — Gapless PAM3 Writes, Different Banks (tCCD = 2)

Figure 72 — Non-Gapless PAM3 Writes, Different Banks (tCCD = 3)

Figure 73 — Non-Gapless PAM3 Writes, Same Bank (tCCDSB = 4)

Figure 74 — Write to Precharge (PAM3 Mode)

Figure 75 — Write to Read (PAM3 Mode)

Figure 76 — Write to MRS (PAM3 Mode)

6.8 Write (cont'd)

FIGURE 77 through *FIGURE 79* illustrate Write operations in NRZ mode including:

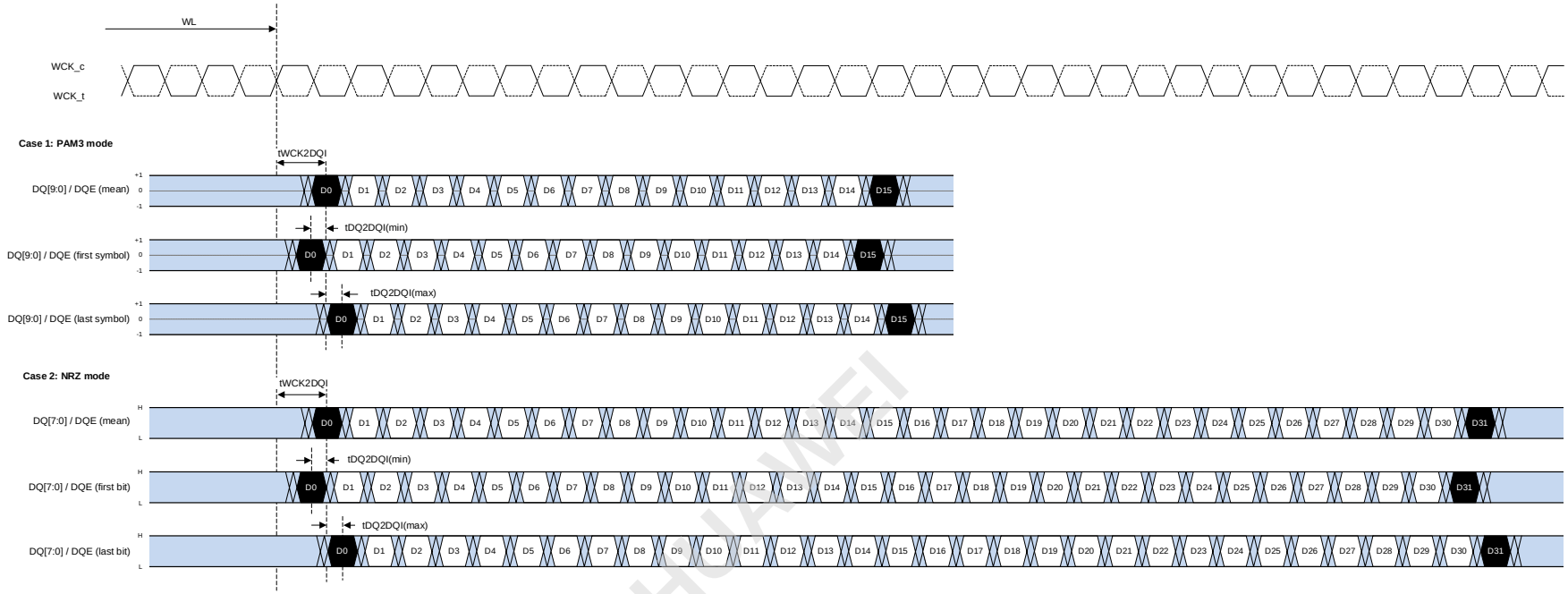
Figure 77 — Single NRZ Write with WRCRC and CA Parity Enabled

Figure 78 — Gapless NRZ Writes (tCCD = 4)

Figure 79 — Single NRZ Write with DQE Disabled (RDCRC, WRCRC, Severity, and Poison Disabled)

See the *DATA INTEGRITY* Section for details on Command Address Parity, WRCRC, Poison, Severity, and ERR signaling.

6.8 Write (cont'd)

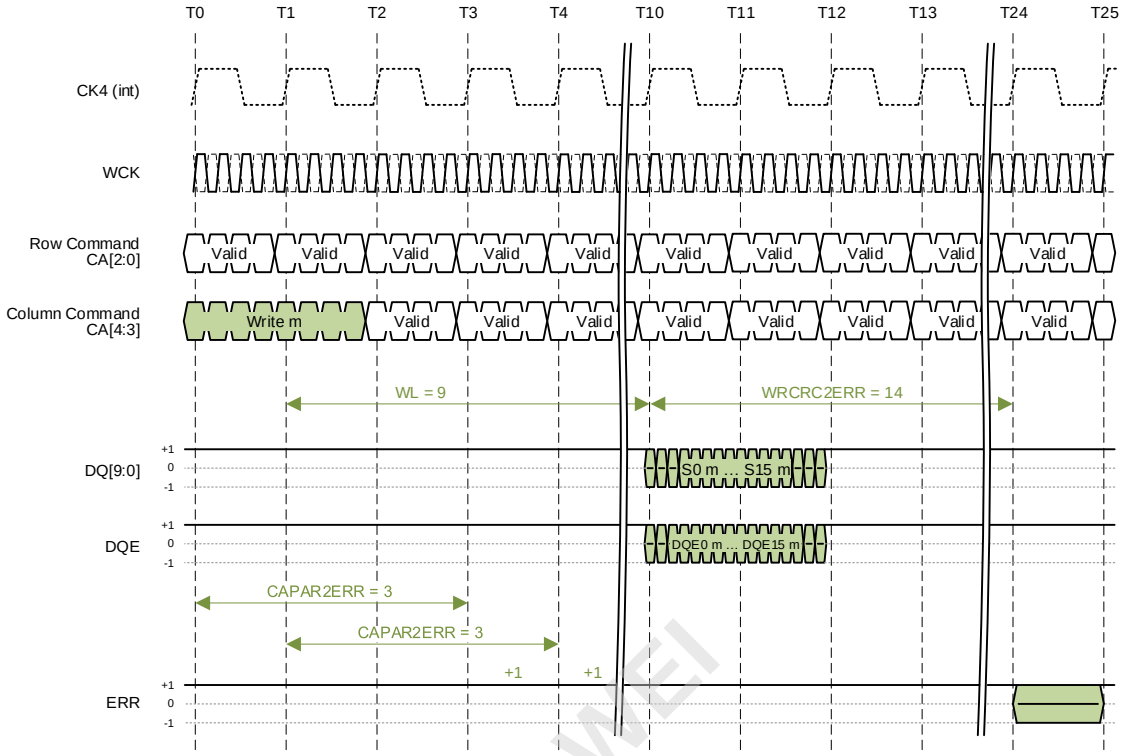


NOTES:

1. WL is the Write latency programmed in Mode Register MR2.
2. tWCK2DQI parameter values are positive numbers and could be less than 1 nCK4 as illustrated or more than 1 nCK4 depending on design implementation, and can vary across PVT. Data training is required to determine the actual tWCK2DQI value for stable Write operation.
3. tDQ2DQI defines the minimum to maximum variation of tWCK2DQI within DQ/DQE.

Figure 68 — Write Lane Timings

6.8 Write (cont'd)

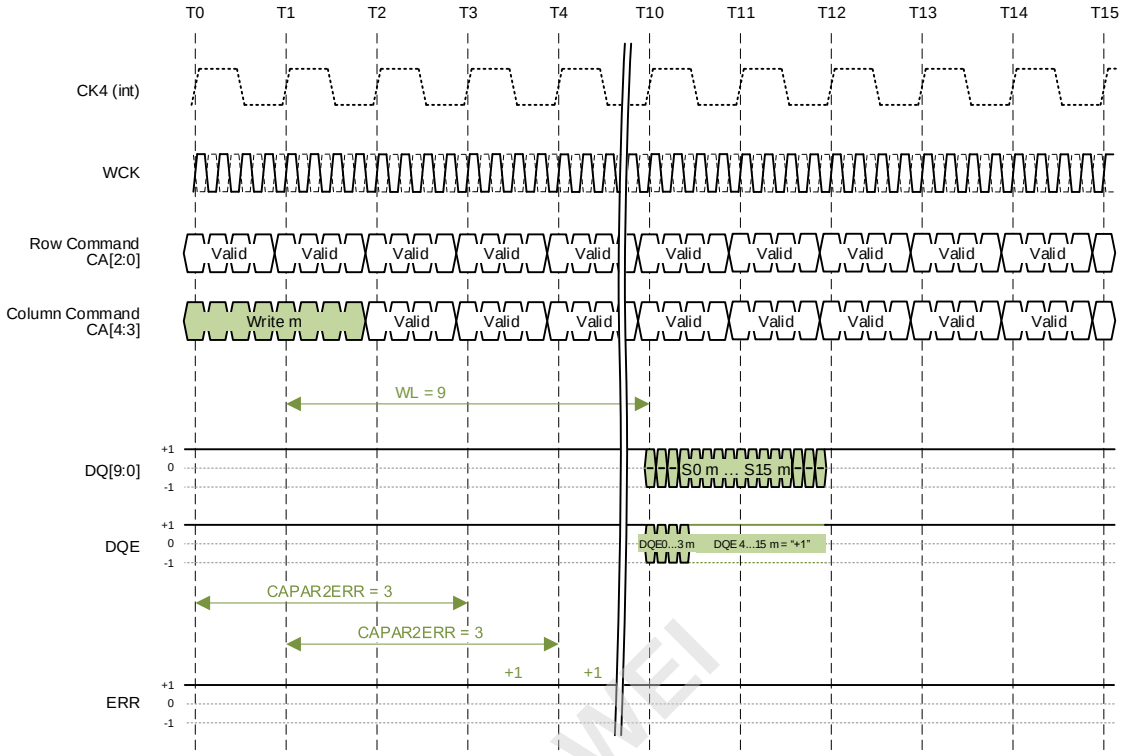


NOTES:

1. WL = 9 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T23, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Write command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. At T24, the ERR output has +1 (no error), 0 (WRCRC error), or -1 (CA Parity error) as possible outcomes. All outcomes shown for illustration purposes. CA parity will be signaled if both a CA parity and a WRCRC occur in the same cycle.
7. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO. For illustration purposes, WRCRC2ERR is shown with 14 nCK4 digital and 0 ns analog output delay.
8. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 69 — Single PAM3 Write with WRCRC and CA Parity Enabled

6.8 Write (cont'd)

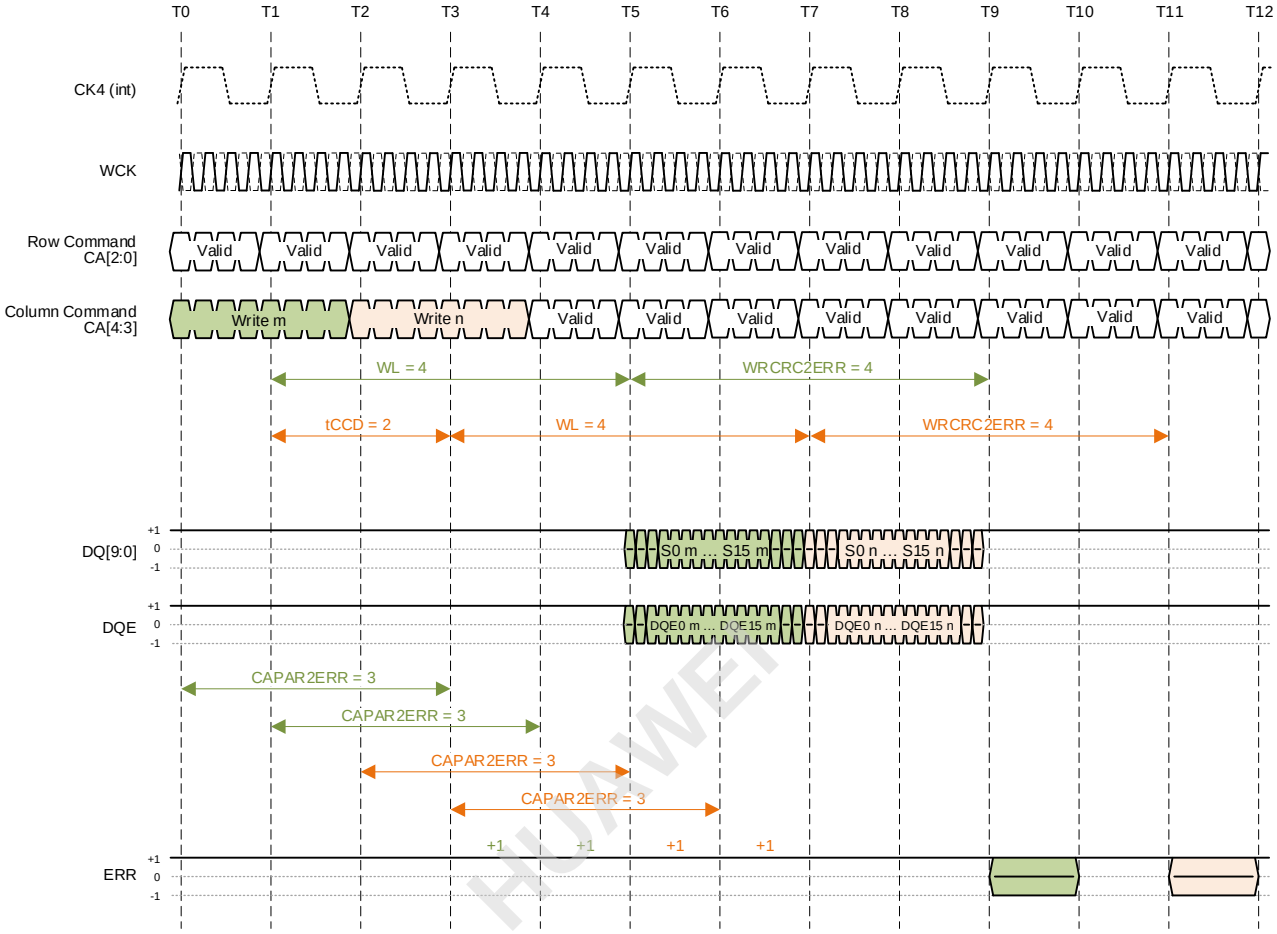


NOTES:

1. WL = 9 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T14, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Write command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 70 — Single PAM3 Write with WRCRC Disabled

6.8 Write (cont'd)

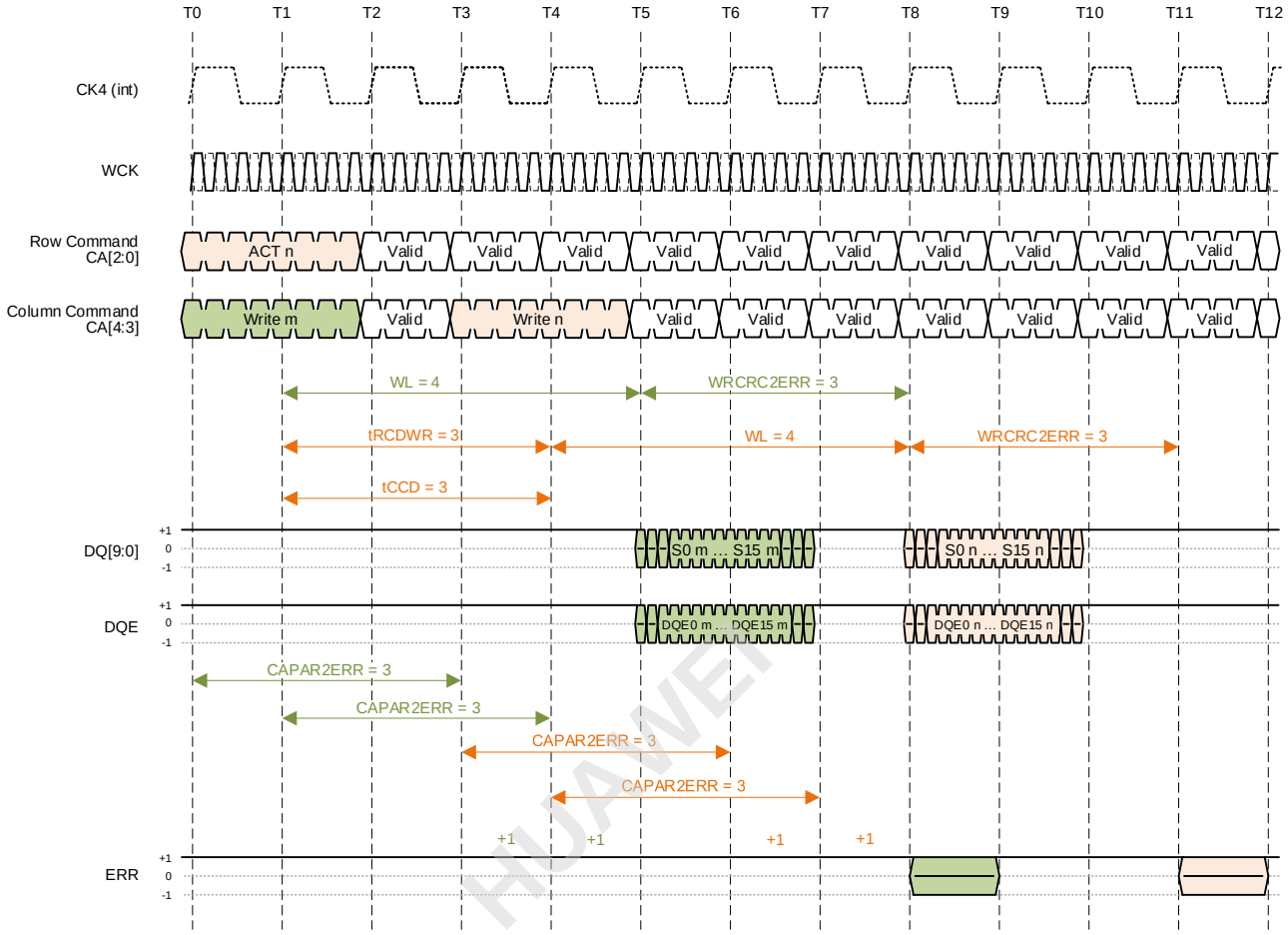


NOTES:

1. WL = 4 is shown for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Write commands and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T8 and T10, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Write command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. At T9 and T11, the ERR output has +1 (no error), 0 (WRCRC error) or -1 (CA Parity error) as possible outcomes. All outcomes shown for illustration purposes. CA parity will be signaled if both a CA parity and a WRCRC occur in the same cycle.
7. Gapless Writes must be to different banks as tCCD = 2 < tCCDSB.
8. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO. For illustration purposes, WRCRC2ERR is shown with 4 nCK4 digital and 0 ns analog output delay.
9. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 71 — Gapless PAM3 Writes, Different Banks (tCCD = 2)

6.8 Write (cont'd)

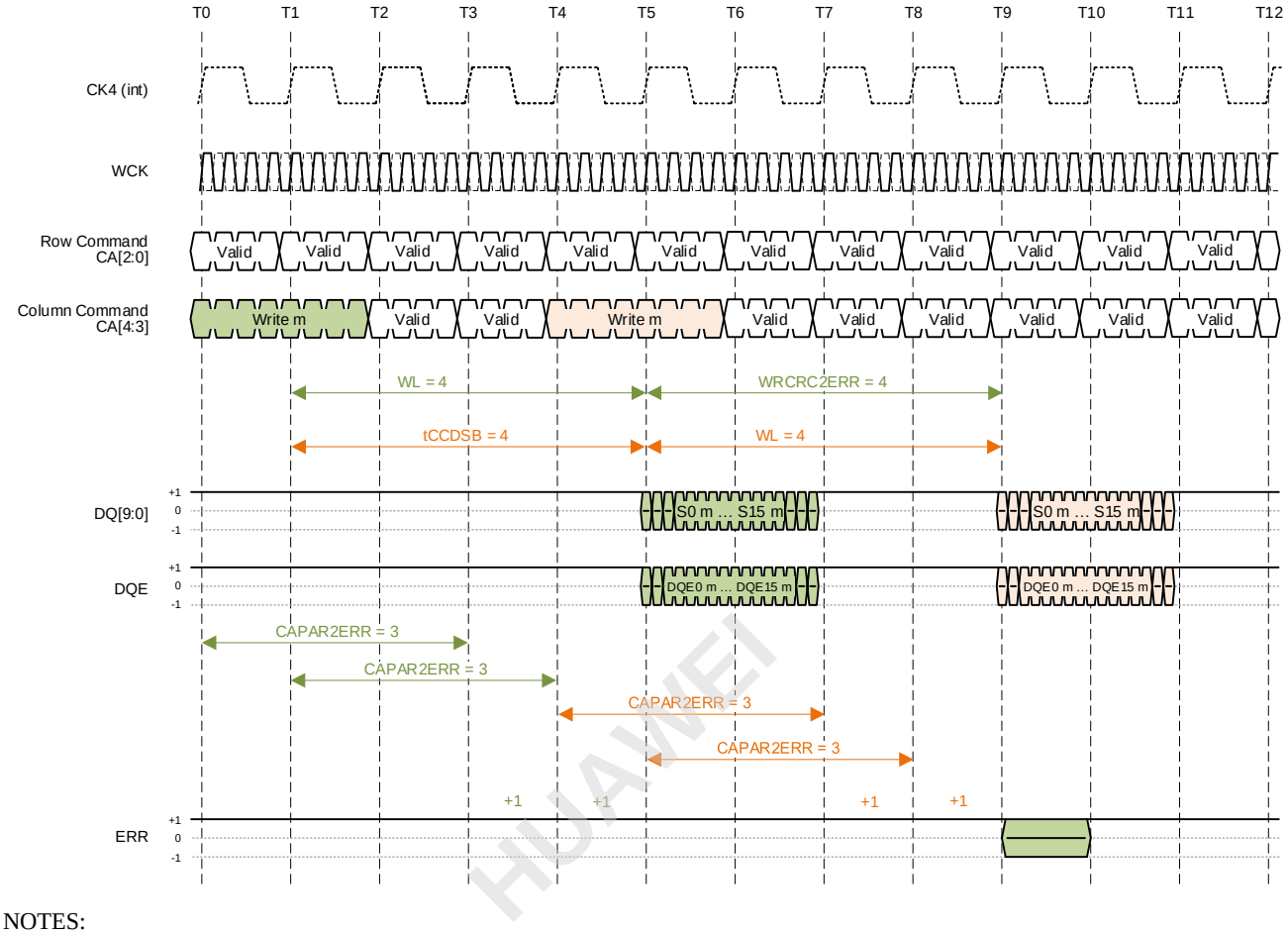


NOTES:

1. WL = 4, tCCD = 3 and tRCDWR = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the first Write command and tRCDWR must be met. Non-gapless Writes with tCCD = 3 must be to different banks as tCCD < tCCDSB.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T7, T9 and T10, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Write command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. At T8 and T11, the ERR output has +1 (no error), 0 (WRCRC error) or -1 (CA Parity error) as possible outcomes. All outcomes shown for illustration purposes. CA parity will be signaled if both a CA parity and a WRCRC occur in the same cycle.
7. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO. For illustration purposes, WRCRC2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.
8. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 72 — Non-Gapless PAM3 Writes, Different Banks (tCCD = 3)

6.8 Write (cont'd)

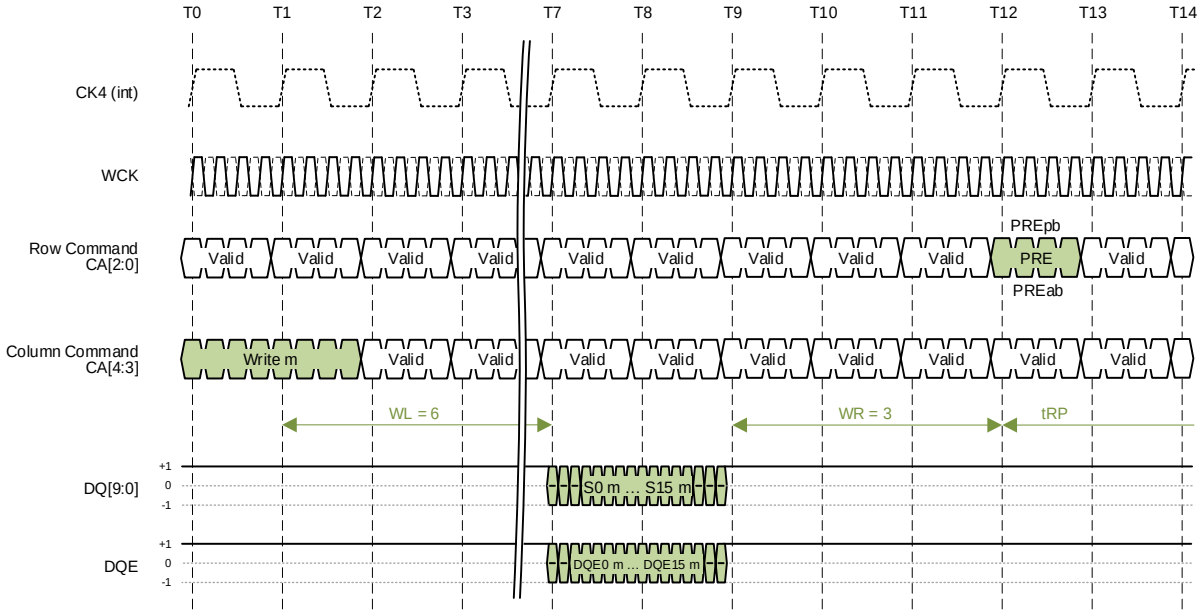


NOTES:

1. WL = 3 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the first Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T8, T10 and T11, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Write command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. At T9, the ERR output has +1 (no error), 0 (WRCRC error) or -1 (CA Parity error) as possible outcomes. All outcomes shown for illustration purposes. CA parity is signaled if both a CA parity and a WRCRC occur in the same cycle.
7. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO. For illustration purposes, WRCRC2ERR is shown with 4 nCK4 digital and 0 ns analog output delay.
8. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.

Figure 73 — Non-Gapless PAM3 Writes, Same Bank (tCCDSB = 4)

6.8 Write (cont'd)

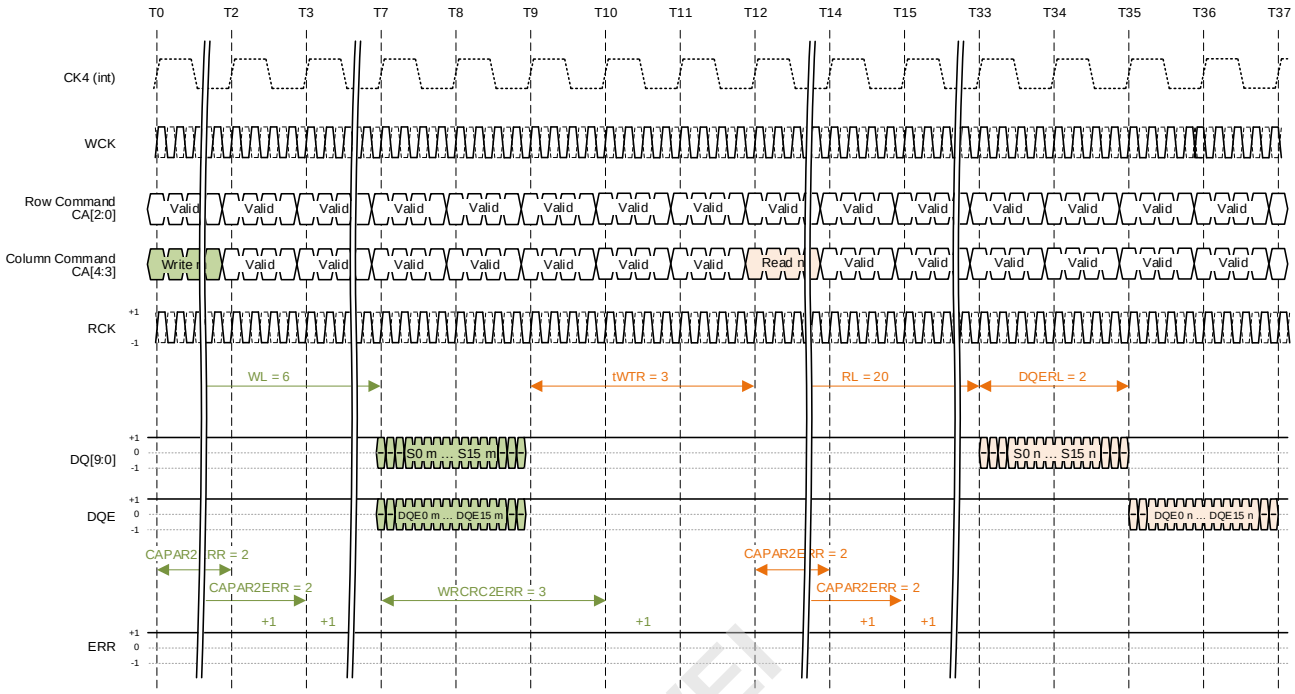


NOTES:

1. WL= 6 and WR = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.

Figure 74 — Write to Precharge (PAM3 Mode)

6.8 Write (cont'd)

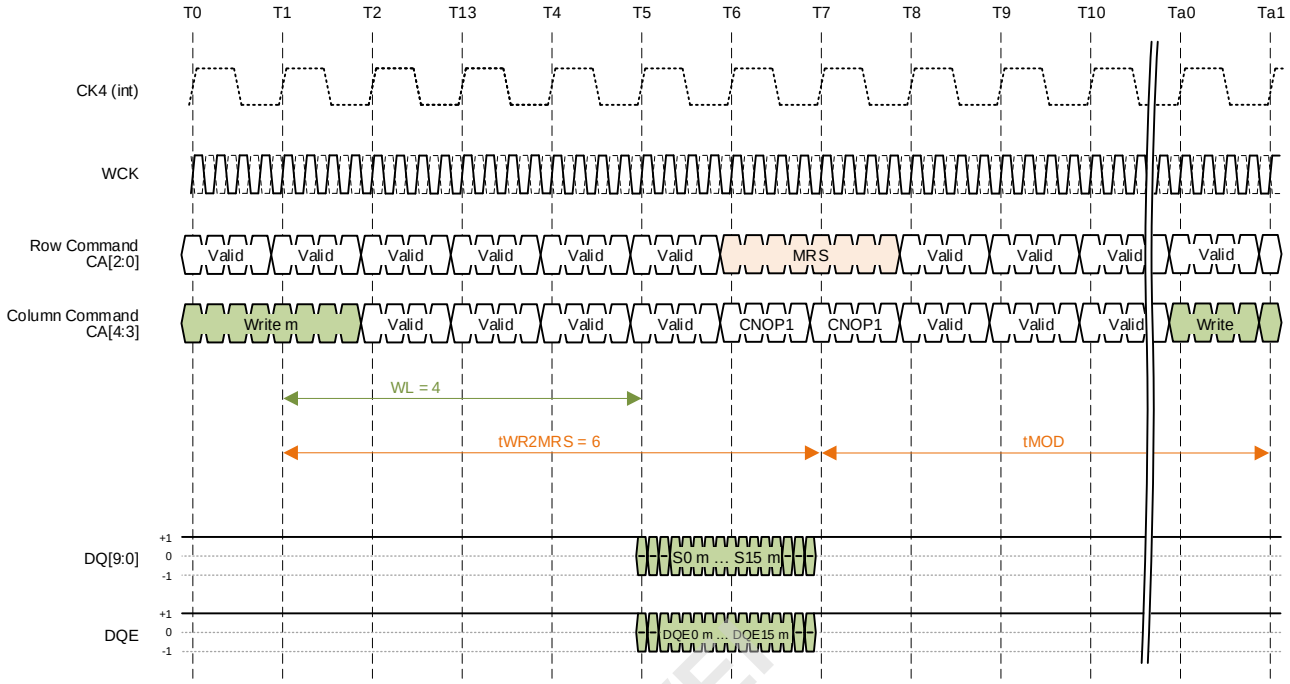


NOTES:

1. RL = 20, tWTR = 3, WL = 6, DQERL = 2 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read and Write commands and tRCDRD and tRCDWR respectively, must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA, tWCK2DQI, tWCK2DQO = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T7 and T8 thru T17, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Read and Write commands is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO. For illustration purposes, WRCRC2ERR is shown with 3 nCK4 digital and 0 ns analog output delay.
7. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
8. Write and Read commands are 2 cycle commands but shown with time breaks for illustration purposes.

Figure 75 — Write to Read (PAM3 Mode)

6.8 Write (cont'd)

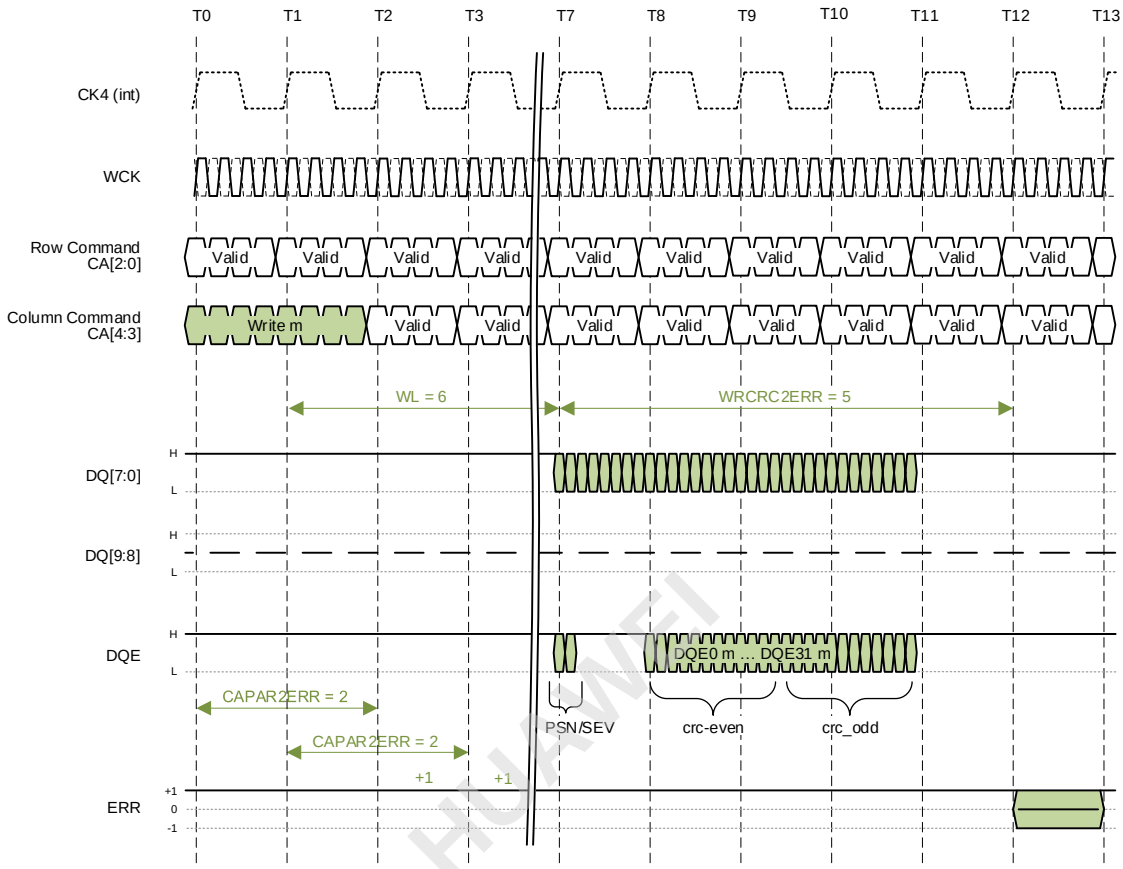


NOTES:

1. WL = 4 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. tWR2MRS = tWR2MRS(min) = 6 is shown as example for illustration purposes. tWR2MRS(min) = WL + BL/8.
3. A MRS command while a bank is active is only legal to certain MR. See the *MODE REGISTERS* section for details.
3. An Activate (ACT) command is required to be issued before the first Write command and tRCDWR must be met.
4. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
5. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.

Figure 76 — Write to MRS (PAM3 Mode)

6.8 Write (cont'd)

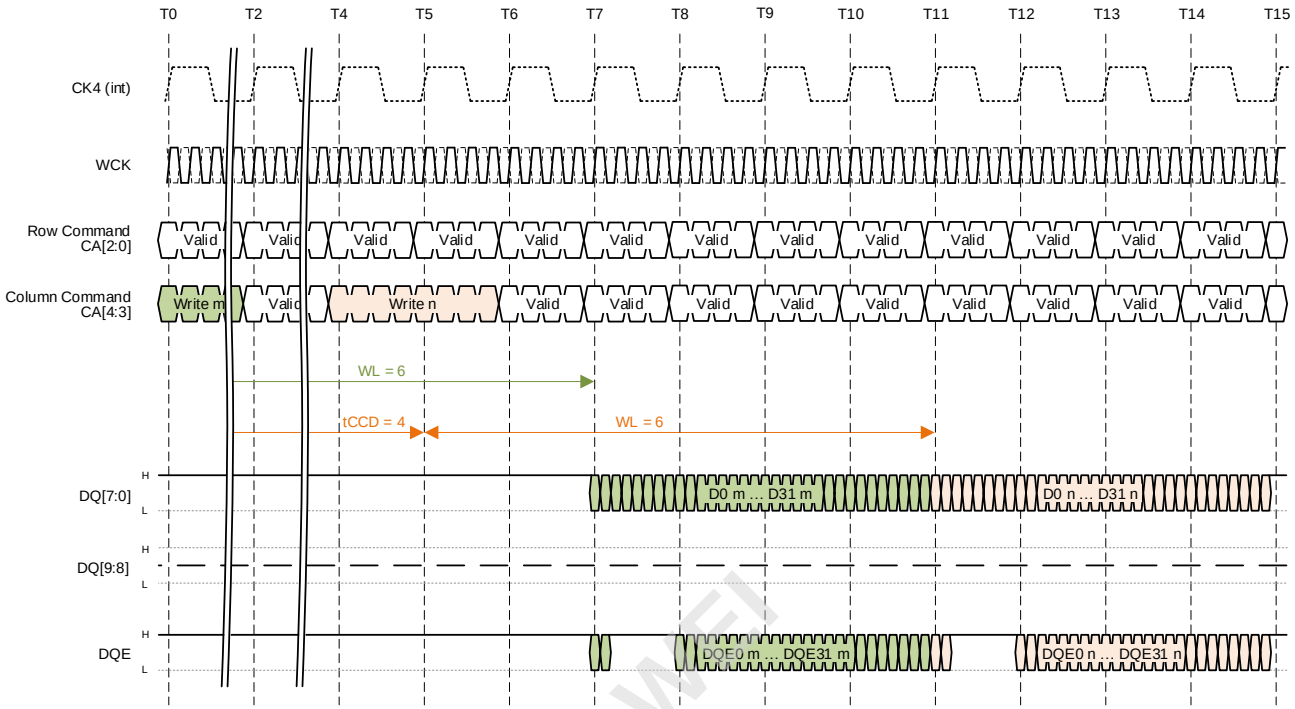


NOTES:

1. WL = 6 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. At T0 thru T11, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Write command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. At T12, the ERR output has +1 (no error), 0 (WRCRC error) or -1 (CA Parity error) as possible outcomes. All outcomes shown for illustration purposes. CA parity will be signaled if both a CA parity and a WRCRC occur in the same cycle.
7. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQ[9:8] are turned off on the GDDR7 device.
8. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO. For illustration purposes, WRCRC2ERR is shown with 5 nCK4 digital and 0 ns analog output delay.
9. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
10. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQE bit positions [7:2] = 6'b111111.
11. When the PAM3 register (MR0 OP8) is set to NRZ mode, the CRC is PAM3 coded and the binary representation is transmitted on the DQE signal at bit positions [19:8] for crc_even and bit positions [31:20] for crc_odd.

Figure 77 — Single NRZ Write with WRCRC and CA Parity Enabled

6.8 Write (cont'd)

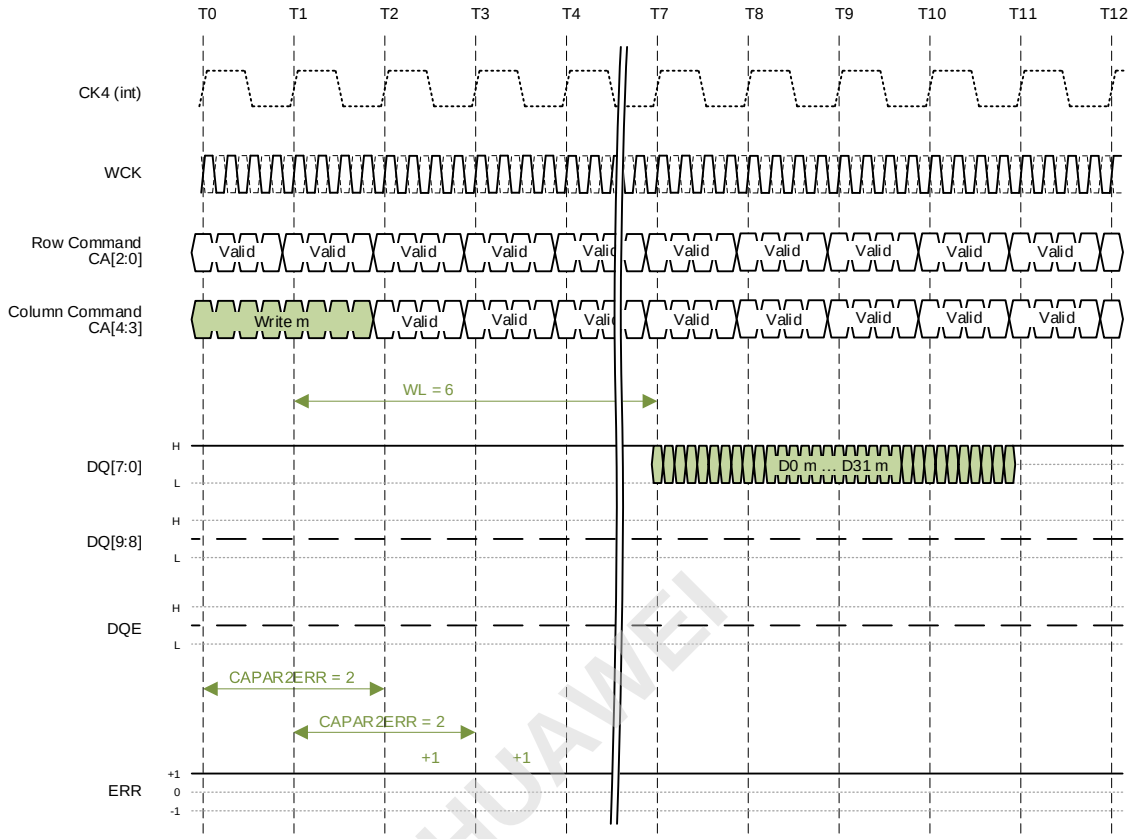


NOTES:

1. WL = 6 and tCCD = 4 are shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the first Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQ[9:8] are turned off on the GDDR7 device.
6. Gapless Writes in NRZ mode could be to any bank as tCCD = tCCDSB = 4.
7. Write commands are 2-cycle commands but are shown with time breaks for illustration purposes.
8. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQE bit positions [7:2] = 6'b111111.
9. When the PAM3 register (MR0 OP8) is set to NRZ mode, the CRC is PAM coded and the binary representation is transmitted on the DQE signal at bit positions [19:8] for crc_even and bit positions [31:20] for crc_odd.

Figure 78 — Gapless NRZ Writes (tCCD = 4)

6.8 Write (cont'd)



NOTES:

1. WL = 6 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Write command and tRCDWR must be met.
3. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA and tWCK2DQI = 0 are shown for illustration purposes.
4. For Write operations it is important that the latching point meet the data valid window requirements, which may or may not be center aligned at the balls.
5. DQE is disabled using MR5 OP10 (DQE_HZ). DQE can be turned off when RDCRC (MR0 OP3), WRCRC (MR0 OP4), Severity (MR0 OP9) and Poison (MR0 OP10) are disabled in the Mode Registers. If the host still requires Severity, MR5 OP9 (SEV2ERR) enables the host to receive severity information on the ERR signal.
6. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.

Figure 79 — Single NRZ Write with DQE Disabled (RDCRC, WRCRC, Severity, and Poison Disabled)

6.9 Read Clock (RCK)

GDDR7 SGRAMs provide a Read Clock signal (RCK) per channel, on the RCK_t and RCK_c signals. The RCK drives out a clock pattern in phase (edge aligned) with the DQs, that can be used by the host to latch read data. The RCK can be set to operate as a differential or as a single ended signal, as selected by the RCK_TYPE bit in MR9 OP2. When configured as single ended, only the RCK_t signal toggles while the RCK_c is kept High-Z.

The RCK signals toggle between high and low levels; in PAM3 mode it can be configured to full swing or half swing by setting the RCKLEVEL bit in MR9 OP3 to 0_B or $1_{B,}$ respectively. In NRZ mode MR9 OP3 is ignored and the RCK always toggles H/L full swing.

When RCK is disabled both RCK signals are set to High-Z. RCK can be enabled or disabled by setting the RCKMODE bits in MR9 OP[1:0] to one of the following values:

- MR9 OP[1:0] = 00_B : RCK Disabled (default).
- MR9 OP[1:0] = 11_B : RCK Always On. RCK toggles at WCK rate as a free running clock.
- RCK start/stop: There are two RCK start/stop modes, that differ in the command that starts the RCK, as described in the RCK start/stop modes section below.

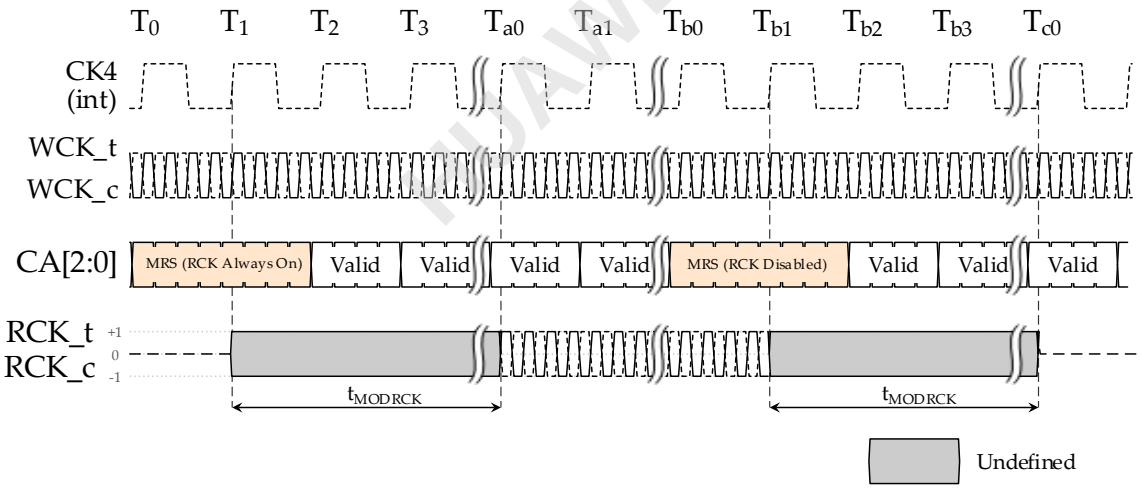
It is not allowed to switch directly between different active RCK modes. Instead, the host must first change to RCK Disable mode and wait for tMOD prior switching to a different RCK mode.

Once RCK always on mode is enabled, host is not allowed to change MR9 OP[11:2], MR12 OP[11:10] fields until RCK mode is disabled. If RCK start mode with Read (MR9 OP[1:0]=01_B) is selected and RCK starts toggle starts with Read (RD, RDA, IRD, RDTR or RDWTEC) command, host must issue RCKSTOP command to GDDR7 SGRAM and wait tRCKSTOP2MR9_12 to change any setting in MR9 or MR12. If RCK start/stop mode with RCKSTRT command (MR9 OP[1:0]=10_B) is selected and RCK starts toggle starts with RCKSTART command, host must issue RCKSTOP command to GDDR7 SGRAM and wait tRCKLSTOP2MR9_12 to change any settings in MR9 or MR12.

6.9.1 RCK Always On

RCK can be configured to Always On by setting RCKMODE in MR9 OP[1:0] to 11_B. When set, the RCK signals will asynchronously transition within tMOD from High-Z to start toggling at WCK rate, as shown in *FIGURE 80*. RCK will keep toggling until one of the following events occurs:

- The host changes from RCK Always On to RCK Disabled: RCK will transition asynchronously from toggling to High-Z before tMOD expires.
- The channel enters sleep mode: the RCK signals will transition asynchronously from toggling to High-Z within tCPDED time after the SLE or SRSE commands are issued. The RCK will automatically resume toggling upon Sleep Exit, during tRCK_AON_SLX.
- Channel enters CA training: the RCK will transition asynchronously from toggling at WCK rate to toggling at CK4 rate during tCATE. Upon CA training exit RCK will asynchronously resume toggling at WCK rate during tRCK_AON_CATX.
- When CAPARBLK is enabled (MR15_OP1=1_B) and GDDR7 DRAM is in RCK always on mode (MR9 OP[1:0]=11_B), GDDR7 DRAM RCK will continue to toggle at WCK rate until tCAPAR_UNLOCK expires if CA parity error occurs. During tCATE2RCK, the GDDR7 DRAM RCK toggling will asynchronously transition from WCK rate to CK4 rate. The asynchronous transition may include stopping the RCK. First nCK4 pulse may be incomplete. The toggling will resume automatically upon CA training exit during tRCK_AON_CATX as described above.



NOTE 1 After the MRS command that switches from/to RCK Always On and RCK Disabled, RCK starts/stops toggling asynchronously at an undefined time during tMOD. There could be incomplete pulses during the enable and disable transitions.

NOTE 2 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 80 — RCK Always On Enable/Disable Timings

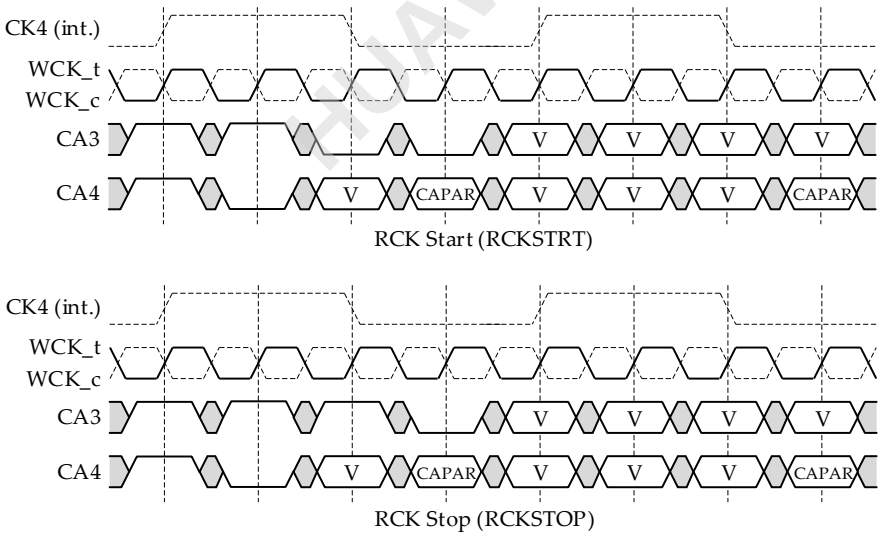
6.9.2 RCK Start/Stop Modes

The average operating power can be reduced if RCK is enabled only during the periods when it is needed. GDDR7 SGRAMs provide RCK modes that allow to synchronously control the RCK start/stop toggling.

There are two different RCK start/stop modes that differ in the command that triggers the start of RCK:

- RCKMODE with MR9 OP[1:0] = 10_B: RCK starts only with the RCKSTRT command; Read commands do not trigger RCK, the host needs to take care to set the RCK to toggle by timely issuing an RCKSTRT command.
- RCKMODE with MR9 OP[1:0] = 01_B: RCK starts triggered by any Read command (RD, RDA, IRD, RDTR, RDWTEC), RCKSTRT is not allowed in this mode.

In both modes, an RCKSTOP command is required to synchronously stop the RCK and set the RCK signals back to High-Z. Please check *FIGURE 81* for the RCKSTRT and RCKSTOP commands encoding. The RCK must be properly stopped before changing RCKMODE and before entering any Sleep modes or CA training mode (SLE, SRSE or CATE commands). In a CA Parity Block event, if RCK is toggling it will continue to toggle at WCK rate until tCAPAR_UNLOCK expires. During tCATE2RCK, the GDDR7 DRAM RCK toggling will asynchronously transition from WCK rate to CK4 rate. The asynchronous transition may include stopping the RCK. First nCK4 pulse may be incomplete. The RCK may also stop during tCAPAR_UNLOCK if the Valid command before the tCAPAR_UNKNOWN is a RCKSTOP command. In this case the RCK will stop and transition to High-Z per the RCKSTOP procedure as outlined in the *READ CLOCK (RCK)* section before transitioning to CK4 rate no later than tCATE2RCK. The RCK signals will remain High-Z after tCATX until the host issues an RCK start command.



NOTE 1 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 81 — RCKSTRT and RCKSTOP Commands

A time tMOD after the RCKMODE (MR9 OP[1:0]) bits are set to one of the RCK start/stop modes, the host can synchronously start the RCK with the proper RCK start command. RCKSTRT and RCKSTOP commands are allowed in bank active, bank idle, SRF with TR = H and REFab with TR = H. RCKSTRT and RCKSTOP commands are illegal in RCK Always On and RCK Disabled Modes.

6.9 Read Clock (RCK) (cont'd)

RCK Start/Stop Preambles, Postambles and Timings

In both RCK Start/Stop modes, RCK is started with a staggered sequence of preambles that ends up with the RCK toggling at WCK rate. The duration of the pre- and post- ambles are programmable via RCKEN, RCK_LS and RCKSTOP_LAT in MR9, please refer to the *MODE REGISTERS* section for details on the allowed ranges and values.

FIGURE 82 shows how RCK starts and stops toggling and the different timings and latencies involved:

1. RCKEN, MR9 OP[7:3]: the programmable synchronous RCK Enable latency involves an initial asynchronous preamble that allows to transition from High-Z to Static levels. The value programmed in the RCKEN register may depend on whether CAPARBLK is enabled or disabled (MR15 OP1).

   RCK Enable latency is defined as

   RCKEN (MR9 OP[7:3])

   when CAPARBLK OFF (MR15 OP1 = 0) or CAPARBLK ON (MR15 OP1 = 1) and implicit CAPARBLK latency.

   RCK Enable latency is defined as

   RCKEN (MR9 OP[7:3]) + CAPARBLK_LAT (MR15 OP[5:3])

   when CAPARBLK ON (MR15 OP1 = 1) and explicit CAPARBLK latency.

   GDDR7 devices support either implicit (IRA3 DQ[3:2] = 11_B); or explicit (IRA3 DQ[3:2] = 10_B); or both implicit and explicit CAPARBLK_LAT (IRA3 DQ[3:2] = 00_B). If the DRAM supports both, the host has the flexibility to program the CAPARBLK_LAT control register (MR15 OP6) to support either implicit or explicit. Refer to the *CAPAR WITH COMMAND BLOCKING (CAPARBLK)* section for more details and vendor datasheets to see which method is supported.

   The duration of this RCK2LZ preamble is defined as tRCK2LZ = MAX(x ns, y nCK4) and there is no minimum tRCK2LZ specified. After tRCK2LZ, RCK_t and RCK_c drive low and high levels, respectively. RCK remains in static preamble during tRCK_ST until RCKEN expires, the minimum allowed tRCK_ST is 4 nCK4.

2. RCK_LS, MR9 OP[2]: The Low-Speed preamble is the time that the RCK will be toggling at half WCK rate before transitioning to full WCK toggling rate. The Low-Speed toggling preamble is controlled via the RCK_LS bit in MR9 OP[2] and can be set to either skip (disabled) with MR9 OP[2] = 0_B or to 2 CK4 cycles (4 RCK cycles) duration setting MR9 OP[2] = 1_B.

3. tRCK_HS is the duration of the High-Speed toggling preamble, from the end of the RCK_LS preamble to the first data out on the DQs, or to the RCKSTOP command in case that there is no read operation between RCKSTRT and RCKSTOP commands. The minimum allowed tRCK_HS is vendor specific.

6.9 Read Clock (RCK) (cont'd)

4. RCKSTOP_LAT, MR9 OP[11:9] is the programmable latency that defines the number of CK4 cycles from RCKSTOP command to stop toggling the RCK signals. The value programmed in the RCKSTOP_LAT register may depend on whether CAPARBLK is enabled or disabled (MR15 OP1).

   RCK Stop latency is defined as

   RCKSTOP_LAT (MR9 OP[11:9])

   when CAPARBLK OFF (MR15 OP1 = 0) or CAPARBLK ON (MR15 OP1 = 1) and implicit CAPARBLK latency.

   RCK Stop latency is defined as

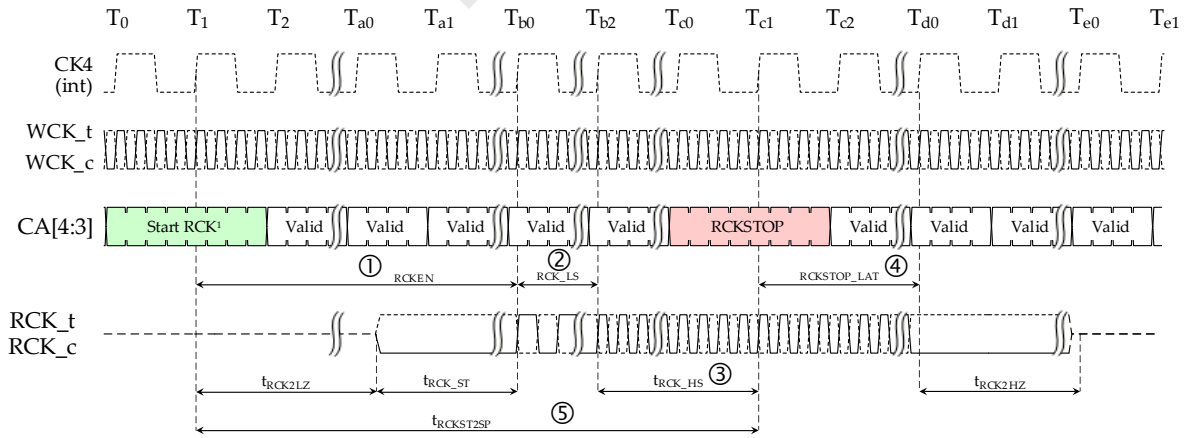
   RCKSTOP_LAT (MR9 OP[11:9]) + CAPARBLK_LAT (MR15 OP[5:3])

   when CAPARBLK ON (MR15 OP1 = 1) and explicit CAPARBLK latency.

   GDDR7 devices support either implicit (IRA3 DQ[3:2] = 11_B); or explicit (IRA3 DQ[3:2] = 10_B); or both implicit and explicit CAPARBLK_LAT (IRA3 DQ[3:2] = 00_B). If the DRAM supports both, the host has the flexibility to program the CAPARBLK_LAT control register (MR15 OP6) to support either implicit or explicit. Refer to the *CAPAR WITH COMMAND BLOCKING (CAPARBLK)* section for more details and vendor datasheets to see which method is supported.

   When RCKSTOP_LAT expires the RCK signals are set to static levels (RCK_t Low, RCK_c High) and will remain static for at least the vendor specific duration of the static postamble tRCK2HZ before transitioning to High-Z.

5. tRCKST2SP is the time between the RCKSTRT command and the RCKSTOP command. The minimum allowed is: tRCKST2SP (MIN) = RCKEN + RCK_LS + tRCK_HS (MIN)



NOTE 1 "Start RCK" is the command that triggers the RCK start toggling, depending on the RCKMODE Mode Register settings it can be either the RCKSTRT command or a Read command (RD, RDA, IRD, RDTR or RDWTEC).

NOTE 2 The duration of the High-Speed preamble tRCK_HS is defined as the time between the end of the RCK_LS and the first data out on the DQs, or the RCKSTOP command if there is no read operation. In this example the second case is shown.

NOTE 3 For illustration purposes, all analog tWCK2RCK timing assumed as 0ps for all RCK state transitions + pulses.

Figure 82 — Overview of the RCK Start/Stop Protocol

6.9 Read Clock (RCK) (cont'd)

6. The duration of the RCK high speed toggling postamble tRCKPST is defined as the time between the last UI of the data burst and the end of the RCKSTOP_LAT as shown in *FIGURE 83*. For the case that there is no read operation between RCKSTRT and RCKSTOP commands, tRCKPST does not apply. The minimum allowed tRCKPST duration is 2 nCK4.

7. tRD2RCKSTOP is the minimum time between a Read command and the next RCKSTOP command as shown in *FIGURE 83*. The minimum allowed is tRD2RCKSTOP(MIN) = RL + BL/8 + DQERL + tRCKPST–RCKSTOP_LAT

8. tRCKSTRT2RD is the time between RCKSTRT and the next RD, RDA, RDTR, IRD or RDWTEC, as shown in *FIGURE 84*.

9. tRCKSP2ST is the time between an RCKSTOP command and the next RCKSTRT (or Read command), as shown in *FIGURE 85*. The minimum allowed is tRCKSP2ST(MIN) = RCKSTOP_LAT - tRCK2LZ + 4 nCK4

Table 85 — RCK Related AC Timings

PARAMETER	SYMBOL	VALUES		UNIT	NOTES
		MIN	MAX		
RCK High-Z to static preamble time	t_{RCK2LZ}	-		ns	1
RCK Static preamble	t_{RCK_ST}	4	-	nCK4	2
RCK High Speed preamble	t_{RCK_HS}		-	nCK4	3
RCK static postamble to High-Z time	t_{RCK2HZ}		-	ns	4
RCK postamble	t_{RCKPST}	2	-	nCK4	
RCKSTRT to RCKSTOP command delay	$t_{RCKST2SP}$	RCKEN + RCK_LS + tRCK_HS(min)	-	nCK4	
RD to RCKSTOP command delay	$t_{RD2RCKSTOP}$	RL + BL/8 + DQERL + tRCKPST - RCKSTOP_LAT	-	nCK4	5
RCKSTRT command to Read command delay	$t_{RCKSTRT2RD}$	2	-	nCK4	5
RCKSTOP to RCKSTRT or Read command delay	$t_{RCKSP2ST}$		-	ns	6
RCKSTOP to MR9 or MR12 MRS programming delay	$t_{RCKSTOP2MR9\_12}$	RCKSTOP_LAT x tCK4 + 4 x tCK4 + tRCK2HZ(max)	-	ns	

NOTE 1 Parameter tRCK2LZ defines the time for RCK to transition from High-Z to driving static H/L levels.

NOTE 2 The RCKEN latency must be set to a value large enough to satisfy the tRCK_ST timing.

NOTE 3 When RCKMODE is Start with Read commands (MR9 OP[1:0]=01_B) tRCK_HS = RL - RCKEN - RCK_LS. When RCKMODE is Start with RCKSTRT command (MR9 OP[1:0]=10_B) tRCK_HS = RL + tRCKSTRT2RD - RCKEN - RCK_LS

NOTE 4 Parameter tRCK2HZ defines the time for RCK to transition from driving static H/L levels to High-Z.

NOTE 5 Read commands in this context comprise RD, RDA, IRD, RDTR and RDWTEC commands.

NOTE 6 Parameter tRCKSP2ST applies when RCKMODE is set to Start with Read commands (MR9 OP[1:0]=01_B) or Start with RCKSTRT command (MR9 OP[1:0]=10_B). Read commands in this context comprise RD, RDA, IRD, RDTR and RDWTEC commands.

6.9 Read Clock (RCK) (cont'd)

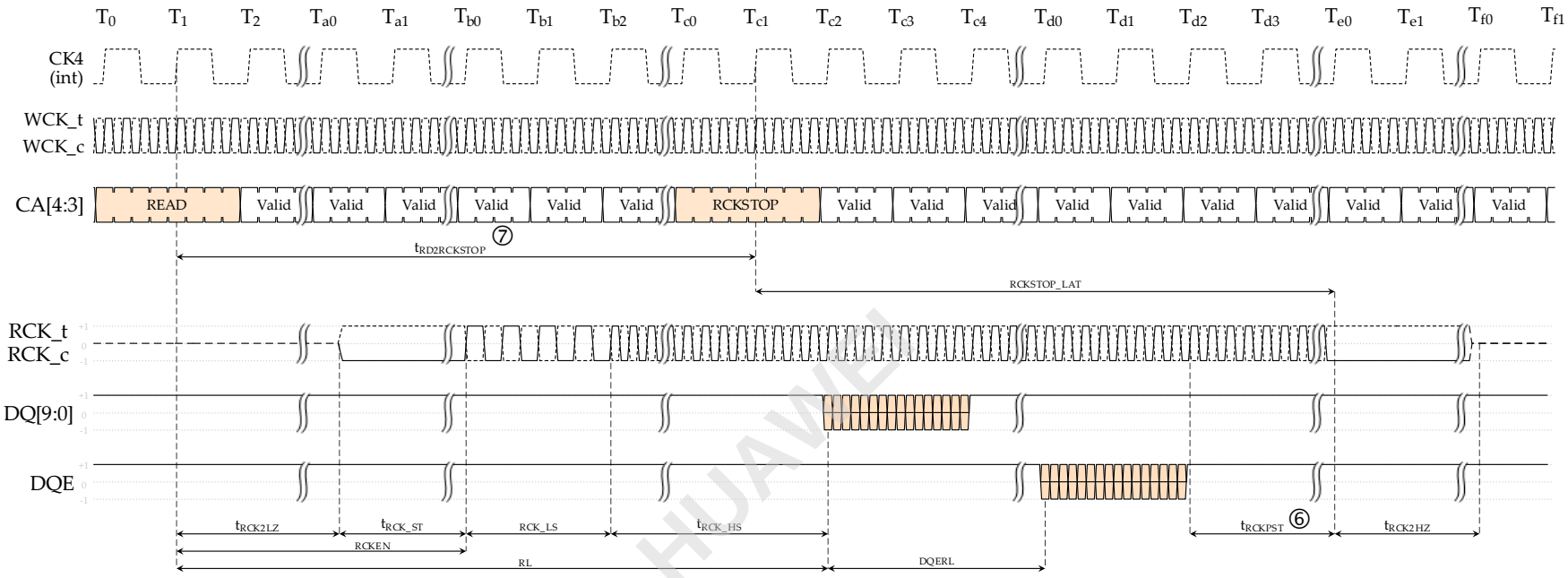


Figure 83 — Example of RCK Triggered with READ Command

6.9 Read Clock (RCK) (cont'd)

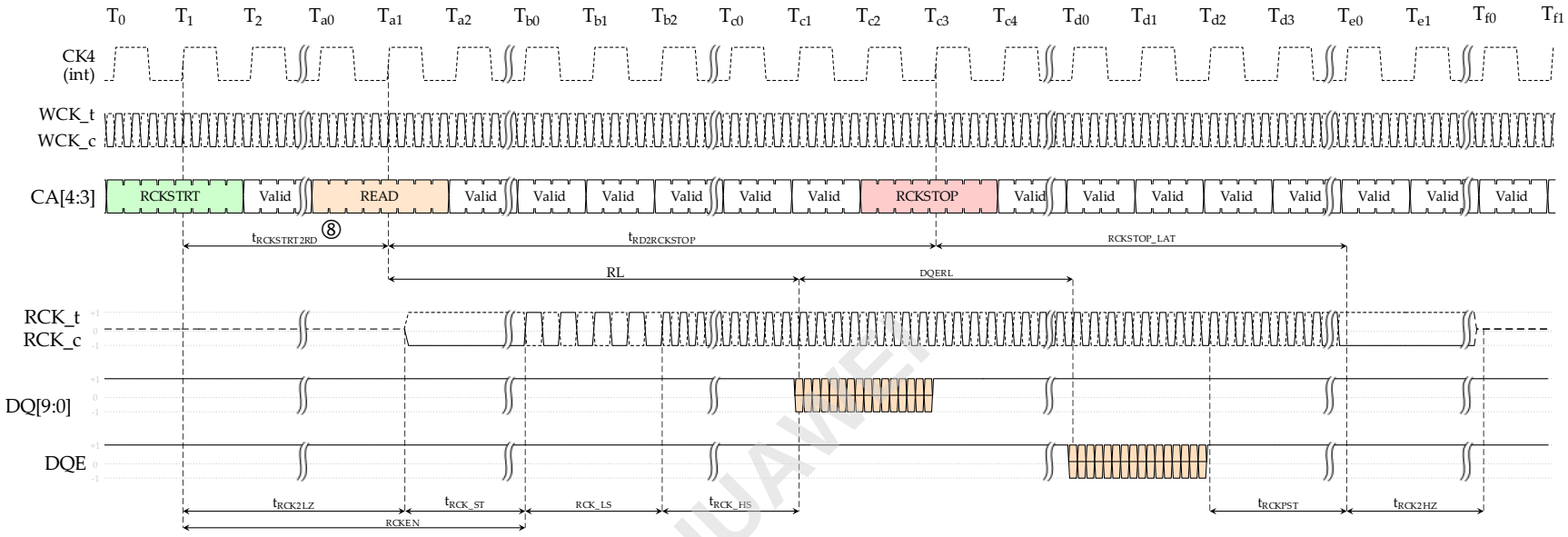
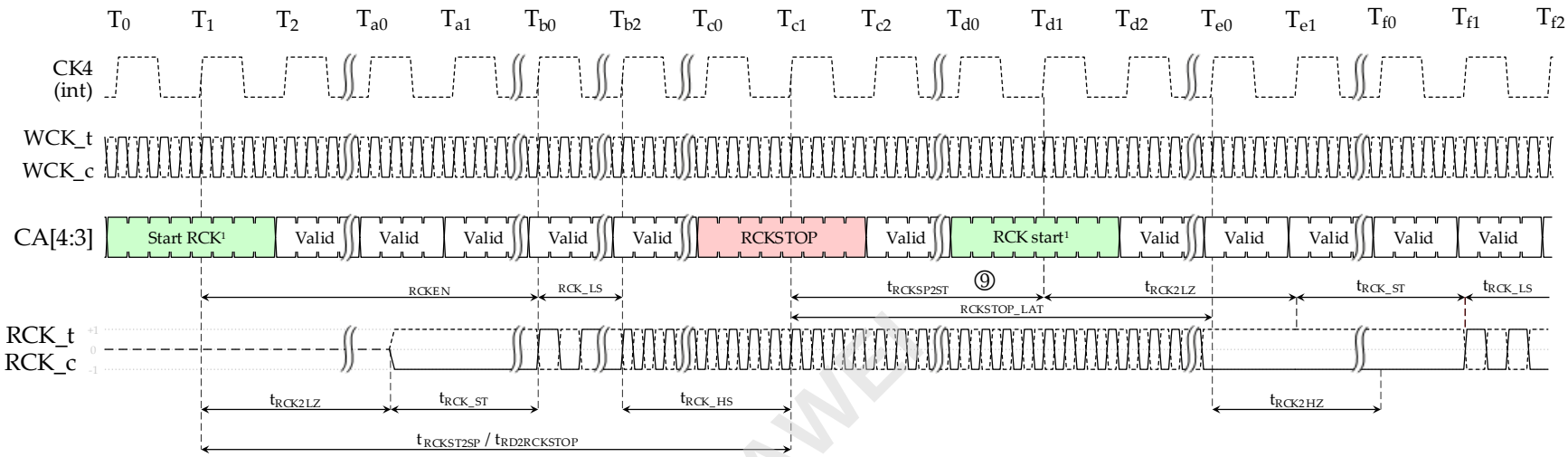


Figure 84 — Example of RCK Triggered with RCKSTRT Command Followed by a READ Command

6.9 Read Clock (RCK) (cont'd)



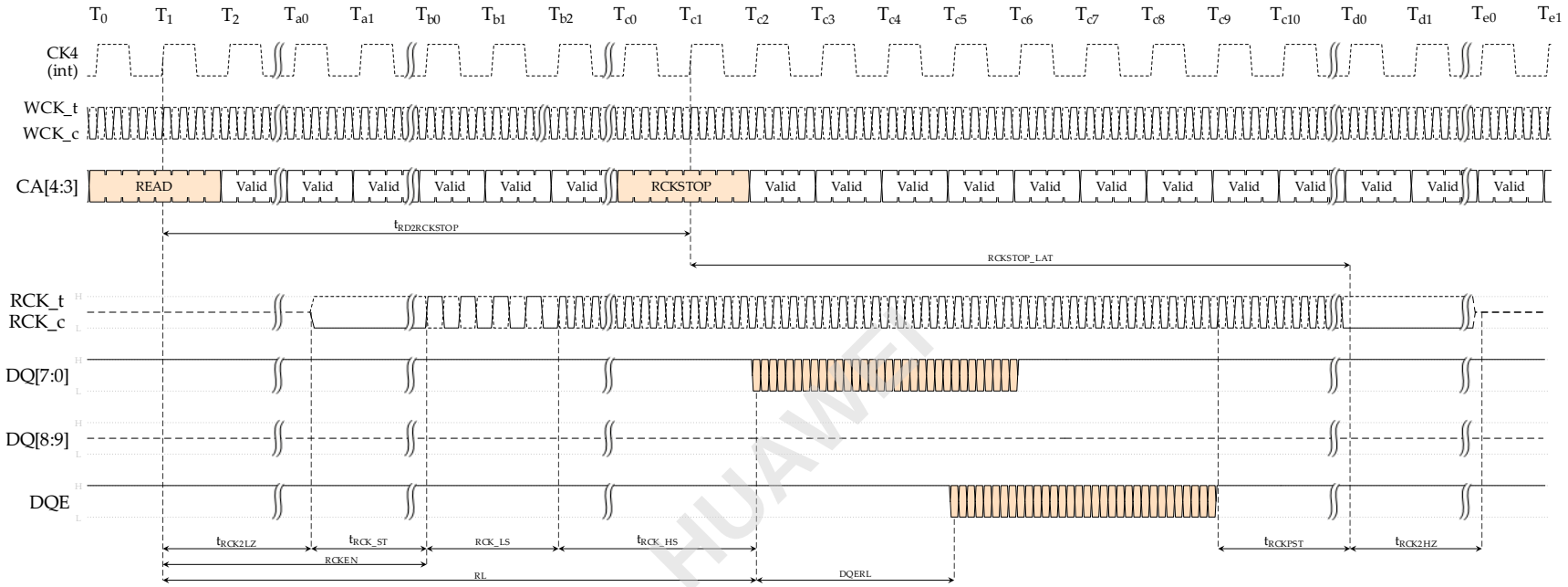
NOTE 1 "Start RCK" is the command that triggers the RCK toggling. Please note that, depending on the enabled RCKMODE, either tRCKST2SP or tRD2RCKSTOP applies, while the same tRCKSP2RCKST is valid for both RCK start/stop modes.

NOTE 2 The host must observe the minimum tRCKSP2ST(MIN) allowed delay between an RCKSTOP command and the following RCK start command to avoid pre- post- ambles collision.

NOTE 3 The drawing shows an example case where the static postamble of the previous RCK start is overlapping with the static preamble of the following RCK start, as would be the case for tRCKSP2ST(min). For a more relaxed tRCKSP2ST (longer) the tRCKLZ might expire earlier and the RCK will transition to High-Z before the tRCK_ST (static pre amble) of the next "RCK start" command starts.

Figure 85 — RCKSTOP Command Followed by RCK Start Command

6.9 Read Clock (RCK) (cont'd)



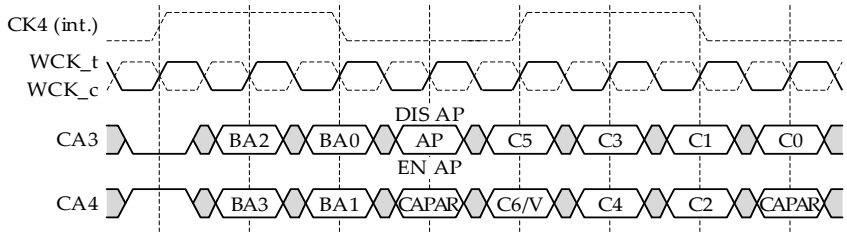
NOTE 1 DQERL = 3 is shown as example for illustration purposes.

Figure 86 — Example of RCK Triggered by READ Command in NRZ Mode

6.10 Read

Read bursts are initiated with a Read command as shown in *FIGURE 87*. The bank and column addresses are provided with the Read command and auto precharge is either enabled or disabled for that access with the AP bit (CA3-3). If auto precharge is enabled, the row being accessed is precharged after both tRTP and tRAS have been met or after the number of CK4 cycles programmed in the RAS (MR3 OP[6:0] and RTP (MR4 OP[11:8]) fields, depending on the implementation choice per DRAM vendor.

The length of the burst initiated with a Read command is sixteen symbols in PAM3 mode and thirty-two bits if in NRZ mode and the column address is unique for this burst of sixteen or thirty-two. There is no interruption nor truncation of Read bursts.



Notes:

1. BA = Bank address; C = Column Address; CAPAR = CA Parity, V = Valid (H or L but not floating), AP = Auto Precharge, EN AP = Enable Auto Precharge, DIS AP = Disable Auto Precharge
2. WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 87 — Read Command

Read timings are shown for PAM3 and NRZ modes in *FIGURE 88*. During Read bursts, the first valid data-out element will be available after the Read Latency (RL).

The Read Latency is defined as

RL (MR1 OP[5:0]) * tCK4 + tWCK2DQO

when CAPARBLK OFF (MR15 OP1 = 0) or CAPARBLK ON (MR15 OP1 = 1) and implicit CAPARBLK latency.

The Read Latency is defined as

RL (MR1 OP[5:0]) * tCK4 + CAPARBLK_LAT (MR15 OP[5:3]) * tCK4 + tWCK2DQO

when CAPARBLK ON (MR15 OP1 = 1) and explicit CAPARBLK latency.

Where RL is the number of CK4 cycles programed in MR1 OP[5:0], CAPARBLK_LAT is the number of CK4 cycles programmed in MR15 OP[5:3] and tWCK2DQO is the WCK to DQ/DQE offset as measured at the DRAM balls. The total delay is relative to the data eye initial edge averaged over DQ/DQE. The maximum skew within DQ/DQE is defined by tDQ2DQO.

GDDR7 devices support either implicit (IRA3 DQ[3:2] = 11_B); or explicit (IRA3 DQ[3:2] = 10_B); or both implicit and explicit CAPARBLK_LAT (IRA3 DQ[3:2] = 00_B). If the DRAM supports both, the host has the flexibility to program the CAPARBLK_LAT control register (MR15 OP6) to support either implicit or explicit. Refer to the *CAPAR WITH COMMAND BLOCKING (CAPARBLK)* section for more details and vendor datasheets to see which method is supported.

6.10 Read (cont'd)

If enabled, the ODT is disabled tODT_off time before the DQ data and independently the ODT on the DQE will be disabled tODT_off time before the DQE data. Upon completion of a burst, assuming no other Read command has been initiated, all DQ and DQE signals will drive a value of '+1' in PAM3 mode and a value of 'H' in NRZ mode. In addition, the ODT is enabled at a maximum of tODT_on time later on the DQ signals as well as the DQE signal as shown in *FIGURE 89*. The drive value and termination value may be different due to separately defined calibration offsets. If the ODT is disabled, the signals will drive High-Z.

Data from any Read burst may be concatenated with data from a subsequent Read command. A continuous flow of data can be maintained. The first data element from the new burst follows the last element of a completed burst. The new Read command should be issued after the previous Read command according to the tCCD or tCCDSB timing. If that Read command is to another bank, then an Activate command must precede the Read command and tRCDRD also must be met.

A Write command can be issued any time after a Read command if the bus turnaround time tRTW is met. If that Write command is to another bank, then an Activate command must precede the Write command and tRCDWR also must be met.

A PRECHARGE can be issued if both tRTP and tRAS are met or after the number of CK4 cycles programmed in the RAS (MR3 OP[6:0] and RTP (MR4 OP[11:8]) fields, depending on the implementation choice per DRAM vendor. After the PRECHARGE command, a subsequent command to the same bank cannot be issued until tRP is met.

A MRS command can be issued while a bank is open after tRD2MRS to program some registers. See the *MODE REGISTER SET (MRS)* command section for details on what registers are allowed to be programmed while a bank is active.

Table 86 — DQE State (NRZ Mode)

Case	PAM3	RDCRC	SEVERITY	POISON	WRCRC	DQE_HZ	SEV2ERR	DQE State Proposal
	MR0 OP8	MR0 OP3	MR0 OP9	MR0 OP10	MR0 OP4	MR5 OP10	MR5 OP9	
1	Disable	Disable	Disable	Disable	Disable	Enable	Disable	High-Z
2	Disable	Disable	Disable	Disable	Disable	Disable	Disable	Vendor Specific, except for driving Low
3	Disable	Disable	Enable	Disable	Disable	Enable	Enable	High-Z
4	Disable	Enable	Disable	Disable	Disable	Disable	Disable	Data (CRC)
5	Disable	Disable	Enable	Disable	Disable	Disable	Disable	Data (SEV)
6	Disable	Disable	Disable	Enable	Disable	Disable	Disable	Data (PSN)
7	Disable	Disable	Disable	Disable	Enable	Disable	Disable	Vendor Specific, except for driving Low

NOTE 1 DQE pin will be used for data transmission while GDDR7 is in read operation, otherwise DQE pin state will be ODT state or off in Case 4, 5 and 6.

NOTE 2 If more than two MR(RDCRC or SEVERITY or POSION or WRCRC) are enabled (PAM3 and DQE_HZ are disabled), DQE pin will be used for data(CRC, SEV and PSN) transmission.

6.10 Read (cont'd)

FIGURE 89 through *FIGURE 96* illustrate Read operations in PAM3 mode including:

Figure 89 — Single PAM3 Read with RDCRC and CA Parity Enabled

Figure 90 — Single PAM3 Read with RDCRC Disabled

Figure 91 — Gapless PAM3 Reads, Different Banks (tCCD = 2)

Figure 92 — Non-Gapless PAM3 Reads, Different Banks (tCCD = 3)

Figure 93 — Non-Gapless PAM3 Reads, Same Bank (tCCDSB = 4)

Figure 94 — Read to Precharge (PAM3 Mode)

Figure 95 — Read to Write (PAM3 Mode)

Figure 96 — Read to MRS (PAM3 Mode)

FIGURE 97 through *FIGURE 101* illustrate Read operations in NRZ mode including:

Figure 97 — Single NRZ Read with RDCRC and CA Parity Enabled

Figure 98 — Single NRZ Read with RDCRC, CA Parity and ODT Enabled

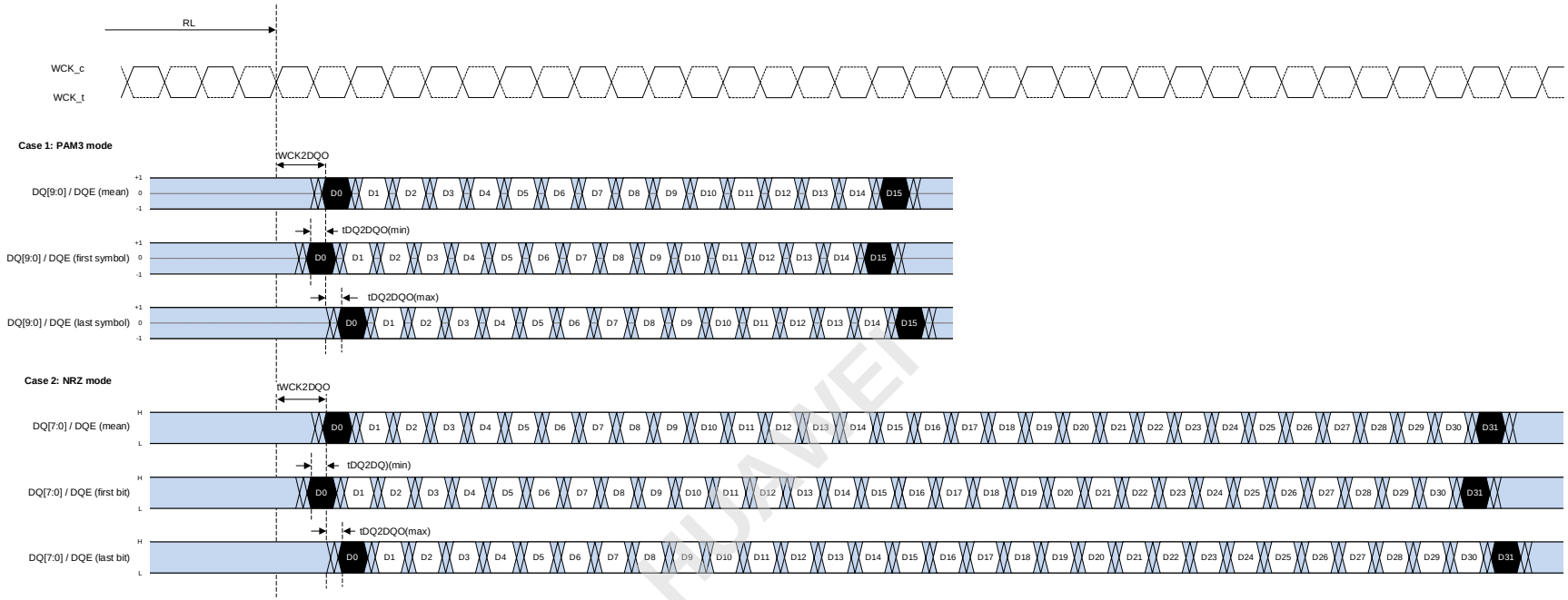
Figure 99 — Gapless NRZ Reads (tCCD = 4)

Figure 100 — Single NRZ Read with DQE Disabled (RDCRC, WRCRC, Severity, and Poison Disabled)

Figure 101 — Single NRZ Read with DQE Disabled and SEV2ERR Enabled

See the *DATA INTEGRITY* Section for details on Command Address Parity, RDCRC, Poison, Severity, and ERR signaling.

6.10 Read (cont'd)

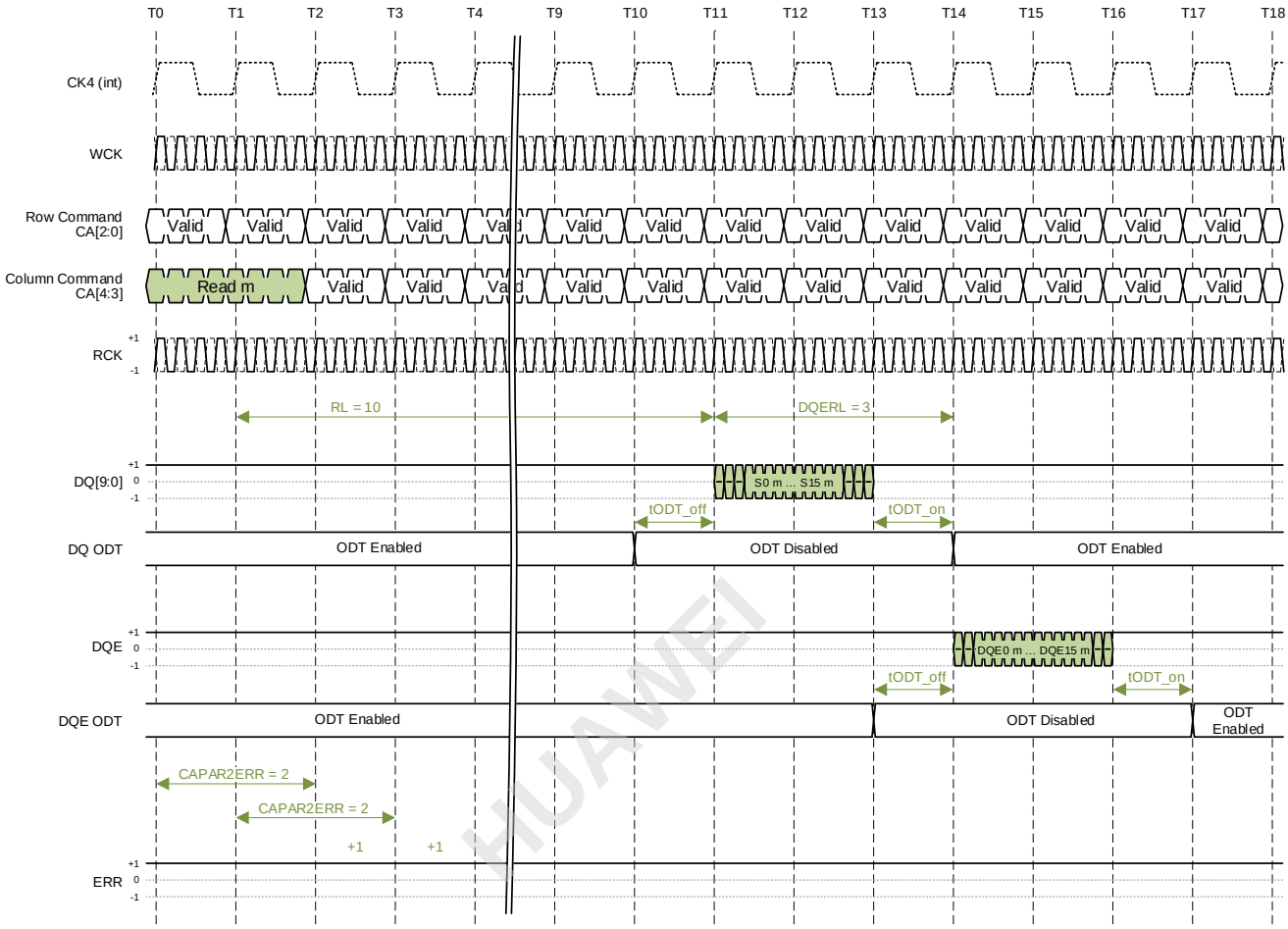


NOTES:

1. RL is the Read latency programmed in Mode Register MR1.
2. tWCK2DQO parameter values are positive numbers and could be less than 1 nCK4 as illustrated or more than 1 nCK4 depending on design implementation, and can vary across PVT. Data training is required to determine the actual tWCK2DQO value for stable Read operation.
3. tDQ2DQO defines the minimum to maximum variation of tWCK2DQO within DQ/DQE.

Figure 88 — Read Lane Timings

6.10 Read (cont'd)

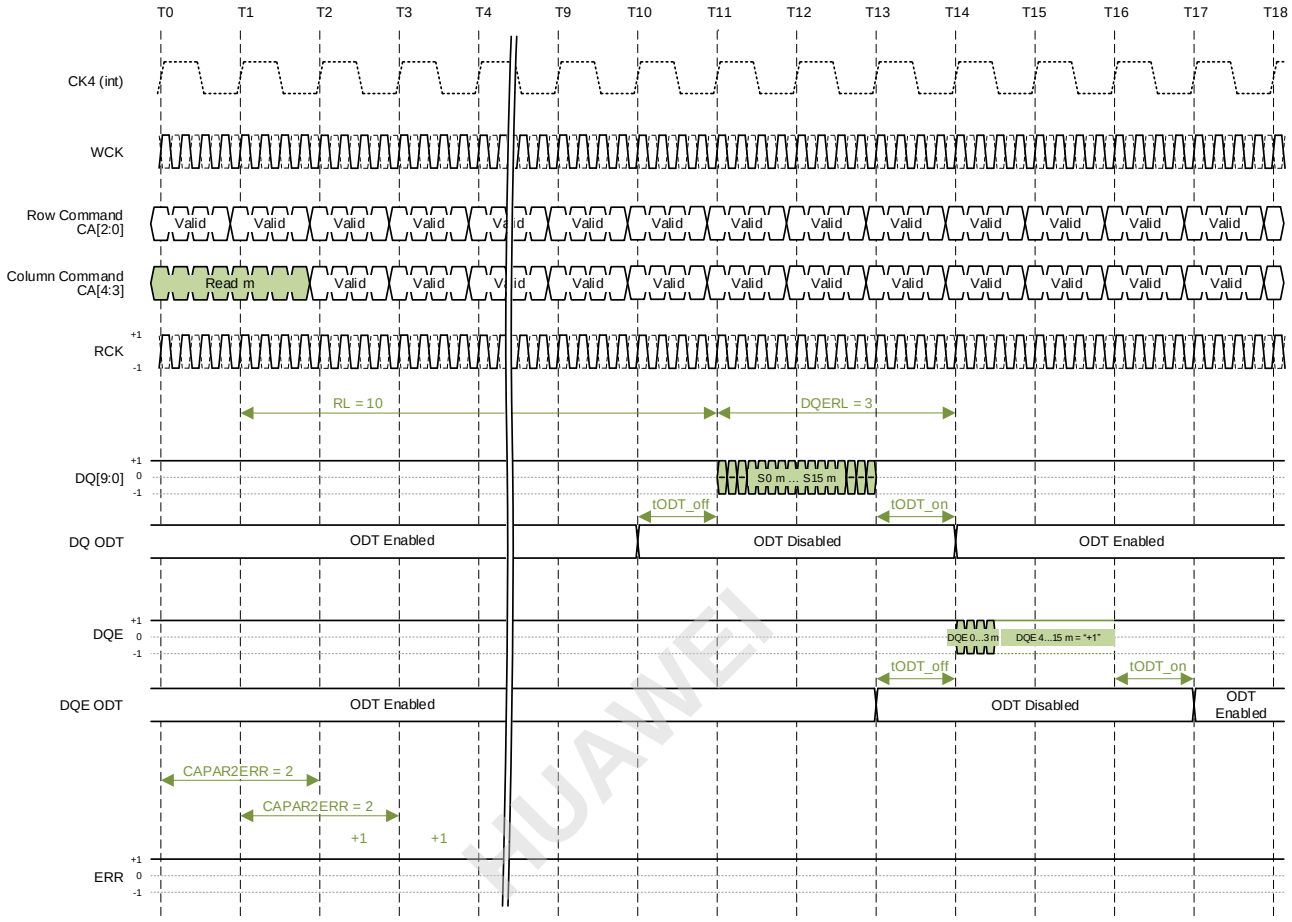


NOTES:

1. RL = 10 and DQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the READ command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCKDQO = 0 are shown for illustration purposes.
4. At T0 thru T17, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the READ command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings and toggle modes.
7. tODT_off and tODT_on = 1 tCK4 are shown as examples for illustration purposes. Actual supported values will be found in the *AC TIMINGS* section.

Figure 89 — Single PAM3 Read with RDCRC and CA Parity Enabled

6.10 Read (cont'd)

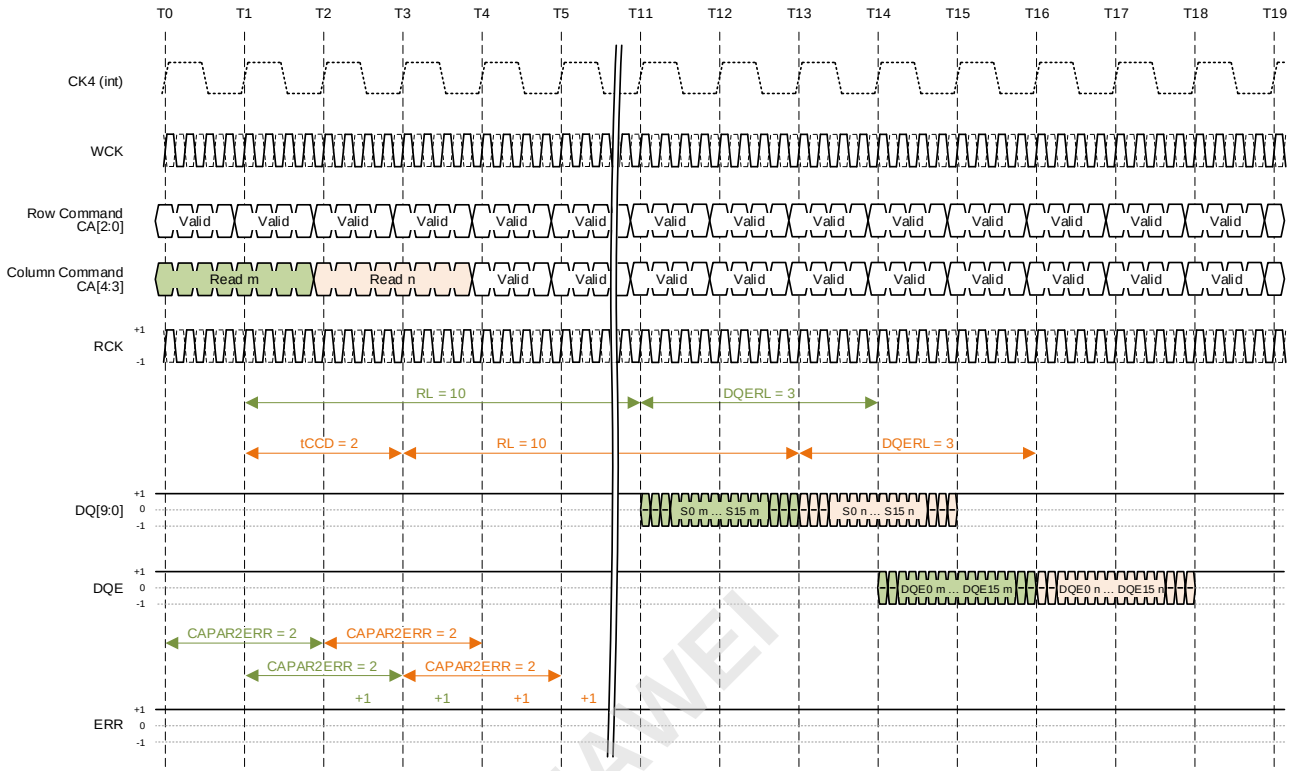


NOTES:

1. RL = 10 and DQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. When RDCRC Is disabled using MR0 OP3, DQERL = 3 is shown for illustration purposes for the Severity/Poison [1:0] and PAM3 followed by 12 symbols of +1.
3. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
4. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCKDQO = 0 are shown for illustration purposes.
5. At T0 thru T17, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the READ command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
6. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
7. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings and toggle modes.
8. tODT_off and tODT_on = 1 tCK4 are shown as examples for illustration purposes. Actual supported values will be found in the *AC TIMINGS* section.

Figure 90 — Single PAM3 Read with RDCRC Disabled

6.10 Read (cont'd)

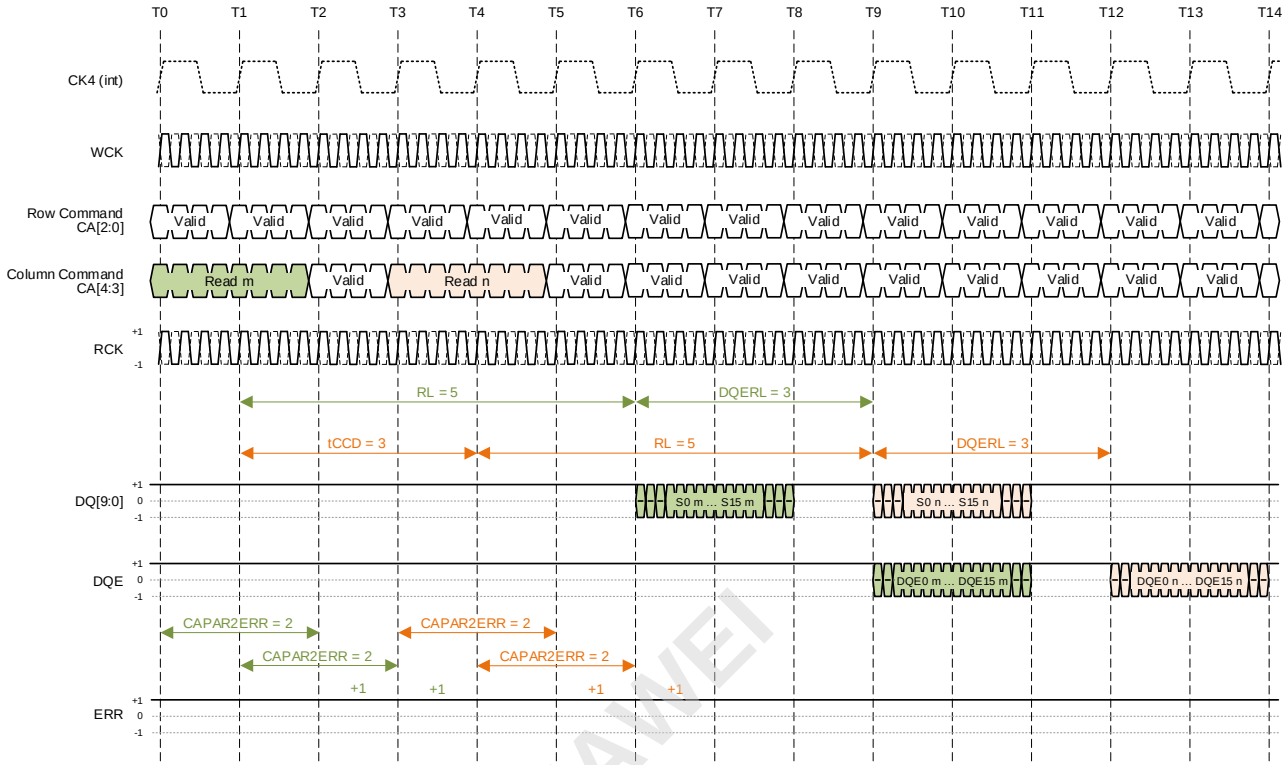


NOTES:

1. RL = 10, tCCD = 2 and DQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Read n must be to a different bank than Read m. Activate (ACT) commands are required to be issued before the Read commands and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCKDQO = 0 are shown for illustration purposes.
4. At T0 thru T18, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the READ command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 91 — Gapless PAM3 Reads, Different Banks (tCCD = 2)

6.10 Read (cont'd)

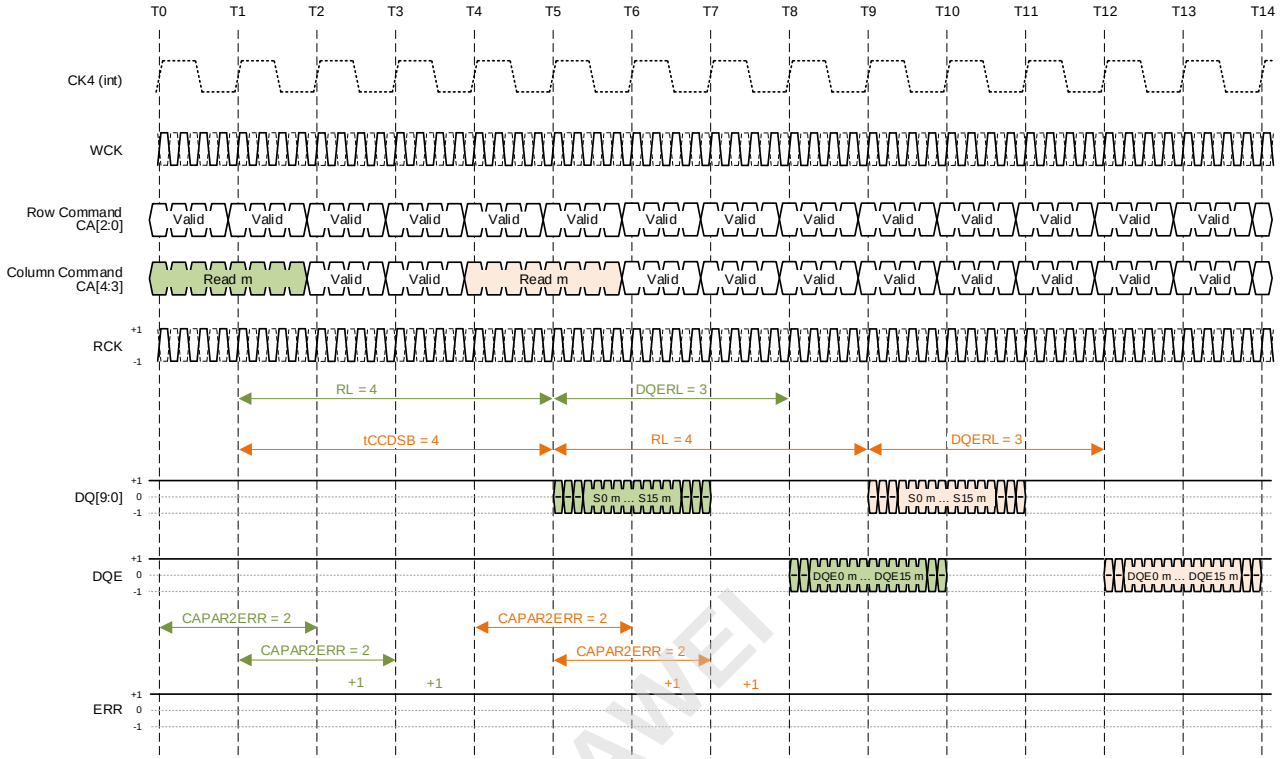


NOTES:

1. RL = 5 and DQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. Read n must be to a different bank than Read m. Activate (ACT) commands are required to be issued before the Read commands and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCKDQO = 0 are shown for illustration purposes.
4. At T0 thru T13, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the READ command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 92 — Non-Gapless PAM3 Reads, Different Banks (tCCD = 3)

6.10 Read (cont'd)

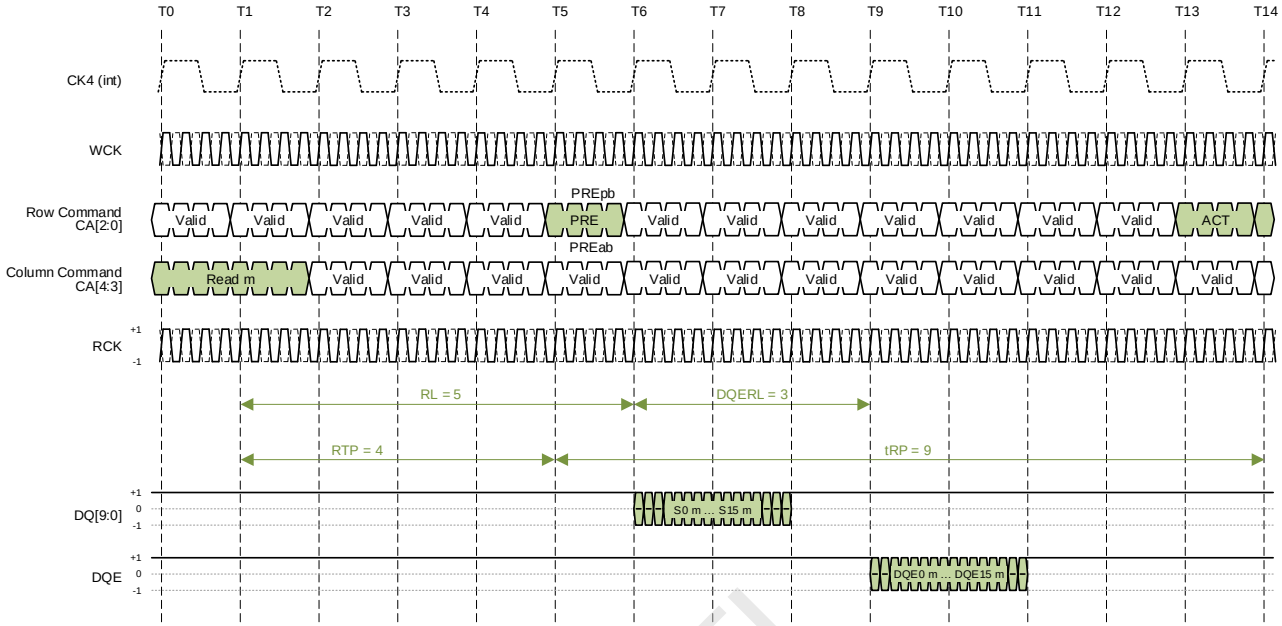


NOTES:

1. RL = 4 and DQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read commands and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCKDQO = 0 are shown for illustration purposes.
4. At T0 thru T13, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the READ command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 93 — Non-Gapless PAM3 Reads, Same Bank (tCCDSB = 4)

6.10 Read (cont'd)

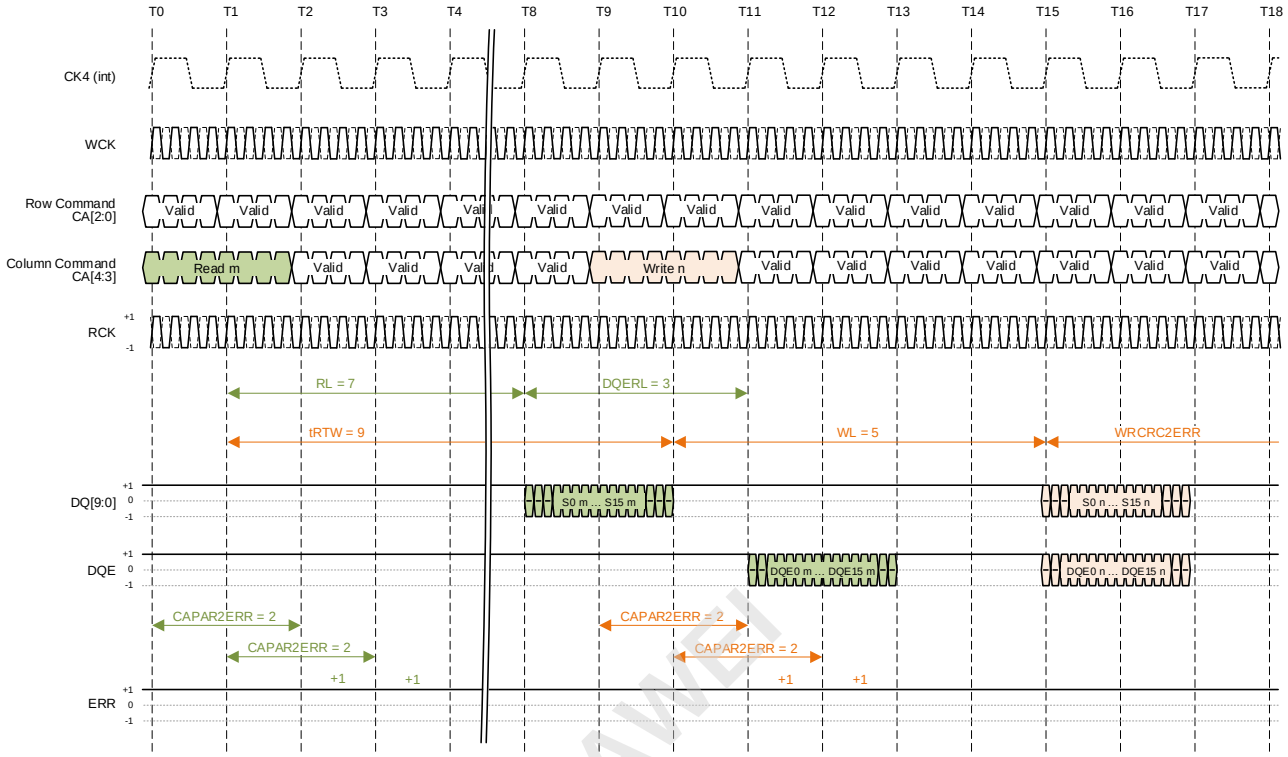


NOTES:

1. RL = 5, DQERL = 3, RTP = 4 and tRP = 9 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCKDQO = 0 are shown for illustration purposes.
4. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings and toggle modes.
5. GDDR7 supports both a mode register (RTP) and an AC timing definition (tRTP) for Read to Precharge. RTP must be programmed to RU(tRTP/tCK4) or greater. If the DRAM does not support the MR definition the register settings will be ignored.

Figure 94 — Read to Precharge (PAM3 Mode)

6.10 Read (cont'd)

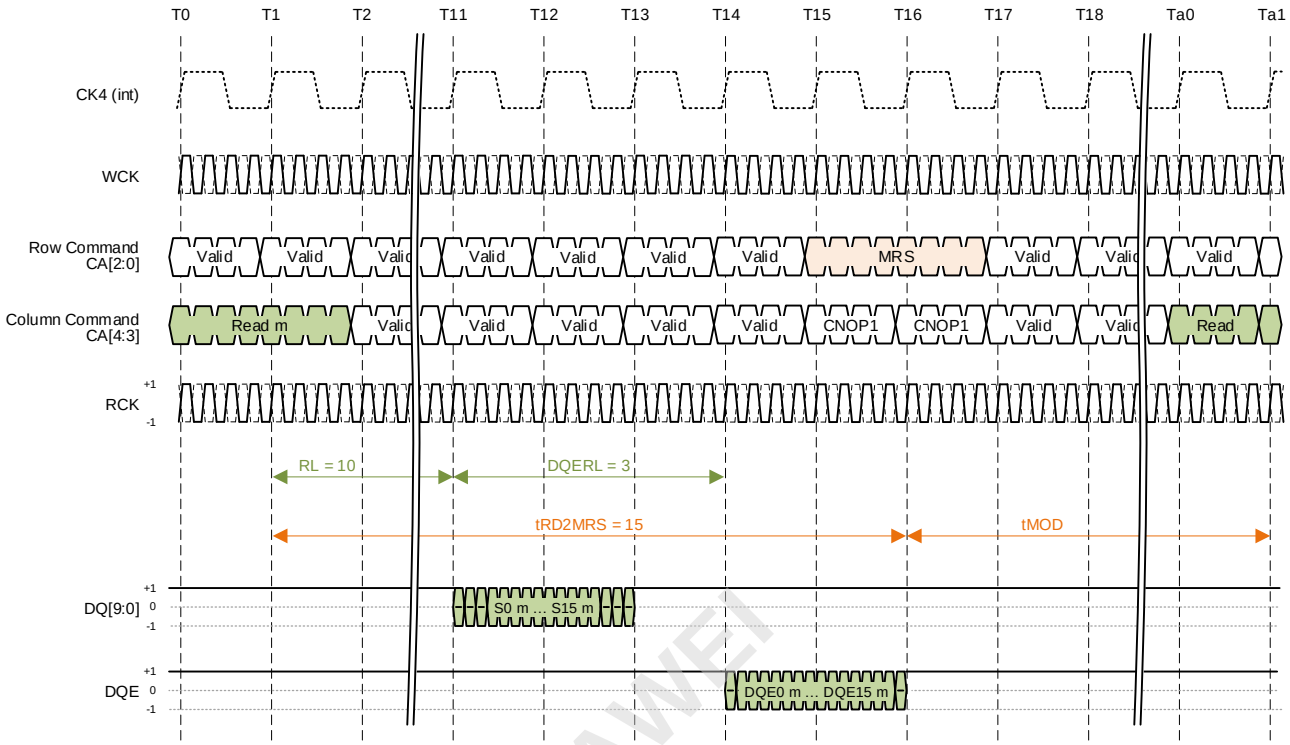


NOTES:

1. RL = 7, tRTW = 9, WL = 5 and DQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met. If the Write is to a different bank then an Activate (ACT) command is required to be issued before the Write and tRCDWR must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. Write Latency = WL * tCK4 + tWCK2DQI. tWCK2CA, tWCK2DQI and tWCK2DQO = 0 are shown for illustration purposes.
4. At T0 thru T17, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Read and Write commands are shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 95 — Read to Write (PAM3 Mode)

6.10 Read (cont'd)

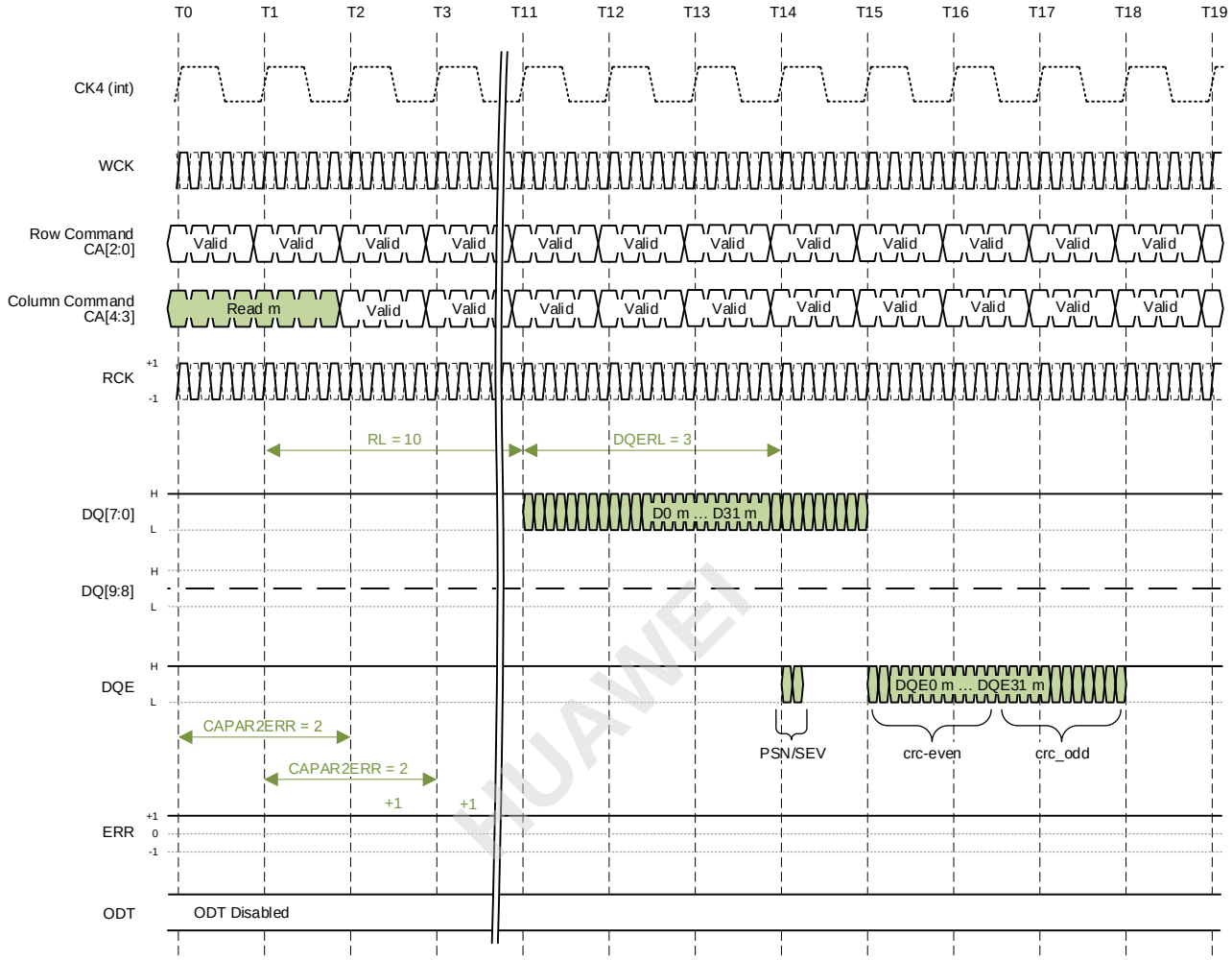


NOTES:

1. RL = 10 and tDQERL = 3 are shown as examples for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. tRD2MRS = tRD2MRS(min) = 15 is shown as example for illustration purposes. tRD2MRS(min) = RL + DQERL + BL/8.
3. A MRS command while a bank is active is only legal to certain MR. See the *MODE REGISTERS* section for details.
4. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
5. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 96 — Read to MRS (PAM3 Mode)

6.10 Read (cont'd)

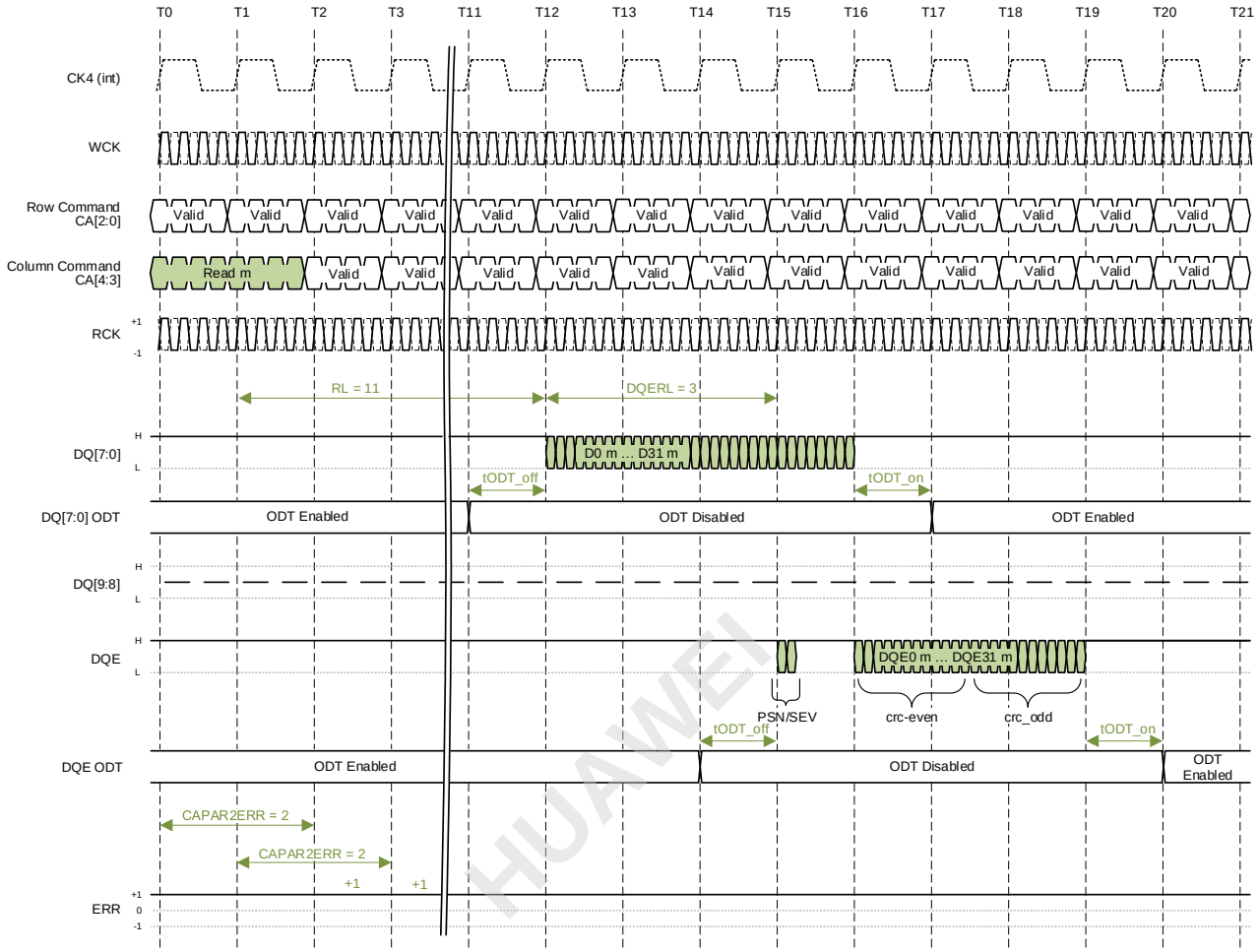


NOTES:

1. RL = 10 and DQERL = 3 are shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
4. At T0 thru T11, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Read command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQ[9:8] are turned off on the GDDR7 device.
6. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
7. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQE bit positions [7:2] = 6'b111111.
8. When the PAM3 register (MR0 OP8) is set to NRZ mode, the CRC is PAM3 coded and the binary representation is transmitted on the DQE signal at bit positions [19:8] for crc_even and bit positions [31:20] for crc_odd.
9. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 97 — Single NRZ Read with RDCRC and CA Parity Enabled, ODT Disabled

6.10 Read (cont'd)

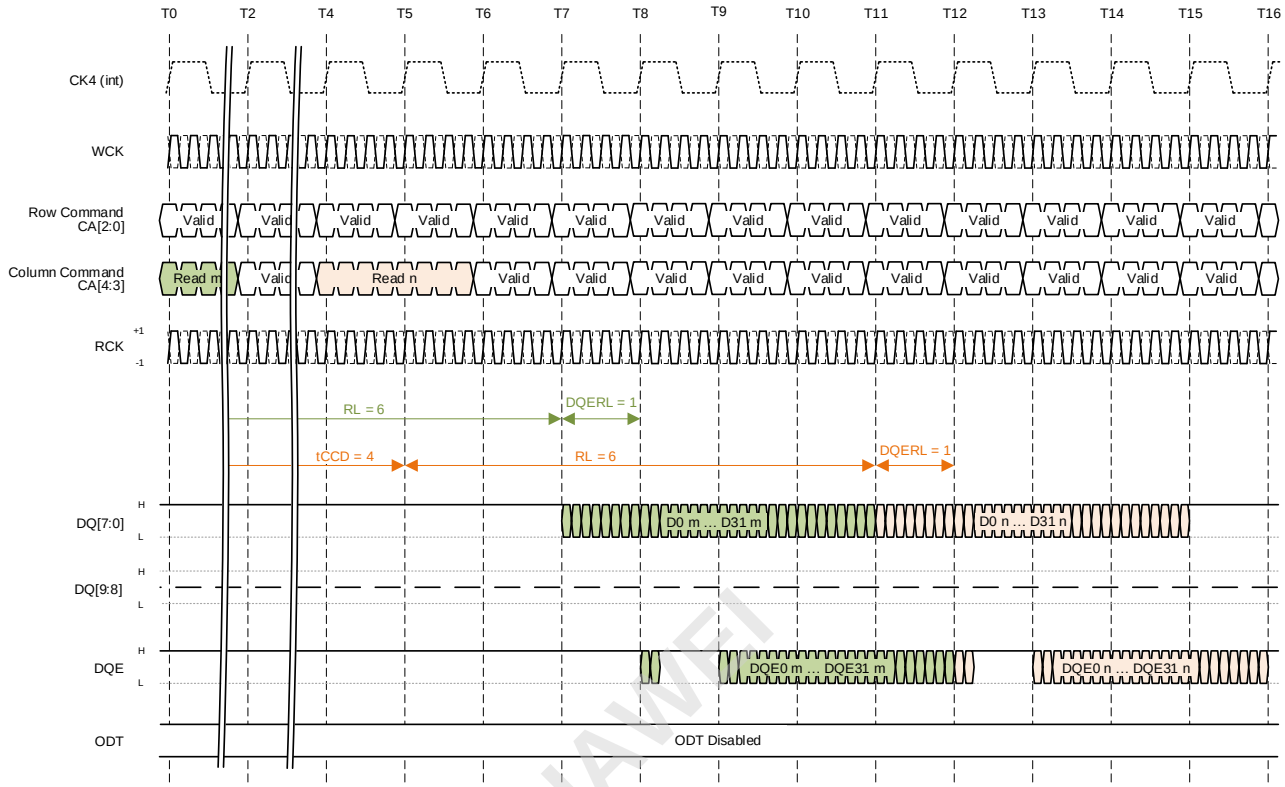


NOTES:

1. RL = 11 and DQERL = 3 are shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
4. At T0 thru T11, the ERR output has +1 (no error) or -1 (CA Parity error) as possible outcomes as only evaluating CA Parity. No error (+1) for the Read command is shown for illustration purposes. See the *COMMAND ADDRESS PARITY (CAPAR)* section for more details.
5. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQ[9:8] are turned off on the GDDR7 device.
6. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
7. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQE bit positions [7:2] = 6'b111111.
8. When the PAM3 register (MR0 OP8) is set to NRZ mode, the CRC is PAM3 coded and the binary representation is transmitted on the DQE signal at bit positions [19:8] for crc_even and bit positions [31:20] for crc_odd.
9. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings and toggle modes.
10. tODT_off and tODT_on = 1 tCK4 are shown as examples for illustration purposes. Actual supported values will be found in the *AC TIMINGS* section.

Figure 98 — Single NRZ Read with RDCRC, CA Parity and ODT Enabled

6.10 Read (cont'd)

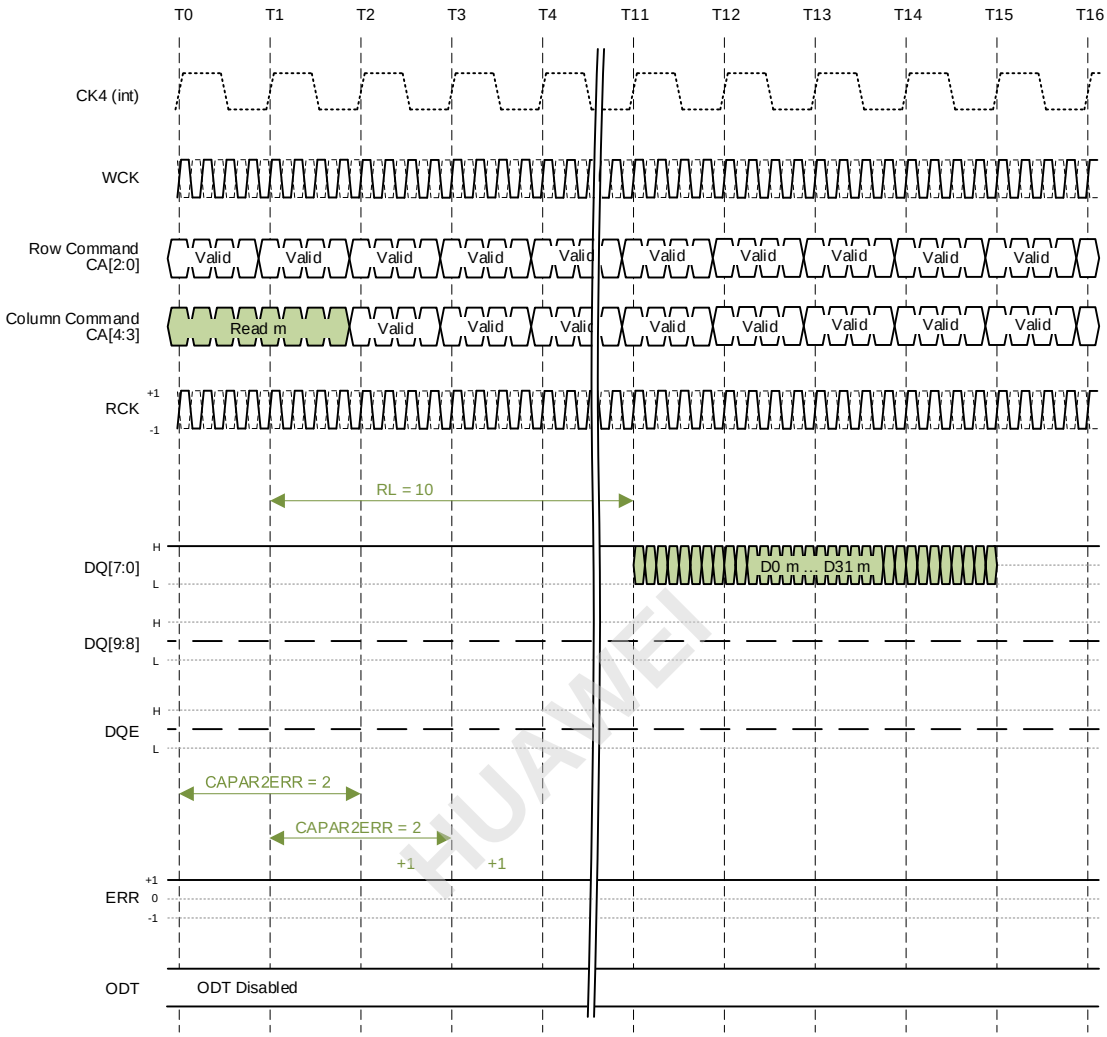


NOTES:

1. RL = 6, DQERL = 1 and tCCD = 4 are shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the first Read command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
4. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQ[9:8] are turned off on the GDDR7 device.
5. Gapless Reads in NRZ mode could be to any bank as tCCD = tCCDSB = 4.
6. Read commands are 2-cycle commands but are shown with time breaks for illustration purposes.
7. When the PAM3 register (MR0 OP8) is set to NRZ mode, DQE bit positions [7:2] = 6'b111111.
8. When the PAM3 register (MR0 OP8) is set to NRZ mode, the CRC is PAM3 coded and the binary representation is transmitted on the DQE signal at bit positions [19:8] for crc_even and bit positions [31:20] for crc_odd.
9. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 99 — Gapless NRZ Reads (tCCD = 4)

6.10 Read (cont'd)

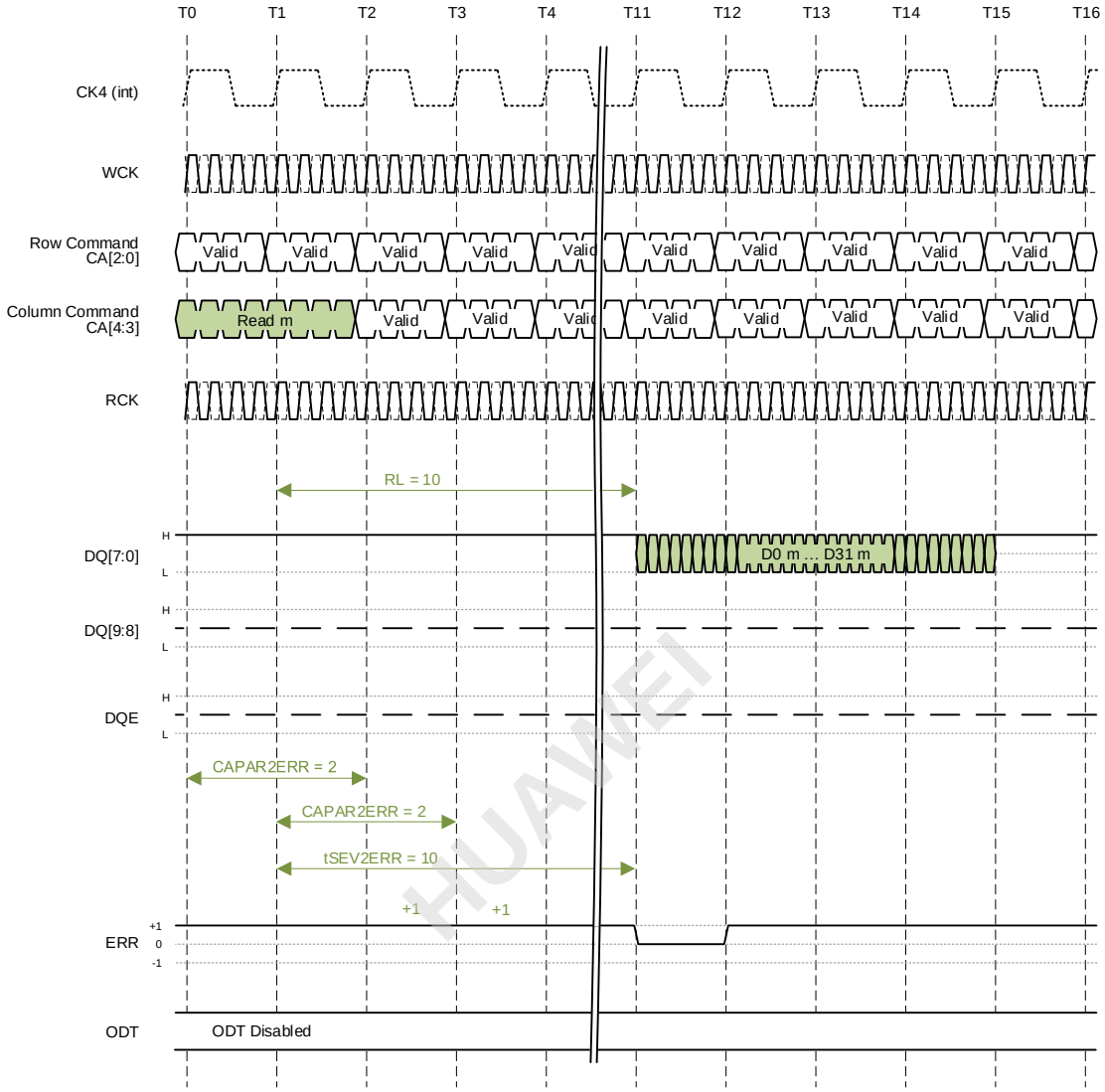


NOTES:

1. RL = 10 is shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
4. DQE is disabled using MR5 OP10 (DQE_HZ). DQE can be turned off when RDCRC (MR0 OP3), WRCRC (MR0 OP4), Severity (MR0 OP9) and Poison (MR0 OP10) are disabled in the Mode Registers. If the host still requires Severity, MR5 OP9 (DQE_SEV_ERR) enables the host to receive severity information on the ERR signal.
5. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
6. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 100 — Single NRZ Read with DQE Disabled (RDCRC, WRCRC, Severity, and Poison Disabled)

6.10 Read (cont'd)



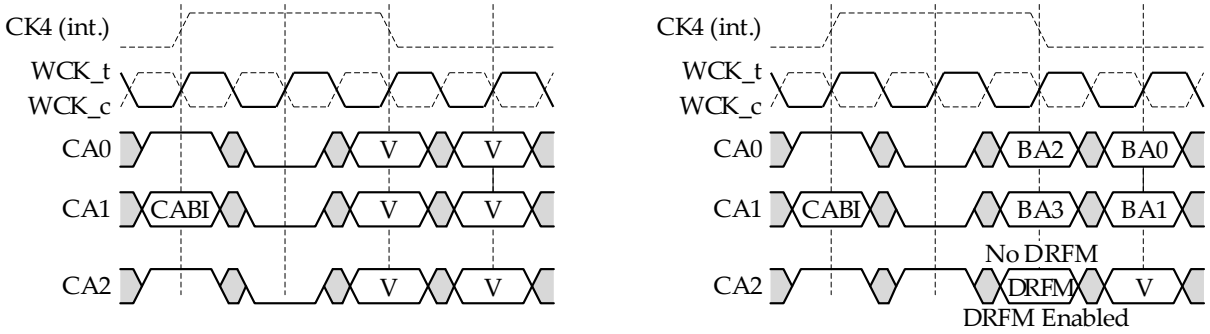
NOTES:

1. RL = 10 and tSEV2ERR =10 are shown as an example for illustration purposes. Actual supported values will be found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. An Activate (ACT) command is required to be issued before the Read command and tRCDRD must be met.
3. Read Latency = RL * tCK4 + tWCK2DQO. tWCK2CA and tWCK2DQO = 0 are shown for illustration purposes.
4. SEV2ERR mode is enabled using MR5 OP9 and requires Severity to be on (MR0 OP9) and the PAM3 mode register to be in NRZ mode. RDCRC (MR0 OP3), WRCRC (MR0 OP4) and Poison (MR0 OP10) must be disabled. DQE is disabled using MR5 OP10 (DQE_HZ).
5. A Severity Error (UE) at cycle T11 is shown for illustration purposes.
6. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO. For illustration purposes, CAPAR2ERR is shown with 2 nCK4 digital and 0 ns analog output delay.
7. RCK is shown differential, full swing with continuous toggle for illustration purposes. See the *READ CLOCK (RCK)* section for details on the RCK options, timings, and toggle modes.

Figure 101 — Single NRZ Read with DQE Disabled and SEV2ERR Enabled

6.11 Row Precharge

The Precharge command is used to deactivate the open row in a particular bank (PREpb) as selected by BA[3:0], or the open row in all banks (PREab) (see *FIGURE 102*). The bank(s) will be available for a subsequent row access a specified time t_{RP} after the Precharge command is issued as illustrated in *FIGURE 66*.



Precharge All-Banks (PREab) **Precharge Per-Bank (PREpb)**

NOTE 1 BA = Bank Address; CABI = Command/Address Bus Inversion; DRFM = Directed Refresh Management; V = Valid (H or L but not floating).

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

NOTE 3 The DRFM bit is only evaluated when DRFM is enabled in MR8 OP2, otherwise the bit is Don't Care. Refer to the *DIRECTED REFRESH MANAGEMENT (DRFM)* section for more details.

Figure 102 — Precharge Commands

Once a bank has been precharged, it is in the idle state and must be activated prior to any Read or Write command being issued to that bank. A Precharge command is allowed if there is no open row in that bank (idle state), or if the previously open row is already in the process of precharging. However, the precharge period shall be determined by the most recent Precharge command issued to the bank. Sequences of Precharge commands must be spaced by at least tPPD.

CA parity is evaluated with the Precharge command when the parity calculation is enabled in MR15 OP0.

6.11.1 Auto Precharge

Auto Precharge is a feature which performs the same individual bank precharge function as described above, but without requiring an explicit Precharge command. This is accomplished by issuing a Read with Auto Precharge (RDA) or Write with Auto Precharge (WRA) command instead of a regular Read or Write. A row precharge in the bank that is addressed with the RDA or WRA command is automatically initiated once the applicable timings tWR, tRAS and tRTP have been met as described in the Write and Read sections and in *TABLE 87*. Auto Precharge is non persistent in that it is either enabled or disabled for each individual Read or Write command.

Auto Precharge ensures that a precharge is initiated at the earliest valid stage after the Read or Write command. The user must not issue another command to the same bank until the precharge time tRP is completed. This is determined as if an explicit Precharge command was issued at the earliest possible time.

6.11 Row Precharge (cont'd)

Table 87 — Auto Precharge Enabled/Disabled and PREab / PREpb Timings

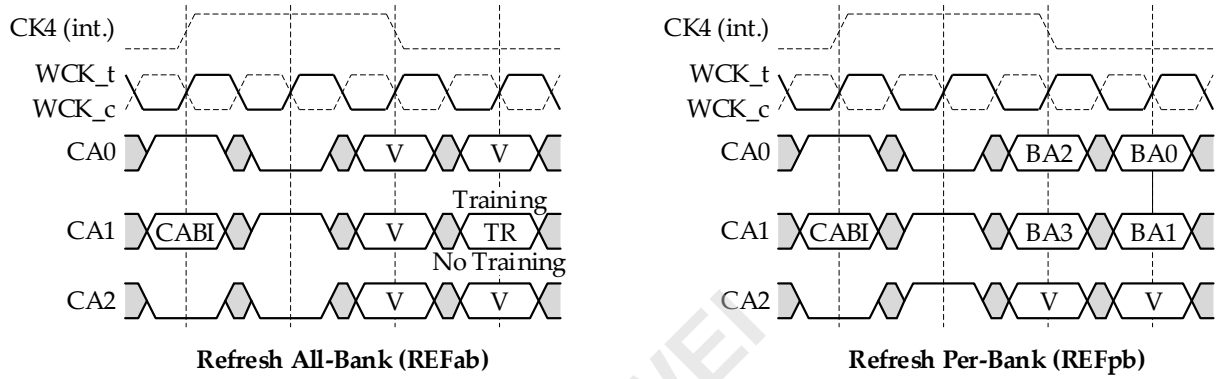
From Command	To Command	Minimum Delay Between "From Command" to "To Command"	Unit	Notes
Read	PREpb (same bank)	$RU(t_{RTP}/t_{CK})$	nCK4	2
	PREpb (different bank)	0	nCK4	3
	PREab	$RU(t_{RTP}/t_{CK})$	nCK4	2
	Write or Write w/ AP (any bank)	$RU(t_{RTW}/t_{CK})$	nCK4	2
	Read or Read w/ AP (same bank)	t_{CCDSB}	nCK4	
	Read or Read w/ AP (different bank)	t_{CCD}	nCK4	
Read with AP	PREpb (different bank)	0	nCK4	3
	PREab	$RU(t_{RTP}/t_{CK})$	nCK4	2
	ACT or REFpb/RFMpb (same bank)	$RTP + RU(t_{RP}/t_{CK})$	nCK4	2
	Write or Write w/ AP (same bank)	Illegal		
	Write or Write w/ AP (different bank)	$RU(t_{RTW}/t_{CK})$	nCK4	2
	Read or Read w/ AP (same bank)	Illegal		
	Read or Read w/ AP (different bank)	t_{CCD}	nCK4	
Write	PREpb (same bank)	$WL + BL/8 + RU(t_{WR}/t_{CK})$	nCK4	1, 2
	PREpb (different bank)	0	nCK4	3
	PREab	$WL + BL/8 + RU(t_{WR}/t_{CK})$	nCK4	1, 2
	Write or Write w/ AP (same bank)	t_{CCDSB}	nCK4	
	Write or Write w/ AP (different bank)	t_{CCD}	nCK4	
	Read (same bank)	$WL + BL/8 + RU(t_{WTR}/t_{CK})$	nCK4	1, 2
	Read w/ AP (same bank)	$WL + BL/8 + MAX(RU(t_{WR}/t_{CK}) - RU(t_{RTP}/t_{CK}), RU(t_{WTR}/t_{CK}))$	nCK4	1, 2
	Read or Read w/ AP (different bank)	$WL + BL/8 + RU(t_{WTR}/t_{CK})$	nCK4	1, 2
Write with AP	PREpb (different bank)	0	nCK4	
	PREab	$WL + BL/8 + RU(t_{WR}/t_{CK})$	nCK4	1, 2
	ACT or REFpb/RFMpb (same bank)	$WL + BL/8 + WR + RU(t_{RP}/t_{CK})$	nCK4	1, 2
	Write or Write w/ AP (same bank)	Illegal		
	Write or Write w/ AP (different bank)	t_{CCD}	nCK4	
	Read or Read w/ AP (same bank)	Illegal		
	Read or Read w/ AP (different bank)	$WL + BL/8 + RU(t_{WTR}/t_{CK})$	nCK4	1, 2
PREpb	PREpb (any bank)	t_{PPD}	nCK4	
	PREab	t_{PPD}	nCK4	
	ACT (different bank), MRS or REFpb (different bank)	t_{RPD}	nCK4	
PREab	PREpb (any bank) or PREab	t_{PPD}	nCK4	

NOTE 1 BL = Burst length; WL = Write latency; RTP = Read-to-Precharge.
NOTE 2 RU = round up to the next integer.
NOTE 3 Commands can be issued in parallel.

6.12 Refresh

Refresh commands are used during normal operation. Since “data” is stored as 0s and 1s in capacitors in a DRAM, and the capacitors leak charge over time. A Refresh command is issued periodically to restore (refresh) the electrical charge in the capacitors. Each Refresh command results in one or more activate operations to a selected row or rows, followed by a self-timed precharge to close the rows opened during the activate. Refresh commands are non-persistent, so they must be issued each time a refresh is required. GDDR7 SGRAMs require Refresh commands to be issued at an average periodic interval of tREFI. GDDR7 SGRAMs support two commands to perform refreshes as shown in *FIGURE 103*:

- the Refresh All-Bank (REFab) command initiates a refresh cycle on all banks simultaneously.
- the Refresh Per-Bank (REFpb) command initiates a refresh cycle on the selected bank only.



NOTE 1 BA = Bank Address; CABI = Command/Address Bus Inversion; TR = Data Training; V = Valid (H or L but not floating).

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 103 — Refresh All-Bank (REFab) and Refresh Per-Bank (REFpb) Commands

6.12.1 Refresh All-Bank (REFab) Command

The Refresh All-Bank (REFab) command can only be issued when all banks are precharged with tRP satisfied. RCK and data output timings may not be guaranteed during tKO which can impact the last Read transaction. In that case, a REFab command shall only be issued when all previous Read operations have completed. For Reads, a burst completion is defined as when the last data element including CRC has been transmitted on the data outputs (tRDREFab timing). The command is nonpersistent, so it must be issued each time a refresh is required. The refresh addressing is generated by the internal refresh controller. This makes the address bits "Don't Care" during a REFab command. A minimum time tRFCab is required between two REFab commands or a REFab command and any subsequent access command (ACT, SRE, SLE or SRSE) after the refresh operation. The banks are in idle state after completion of the REFab command.

REFab cycles are required at an average periodic interval of tREFI(max). To allow for improved efficiency in scheduling and switching between tasks, some flexibility in the absolute refresh interval is provided. A maximum of 8 REFab commands can be postponed during operation of the device, meaning that at no point in time more than a total of 8 REFab commands are allowed to be postponed. In case that 8 REFab commands are postponed in a row, the resulting maximum interval between the surrounding REFab commands is limited to 9 × tREFI (see *FIGURE 104*). At any given time, a maximum of 9 REFab commands can be issued within tREFI.

CA parity is evaluated with the REFab command when the parity calculation is enabled in MR15 OP0.

6.12.1 Refresh All-Bank (REFab) Command (cont'd)

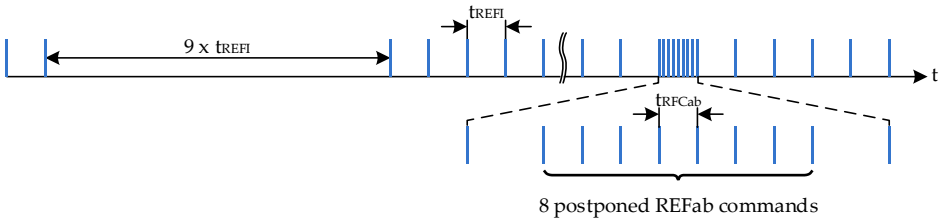


Figure 104 — Postponing Refresh All-Bank Commands (Example)

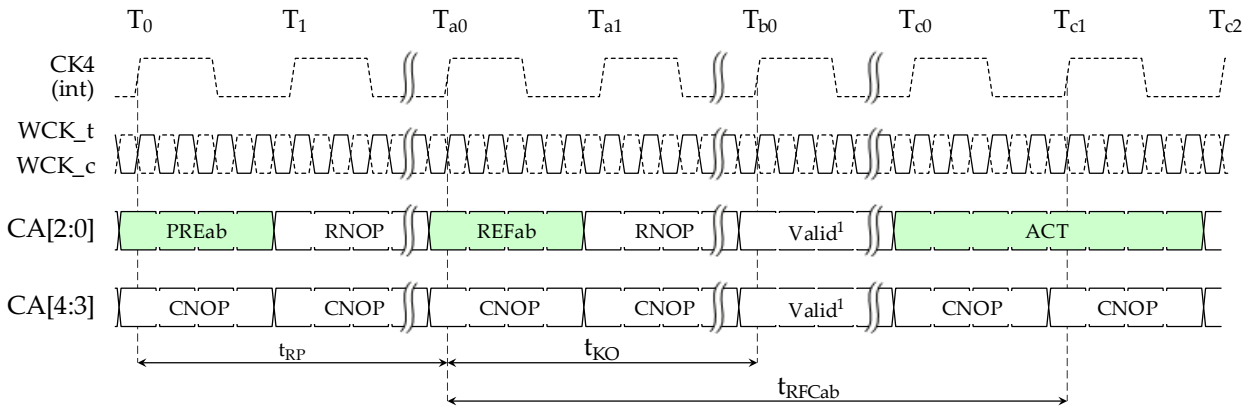
REFab commands must be issued during normal operation at a minimum rate of tABREF to allow impedance updates from the auto-calibration engine to occur. As these impedance updates may occur with any REFab command, it is safe to only issue RNOP2 and CNOP2 commands during tKO period after to prevent false command, address or data latching resulting from impedance updates. If RCK is running, RCK output timings may not be guaranteed during tKO. If the host is concerned about impact from RCK timings, it is recommended to stop RCK (by RCKSTOP or RCK disable) before a REFab when calibration is enabled(default).

When the TR bit is set to H with the REFab command, LDFF, WRTR, RDTR and RDWTEC commands are allowed at time tKO after the REFab command, which enable (incremental) data training to occur in parallel with the internal refresh operation and thus without loss of performance on the interface as illustrated in *FIGURE 105*. See *DATA TRAINING* for details. The data training is allowed to continue even when the tRFCab timing from one REFab command has elapsed; the data training may span the duration of multiple consecutive REFab commands, however, the time tKO shall be observed after each REFab command, unless calibration updates are disabled in MR5 OP11.

GDDR7 SGRAMs also allow Info Read (IRD) and CATE commands to be issued at time tKO after the REFab command. The IRD command requires that the TR bit is set to H with the REFab command. See *INFO READ* and *COMMAND ADDRESS BUS TRAINING* for details.

GDDR7 SGRAMs may also repurpose REFab commands for internal error check and scrub (ECS) cycles when this function is enabled by the ECS_ON bit MR22 OP11. The period of these ECS events (tECSint timing) is dependent on the channel density. For up to 16Gbit per channel, tECSint is larger than the tABREF timing. ECS operation in this case does not require any additional REFab commands to be issued by the host than those already required for regular calibration updates, as long as the host issues enough REFab command with TR=L to meet tECSint, as ECS operations only occur when TR=L and calibration occurs regardless of the TR bit. For example, the host can issue only REFab TR=H to meet tABREF. In this case none of the REFab commands will count towards meeting tECSint and the host will need to issue additional REFab TR=L to meet tECSint. However, if the host issues only REFab TR=L every 1ms, then both tABREF and tECSint are met. For 24Gbit per channel and larger, tECSint is smaller than the tABREF timing. In this case, additional REFab commands beyond those already required for regular calibration updates are required. The host must ensure enough REFab commands with TR=L are issued to ensure tECSint is met. See *TABLE 108* for tECSint values per channel density.

6.12.1 Refresh All-Bank (REFab) Command (cont'd)



NOTE 1 Valid commands comprise IRD, PDE, MRS, RCKSTRT, RCKSTOP, CATE for CA Bus Training, and LDFF, WRTR, RDTR or RDWTEC commands for data training only for TR = H. PDE command can also be issued at time tKO after the REFab command regardless of TR bit. Once host issues PDE during REFab, the host can only issue a PDX command during the tRFCab period. Subsequent commands must not be issued until both tRFCab and tXP have expired in this case.

NOTE 2 ACT, SRE, SLE and SRSE commands require tRFCab. ACT shown as an example for illustration purposes.

NOTE 3 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 105 — REFab Timings

6.12.2 Refresh Per-Bank (REFpb) Command

The REFpb command provides an alternative for the refresh. The REFpb command initiates a refresh cycle on a single bank selected by BA[3:0] while accesses to other banks including writes and reads are not affected. The command can only be issued when the selected bank is precharged with tRP satisfied. The command is non-persistent, so it must be issued each time a refresh is required. The refresh addressing is generated by the internal refresh controller. This makes the row address bits "Don't Care" during a REFpb command.

A minimum time tRRD is required between an Activate command and a REFpb command to a different bank. A minimum time tRREFD is required between any two REFpb commands (see below for an exception requiring tRFCpb), and between a REFpb command and an Activate command to a different bank. A minimum time tRFCpb is required between a REFpb command and an Activate command to the same bank that follows. Refer to *TABLE 89* for all REFpb related timings.

REFpb commands can be issued in any order. After all 16 banks have been refreshed using the REFpb command, and after waiting for at least tRFCpb, the internal refresh counter is incremented, and the host can issue another set of 16 REFpb commands in the same or a different order. However, it is illegal to send another REFpb command to a bank unless all 16 banks have been refreshed using the REFpb command. The host must track the banks being refreshed by the REFpb command.

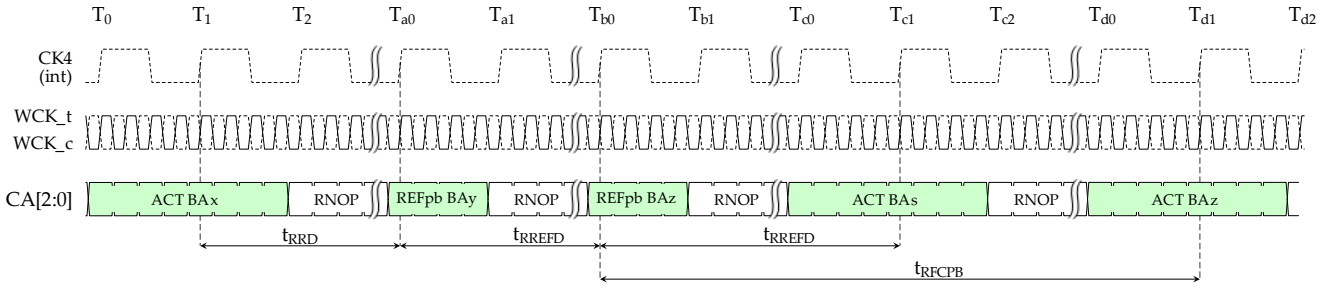
The bank count is synchronized between the host and the device by resetting the bank count to zero. Synchronization occurs upon exit from reset state or by issuing a REFab or Self Refresh Entry command. Both commands may be issued at any time even if a preceding sequence of REFpb commands has not completed cycling through all 16 banks. The internal refresh counter is not incremented in case of such incomplete cycling. It is pointed out that multiple occurrences of synchronization events without refresh counter increment may result in an insufficient refresh of the memory array; it is suggested to issue additional REFab commands in that case.

6.12.2 Refresh Per-Bank (REFpb) Command (cont'd)

The average rate of REFpb commands is given by tREFIpb. The flexibility to postpone refresh commands also extends to REFpb commands. The maximum interval between refreshes to a particular bank is limited to 9 × tREFI.

At any given time, a maximum of 9 × 16 REFpb commands can be issued within tREFI. At any given time, a maximum of 9 refreshes to a particular bank can be issued within tREFI.

CA parity is evaluated with the REFpb command when the parity calculation is enabled in MR15 OP0.



NOTE 1 BA = bank address.

Figure 106 — REFpb Timings

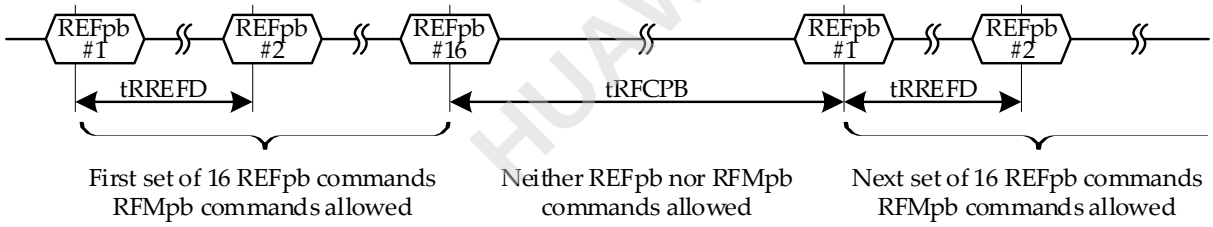


Figure 107 — Sets of REFpb Commands

The example in *TABLE 88* shows two full sets of REFpb commands with the bank counter reset to zero and the refresh counter incremented after 16 REFpb commands each. The third set to REFpb commands is interrupted by the REFab command which resets the bank counter to 0 and performs refreshes to all banks indicated by the refresh counter.

6.12.2 Refresh Per-Bank (REFpb) Command (cont'd)

Table 88 — Refresh Counter Increments with REFpb

Count	Sub-Count	Command	BA[3:0]	Refresh Bank	Bank Counter	Refresh Counter
0	0	Reset, REFab or SRE command			To 0	
1	1	REFpb	0000	0	0 to 1	n
2	2	REFpb	0001	1	1 to 2	
3	3	REFpb	0010	2	2 to 3	
4	4	REFpb	0011	3	3 to 4	
. . .						
15	15	REFpb	1110	14	14 to 15	
16	16	REFpb	1111	15	15 to 0	
17	1	REFpb	0100	4	0 to 1	n + 1
18	2	REFpb	0111	7	1 to 2	
19	3	REFpb	1011	11	2 to 3	
20	4	REFpb	0110	6	3 to 4	
. . .						
31	15	REFpb	1100	12	14 to 15	
32	16	REFpb	0001	1	15 to 0	
33	1	REFpb	0010	2	0 to 1	n + 2
34	2	REFpb	1001	9	1 to 2	
35	3	REFpb	0000	0	2 to 3	
36	0	REFab	V	all	To 0	n + 2
37	1	REFpb	1010	10	0 to 1	n + 3
38	2	REFpb	0101	5	1 to 2	
. . .						

6.12.2 Refresh Per-Bank (REFpb) Command (cont'd)

Table 89 — REFab and REFpb Command Scheduling Requirements

From Command	To Command	Minimum Delay between "From Command" to "To Command"	Notes
REFab	REFab, Self Refresh Entry or Self Refresh Sleep Entry	t_{RFCab}	
	REFpb (any bank)	t_{RFCab}	
	Activate (any bank)	t_{RFCab}	
REFpb	REFab, Self Refresh Entry or Self Refresh Sleep Entry	t_{RFCpb}	
	REFpb (different bank)	t_{RREFD} or t_{RFCpb}	1
	REFpb (same bank)	t_{RFCpb}	2
	Activate (different bank)	t_{RREFD}	
	Activate (same bank)	t_{RFCpb}	
Activate	REFab, Self Refresh Entry or Self Refresh Sleep Entry	t_{RC}	3
	REFpb (different bank)	t_{RRD}	
	REFpb (same bank)	t_{RC}	3

NOTE 1 Use tRFCpb when the first REFpb command completes a set of 16 per-bank refresh operations and the second REFpb command initiates the next set of refresh operations. Use tRREFD in all other cases.

NOTE 2 Back-to-back REFpb commands to the same bank are only allowed when the first REFpb command completes a set of 16 per-bank refresh operations and the second REFpb command initiates the next set of 16 per-bank refresh operations.

NOTE 3 A bank must be in the idle state with tRP satisfied before it is refreshed.

6.13 Refresh Management (RFM)

Periods of high DRAM activity may require additional refresh commands to protect the integrity of the stored data. The requirement for additional Refresh Management (RFM) is optional for GDDR7 SGRAMs, and devices will indicate the requirement in read-only Info Read register 4, bit 6 (see *TABLE 90*).

Table 90 — RFM Support Definition in Info Read Register 4

IRA	Bit	Field	Description
4	DQ6	RFM	Refresh Management (RFM) 0_B –Refresh Management usage is not required 1_B –Refresh Management usage is required

A suggested implementation of Refresh Management by the controller monitors (counts) Activate commands issued per bank to the device. This activity can be monitored as a rolling accumulated Activate (RAA) count. Each Activate command will increment the RAA count by 1 for the individual bank receiving the Activate command.

When the RAA count reaches a DRAM vendor specified Initial Management Threshold (RAAIMT), which is indicated in Info Read register 4, bits [2:0] (see *TABLE 91*), additional refresh management is needed. Executing the Refresh Management (RFM) command allows additional time for the GDDR7 SGRAM to manage refresh internally. The RFM operation can be initiated to all banks with the Refresh Management All-Bank (RFMab) command, or to a single bank with the Refresh Management Per-Bank (RFMpb) command as shown in *FIGURE 103*.

Table 91 — RAAIMT Definition in Info Read Register 4

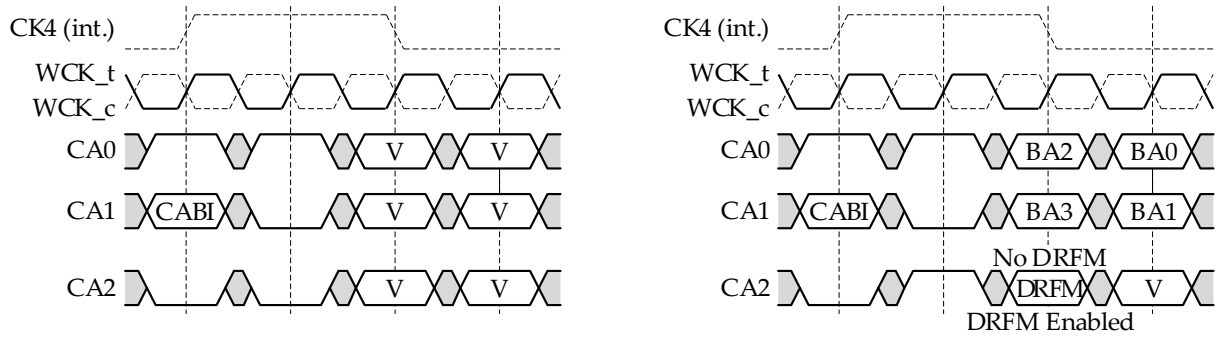
IRA	Bit	Field	Description
4	DQ[2:0]	RAAIMT	Default RAA Initial Management Threshold (RAAIMT) The field shall be ignored when the RFM required bit is 0. 000_B –16 001_B –20 010_B –24 011_B –28 100_B –32 101_B –36 110_B –40 111_B – Reserved

A GDDR7 SGRAM not requiring refresh management will ignore RFMab and RFMpb commands and execute an RNOP command instead.

The Refresh Management command scheduling shall meet similar minimum separation requirements as those for the Refresh commands (see *REFRESH* section); this includes that RFMpb commands are not allowed in the tRFCpb pause between two sets of 16 REFpb commands. The RFMab command period is tRFMab, and the RFMpb command period is tRFMpb. The requirement for REFpb commands to be issued to all banks in a rolling fashion does not apply to RFMpb commands.

CA parity is evaluated with the RFMab and RFMpb commands when the CA parity calculation is enabled in MR15 OP0.

6.13 Refresh Management (RFM) (cont'd)



Refresh Management All-Bank (RFMab) **Refresh Management Per-Bank (RFMpb)**

NOTE 1 BA = Bank Address; CABI = Command/Address Bus Inversion; DRFM = Directed Refresh Management; V = Valid (H or L but not floating).

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

NOTE 3 The DRFM bit is only evaluated when the DRFM feature is enabled in MR8 OP2, otherwise the bit is Don't Care. Refer to the *DIRECTED REFRESH MANAGEMENT (DRFM)* section for more details.

NOTE 4 An RFMpb command with the DRFM bit set 0_B is referred to as a DRFMpb command.

Figure 108 — Refresh Management All-Bank (RFMab) and Refresh Management Per-Bank (RFMpb) Commands

When an RFM command is issued, the RAA count in any bank receiving the command can be decremented by the RAAIMT value, down to a minimum RAA value of 0 (no negative or "pull-in" of RFM commands is allowed). Issuing an RFMab command allows the RAA count in all banks to be decremented by the RAAIMT value. Issuing an RFMpb command allows the RAA count only in the bank selected by BA[3:0] to be decremented by the RAAIMT value.

RFM commands are allowed to accumulate or "postpone", but the RAA count shall never exceed a vendor specified RAA Maximum Management Threshold (RAAMMT), which is indicated in Info Read register 4, bits [4:3] (see *TABLE 92*). If the RAA count reaches RAAMMT, the host shall not issue additional ACT commands to that bank until one or more REF or RFM commands have been issued to reduce the RAA count below the maximum value.

Table 92 — RAAMMT Definition in Info Read Register 4

IRA	Bit	Field	Description
4	DQ[4:3]	RAAMMT	Default RAA Maximum Management Threshold (RAAMMT) The field shall be ignored when the RFM bit is 0_B. 00_B –6 × RAAIMT 01_B – 8 × RAAIMT 10_B – 10 × RAAIMT 11_B –12 × RAAIMT

6.13 Refresh Management (RFM) (cont'd)

An RFM command does not replace the requirement for the controller to issue periodic REF commands, nor does an RFM command affect internal refresh counters. The RFM commands are bonus time for the GDDR7 SGRAM to manage refresh internally. However, issuing a REF command also allows decrementing the RAA count by a value indicated by the RAADEC field Info Read register 4, bit 5 (see *TABLE 93*). Hence, any periodic REF command issued to the GDDR7 SGRAM allows the RAA count of the banks being refreshed to be decremented by that value. Issuing an REFab command allows the RAA count in all banks to be decremented by that value. Issuing an REFpb command allows the RAA count only in the bank selected by BA[3:0] to be decremented by that value.

Table 93 — RAADEC Definition in Info Read Register 4

IRA	Bit	Field	Description
4	DQ5	RAADEC	Default RAA count decrement per REF command. The field shall be ignored when the RFM bit is 0. 0_B – 1.0 × RAAIMT 1_B – 0.5 × RAAIMT

The per-bank RAA count values may be reset to 0 when the channel is held in Self Refresh mode (including Self Refresh Sleep and Hibernate Self Refresh Sleep modes) for at least tRAASRF time. No decrement to the per-bank RAA count is allowed for entering or exiting Self Refresh and when a channel is held in Self Refresh for less than tRAASRF time.

6.13.1 Adaptive Refresh Management (ARFM)

GDDR7 SGRAMs optionally support a Refresh Management mode called Adaptive Refresh Management (ARFM), and devices will indicate the support of ARFM in Info Read register 4, bit 7 (see *TABLE 94*). Since RFM related parameters RAAIMT, RAAMMT and RAADEC are read-only, the ARFM mode allows the controller flexibility to choose additional (lower) RFM threshold settings called "RFM Levels". The RFM levels permit alignment of the controller issued RFM commands with the SGRAM internal management of these commands. ARFM related mode register bits in MR8 OP[1:0] select the RFM Level as shown in *TABLE 95*.

Table 94 — ARFM Support Definition in Info Read Register 4

IRA	Bit	Field	Description
4	DQ7	ARFM	Adaptive Refresh Management (ARFM) 0_B – Adaptive Refresh Management is not supported 1_B – Adaptive Refresh Management is supported

Table 95 — Mode Register Definition for Adaptive RFM Levels

MR8 OP[1:0]	RFM Level	RFM Requirement	RAAIMT	RAAMMT	RAA Decrement per REF Command	Notes
00_B	Default	Default	RAAIMT	RAAMMT	RAADEC	
01_B	Level A	RFM is required	RAAIMT_A	RAAMMT_A	RAADEC_A	
10_B	Level B	RFM is required	RAAIMT_B	RAAMMT_B	RAADEC_B	
11_B	Level C	RFM is required	RAAIMT_C	RAAMMT_C	RAADEC_C	

The Adaptive RFM mode inherits the RAA counting and decrement attributes of the standard RFM mode, while using the alternate RAAIMT, RAAMMT and RAADEC values for the selected RFM level. Increasing the RFM level results in increased need for RFM commands. Level C is highest RFM level. The alternate RAAIMT, RAAMMT and RAADEC values for RFM levels A to C can be retrieved from the corresponding fields in Info Read registers 5, 6 and 7, respectively (see *TABLE 97*).

Setting the bits in MR8 OP[1:0] to something other than the default 00_B value will select one of the RFM levels A, B or C. The host shall decrement the Rolling Accumulated ACT (RAA) count to 0, either with RFM or pending REF commands, prior to making a change to the ARFM level. It is required to set the same RFM level on all channels of the GDDR7 SGRAM.

Adaptive RFM also allows a GDDR7 SGRAM shipped with 'RFM not required' (RFM bit = 0_B) to override that initial setting and enable RFM by programming a non-default ARFM level. The device internally manages the change to treat the RFM command as an RFM command in this special override case as shown in *TABLE 96*.

6.13.1 Adaptive Refresh Management (ARFM) (cont'd)

Table 96 — RFM Commands Perceived by GDDR7 SGRAM

Command Issued	Bits in Info Mode		RFM Level MR8 OP[1:0]	Command Perceived	Notes
	RFM	ARFM			
RFMab or RFMpb	0_B (RFM not required)	0_B (ARFM not supported)	00_B	RNOP	
			01_B, 10_B or 11_B	Illegal	1
		1_B (ARFM supported)	00_B	RNOP	
			01_B, 10_B or 11_B	RFMab / RFMpb	2
	1_B (RFM required)	0_B (ARFM not supported)	00_B	RFMab / RFMpb	
			01_B, 10_B or 11_B	Illegal	1
		1_B (ARFM supported)	00_B, 01_B, 10_B or 11_B	RFMab / RFMpb	

NOTE 1 These cases are marked as 'Illegal' because GDDR7 SGRAMs not supporting Adaptive RFM do not support the selection of an ARFM level via MR8 OP[1:0] and therefore define these bits as RFU which implies that the only supported setting for these bits is 00_B.

NOTE 2 Adaptive RFM enables an GDDR7 SGRAM shipped with RFM = 0_B (RFM not required) to override the initial setting and enable Adaptive RFM by programming a non-default RFM level.

Table 97 — ARFM Level Definition in Info Read Registers 5 to 7 [1]

IRA	Bit	Field	Description
5	DQ[7:6]	RFU	00_B
	DQ5	RAADEC_A	RAA Counter Decrement per REF Command for RFM level A (MR8 OP[1:0] = 01_B). Same encoding as in RAADEC
	DQ[4:3]	RAAMMT_A	RAA Maximum Management Threshold (RAAMMT) for RFM level A (MR8 OP[1:0] = 01_B). Same encoding as in RAAMMT
	DQ[2:0]	RAAIMT_A	RAA Initial Management Threshold (RAAIMT) for RFM level A (MR8 OP[1:0] = 01_B). Same encoding as in RAAIMT
6	DQ[7:6]	RFU	00_B
	DQ5	RAADEC_B	RAA Counter Decrement per REF Command for RFM level B (MR8 OP[1:0] = 10_B). Same encoding as in RAADEC
	DQ[4:3]	RAAMMT_B	RAA Maximum Management Threshold (RAAMMT) for RFM level B (MR8 OP[1:0] = 10_B). Same encoding as in RAAMMT
	DQ[2:0]	RAAIMT_B	RAA Initial Management Threshold (RAAIMT) for RFM level B (MR8 OP[1:0] = 10_B). Same encoding as in RAAIMT
7	DQ[7:6]	RFU	00_B
	DQ5	RAADEC_C	RAA Counter Decrement per REF Command for RFM level C (MR8 OP[1:0] = 11_B). Same encoding as in RAADEC
	DQ[4:3]	RAAMMT_C	RAA Maximum Management Threshold (RAAMMT) for RFM level C (MR8 OP[1:0] = 11_B). Same encoding as in RAAMMT
	DQ[2:0]	RAAIMT_C	RAA Initial Management Threshold (RAAIMT) for RFM level C (MR8 OP[1:0] = 11_B). Same encoding as in RAAIMT

NOTE 1 These registers shall be ignored when ARFM is not supported by the device (ARFM bit = 0_B).

6.13.2 Directed Refresh Management (DRFM)

Directed Refresh Management (DRFM) is a feature that gives the controller additional flexibility for maintaining data integrity within the GDDR7 SGRAM. The DRFM feature allows the SGRAM to capture a host-requested row address, which then is followed by a host-directed RFMpb command allowing the SGRAM to refresh physically adjacent neighboring rows of the requested row address. DRFM is disabled by default and can be enabled by setting the DRFM bit in MR8 OP2 to 1_B.

When DRFM is enabled, executing a PREpb command to an open row with the DRFM bit set to 0_B will instruct the SGRAM to sample the currently activated row address for DRFM while initiating the row precharge as normal. Note that a PREpb command is also legal when issued to a closed bank, however, in that case no row address will be captured.

Each bank has an independent DRFM address register for the DRFM row address sample. This DRFM address register is updated with each DRFM address sample to the bank, resulting in the last (most recent) address sample being retained for the host directed DRFMpb command.

After the DRFM address sample, the host can then issue an RFMpb command with the DRFM bit set to 0_B (referred to as DRFMpb command, see *FIGURE 103*) to service the sampled DRFM address. This DRFMpb command is supplemental to the device's RFM requirements and does not allow RAA count to be decremented. A DRFMpb command issued to a bank without a valid address sample will be executed as a regular RFMpb command.

Following a DRFMpb command to the GDDR7 SGRAM, any host-requested row address sample to the bank that received the DRFMpb command will be cleared from further use.

Aside from the DRFMpb command, a chip reset is the only other way to clear DRFM sampled addresses. DRFM sampled addresses will be retained during Self Refresh modes, requiring the host to resample prior to issuing a DRFMpb command if the address retained in a bank is no longer relevant. Additionally, no RAA credit is given to banks with DRFM sampled addresses, regardless of relevancy.

The DRFMpb command scheduling shall meet the same minimum separation requirements like tRP, tRRD or tRREFD as for a RFMpb command (see *REFRESH MANAGEMENT (RFM)* section). DRFMpb commands and row commands such as Activate, Refresh (REFpb or REFab) or Refresh Management (RFMpb, RFMab) to other banks are not supported while a DRFMpb command is in progress and the t_{DRFM} timing has not been met. Precharge commands as well as Reads and Writes to different banks with or without auto precharge however are supported.

On average, no more than 1,024 DRFMpb commands are allowed to be issued to the same bank/row address combination within tREF.

6.13.2.1 Bounded Refresh Configuration

The DRFMpb command refreshes physically adjacent neighboring rows to the DRFM sampled address, up to the distance specified by the Bounded Refresh Configuration (BRC) as defined by MR8 OP[4:3]. The SGRAM is responsible for applying a refresh ratio to the outermost rows being refreshed to protect the SGRAM from excessive refreshes on rows adjacent to the outermost rows.

For example, BRC2 will always refresh the +1 physically adjacent neighboring rows, and the ±2 physically adjacent neighboring rows may be refreshed at a reduced rate as determine by the SGRAM. Likewise, if BRC4 is programmed, the SGRAM will always refresh the ±1, +2 and ± 3 physically adjacent neighboring rows, while applying a ratio to ±4 physically adjacent neighboring rows. The support of BRC3 and BRC4 is optional and indicated in Vendor ID4 (IRA 3) bit DQ0.

The DRFM cycle time per row is tRRF = 60ns. The corresponding DRFMpb command duration tDRFM is determined by the selected BRC option and given as tDRFM = 2 × tRRF × BRC as summarized in *TABLE 98*.

Table 98 — Bounded Refresh Configuration and tDRFM Timings

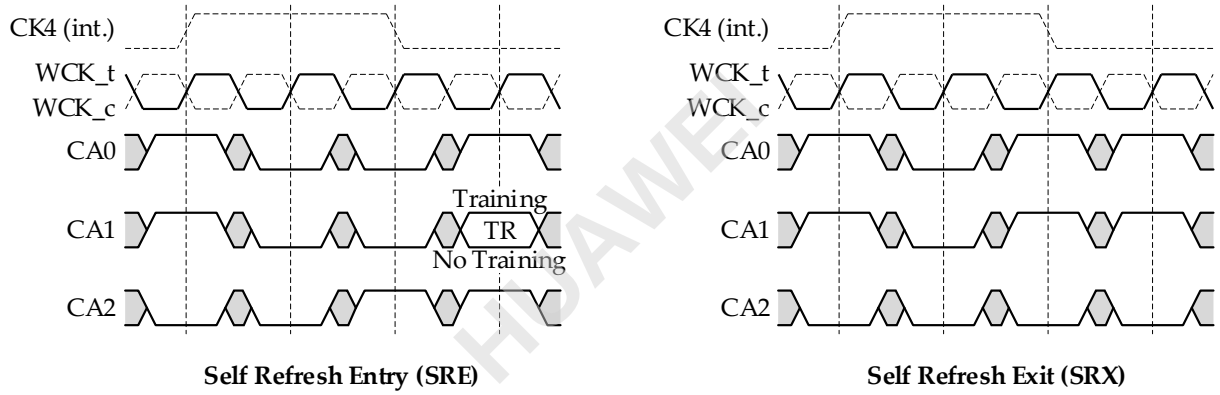
BRC	MR8 OP[4:3]	Rows Refreshed	tDRFM
2	00	Always ±1, Ratio ±2	4 x tRRF
3 (optional)	01	Always ±1, ±2, Ratio ±3	6 x tRRF
4 (optional)	10	Always ±1, ±2, ±3, Ratio ±4	8 x tRRF
RFU	11	RFU	RFU

6.14 Self Refresh

In Self Refresh mode the GDDR7 SGRAM autonomously retains the data in the array without the need for the host to periodically issue Refresh commands. GDDR7 Self Refresh mode is not a low power mode like it was in GDDR6. The equivalent low power modes in GDDR7 are Self Refresh Sleep mode and Hibernate Self Refresh Sleep mode, which combine low power states with autonomous internal refresh provided by Self Refresh. See the *SLEEP MODES* section for more details on these other modes.

Self Refresh mode shall be entered whenever changes of the device configuration via MRS commands, CA bus training or data training are expected to take longer than allowed by the refresh requirements of the device. Entering Self Refresh prior to performing these operations will ensure that data in the array will autonomously be refreshed properly.

Self Refresh is entered with the Self Refresh Entry (SRE) command as shown in *FIGURE 109*. The command is only allowed when all banks are precharged with tRP satisfied; it must be paired with a CNOP1 command on the column command inputs CA[4:3]; RNOP2 and CNOP2 commands must be issued during the tCPDED period following the SRE command. Upon entering Self Refresh the bank counter for REFpb will be reset by the DRAM.



NOTE 1 TR = Data Training.

NOTE 2 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 109 — Self Refresh Entry and Exit Commands

The CA interface and WCK receiver remain active upon entering Self Refresh mode, and a stable WCK clock must be maintained. The state of RCK is not affected by entering Self Refresh mode, and is allowed to change as described in the *READ CLOCK (RCK)* section.

TABLE 99 summarizes the commands allowed during self refresh mode; some of them are only allowed when the TR bit set to H with the SRE command.

The Self Refresh related IDD6 specification applies when the TR bit is set to L with the SRE command ("No Training") and only NOP commands are issued as shown in *FIGURE 110;* higher power consumption shall be expected when any of the above operations is initiated while Self Refresh mode is active.

The channel initiates a minimum of one internal refresh once it enters Self Refresh mode.

6.14 Self Refresh (cont'd)

Table 99 — Commands Supported during Self Refresh

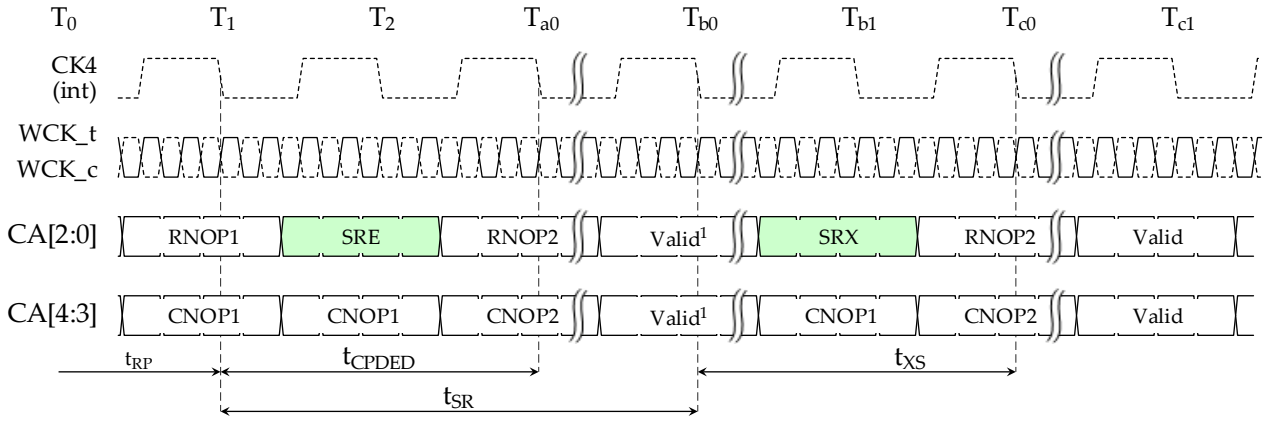
TR = L	TR = H	Notes
–	Data training (LDFF, WRTR, RDTR, RDWTEC)	1
–	Mode Register Set (MRS)	
–	Info Read (IRD)	2
CA Training Entry (CATE)		3
Sleep Mode Entry (SLE)		4
Self Refresh Exit (SRX)		

NOTE 1 Refer to the *DATA TRAINING* section for details.
NOTE 2 Refer to the *INFO READ* section for details.
NOTE 3 Refer to the *COMMAND ADDRESS BUS TRAINING* section for details. Exit from CA training requires a CSP command before the channel returns to regular self refresh state and is ready to accept subsequent access commands.
NOTE 4 Refer to the *SLEEP MODES* section for details. Exit from sleep mode requires a CSP command before the channel returns to regular self refresh state and is ready to accept subsequent access commands.

The channel remains in Self Refresh mode even when a configuration change, CA bus training or data training have completed. The channel must be held in Self Refresh for at least tSR time.

Self Refresh mode is synchronously exited with the Self Refresh Exit (SRX) command as shown in *FIGURE 109*. A delay of at least tXS must be satisfied before a valid access command can be issued to the channel to allow for completion of any internal refresh in progress. RNOP2 and CNOP2 commands must be issued during that period. The channel can be put back into Self Refresh mode after issuing one extra REFab command.

CA parity is evaluated with the SRE and SRX commands as well as all commands issued while in Self Refresh mode when the parity calculation is enabled in MR15 OP0.



NOTE 1 Valid commands comprise MRS, IRD, RCKSTRT, RCKSTOP, LDFF, WRTR, RDTR or RDWTEC for data training when TR = H. CATE for entering CA Bus Training and SLE for entering Self Refresh Sleep mode can be issued regardless of the TR bit. See the *AC TIMINGS* section for timings between SRE and valid data training commands as they require additional timings to be met beyond tCPDED.

NOTE 2 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 110 — Self Refresh Entry and Exit

6.14 Self Refresh (cont'd)

Table 100 — Signal States During Self Refresh

Signal	State [1]		Signal	State
WCK_t, WCK_c	Active		RCK_t, RCK_c	Active or High-Z (2, 3)
CA	Active			
DQ, DQE (2)	TR = L	High-Z	ERR	Active
	TR = H	Active		

NOTE 1 See the *SLEEP MODES* section for pin states during Self Refresh Sleep and Hibernate Self Refresh Sleep modes.
NOTE 2 See *DATA TRAINING* for the pin states during Data Training.
NOTE 3 The state of RCK_t and RCK_c outputs also depend on the selected RCK type and mode.

6.15 Sleep Modes

GDDR7 SGRAMs supports different variations of sleep modes to help reducing the device's power consumption during periods of no activity:

- Sleep mode: the channel is in a low power state without internal autonomous refresh; this mode is primarily intended for frequency changes with a duration shorter than the refresh requirements of the device, or cases where retaining data in the array is not required.

- Self Refresh Sleep mode: the channel is in the same low power state as above, but with internal autonomous refresh activated. This mode is equivalent to the Self Refresh mode of GDDR6.

- Hibernate Self Refresh Sleep mode: this mode is intended for longer periods of no activity; it sets the channel in a state with even lower power consumption than with Self Refresh Sleep mode, at the expense of a significantly extended period to return to normal operation.

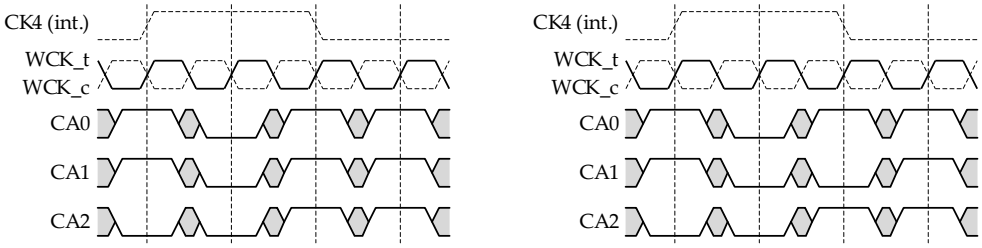
Sleep mode can be entered from bank idle state with the Sleep Entry (SLE) command as shown in *FIGURE 111*. Self Refresh Sleep mode can be entered from Self Refresh with the Sleep Entry (SLE) command, or directly from bank idle state with the Self Refresh Sleep Entry (SRSE) command as shown in *FIGURE 111*. The commands must be paired with a CNOP1 command on the column command inputs CA[4:3]. Hibernate Self Refresh Sleep mode is entered instead of Self Refresh Sleep mode when the Hibernate bit in MR8 OP11 has been set prior to the SLE or SRSE command. Upon entering Self Refresh Sleep or Hibernate Self Refresh Sleep the bank counter for REFpb will be reset by the DRAM.

Both the Sleep Entry (SLE) command and the Self Refresh Sleep Entry (SRSE) command can only be issued when all banks are precharged with tRP satisfied, and when all previous Read or Write operations have completed. For Reads, a burst completion is defined as when the last data element including CRC has been transmitted on the data outputs (tRDSLE timing); for Writes, a burst completion is defined as when the last data element has been written to the memory array and the result of the write CRC check has been returned to the host (tWRSLE timing). Other operations, for example a MRS from bank idle or Self Refresh states, must meet the related timings before SRSE (bank idle) or SLE (Self Refresh) commands. See the *SIMPLIFIED STATE DIAGRAM* for relevant commands and states legal from bank idle and Self Refresh.

If RCK is configured to one of the Start/Stop modes and is running, it is required to stop RCK via the RCKSTOP command prior to Sleep mode entry (refer to the *READ CLOCK (RCK)* section for details on RCK). If RCK is in the always running mode, there is no need to stop the RCK prior to entering Sleep. In this case the RCK will resume toggling during tRCK_AON_SLX.

RNOP2 and CNOP2 commands along with valid WCK clocks are required for a period of tCPDED cycles following the SLE or the SRSE command.

6.15 Sleep Modes (cont'd)



NOTE 1 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

NOTE 2 The channel enters Hibernate Self Refresh Sleep mode when the Hibernate bit in MR8 OP11 has been set prior to the SLE or SRSE command.

Figure 111 — Sleep Mode Entry Commands

CA0 must be held High to keep the channel in any of the Sleep modes (Sleep, Self Refresh Sleep and Hibernate Self Refresh Sleep) while the channel's other external signals are "Don't care". CA[4:1] and WCK inputs are in ODT state. DQ and DQE signals as well as RCK_t, RCK_c and ERR outputs are in High-Z state. All power supplies (VDD, VDDQ, VPP) must be maintained at valid levels. Refer to *TABLE 101*.

The WCK clock is internally disabled during Sleep modes to save power. The host may change the WCK clock frequency or halt the WCK clock after the tCPDED timing has been met. Sleep Mode is the only state (except reset state) in which a clock frequency change is allowed.

The state that the channel shall enter upon Sleep mode exit must be selected prior to entering Sleep mode:

- Exit from Sleep mode: if the SLX2CAT mode register bit in MR0 OP5 is set to 0_B, the channel will enter CA Bus Training after the tSLX timing has been met; if the bit is set to 1_B, the channel will await the CSP command.
- Exit from Self Refresh Sleep mode: if the SRSLX2CAT mode register bit in MR0 OP6 is set to 0_B, the channel will enter CA Bus Training Self Refresh after the tSLX timing has been met; if the bit is set to 1_B, the channel will await the CSP command. Note that Self Refresh remains active, and a Self Refresh exit (SRX) command will be required before the channel can resume normal operation.

The minimum time that the channel must remain in Sleep mode is tSLEEP and tSRFSLP for Self Refresh Sleep mode.

The procedure for exiting Sleep mode requires a sequence of events.

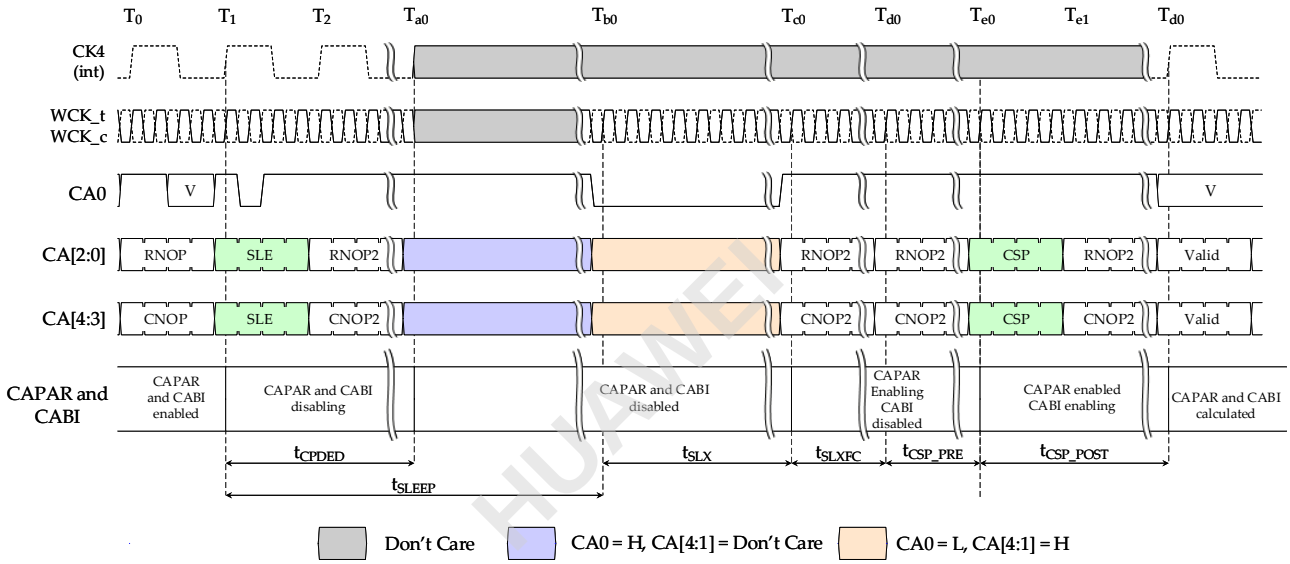
- Sleep mode exit is asynchronously initiated when CA0 is pulled Low for at least tSLX time; the other CA inputs must be driven High during this time;
- The WCK clock must be stable when CA0 is pulled Low, and the WCK receiver will subsequently be enabled;
- After tSLX time has elapsed, if SLX2CAT (MR0 OP5) or SRSLX2CAT (MR0 OP6) are set to 1_B, RNOP2/CNOP2 commands must be issued for at least tCSP_PRE period, followed by one CSP command. Whereas if SLX2CAT or SRSLX2CAT (MR0 OP6) are set to 0_B, RNOP2/CNOP2 commands must be issued for at least tSLX_CAT followed by entry to CA Training.
- If the WCK frequency in MR12 OP[8:4] has changed, an extra time tSLXFC as shown in *FIGURE 112* and *FIGURE 113*.
- When in Self Refresh Sleep with Auto ECS operation enabled (MR22 OP11=1_B), the exit procedure requires tSLX_ECS must be met before issuing the CSP command.
- After tCSP_POST has elapsed, the device will return to bank idle state if exiting from Sleep mode, and to Self Refresh with TR = H state if exiting from Self Refresh Sleep mode.

6.15 Sleep Modes (cont'd)

During Sleep modes the on-die termination (ODT) and driver will not be auto-calibrated. Recalibration will be automatically resumed within tSLX after Sleep mode exit or Self Refresh Sleep mode exit. No recalibration is performed when calibration update has been set to off in MR5 OP11.

CA parity is evaluated with the Sleep Entry or Self Refresh Sleep Entry commands when the parity calculation is enabled in MR15 OP0. CA parity calculation is suspended during Sleep modes and will be resumed during the tCSP_PRE period prior to when the CSP command is allowed.

When enabled in MR0 OP0, CA bus inversion (CABI) is suspended during Sleep modes and will be resumed within the tCSP_POST period following the CSP command. See the *COMMAND ADDRESS BUS INVERSION (CABI)* section for details on CABI.



NOTE 1 CA0 is part of the row command bus CA[2:0] but broken out and shown separately to illustrate how CA0 assumes the function of a CKE input during Sleep Modes.

NOTE 2 The states of CAPAR and CABI calculations are applicable when the functions are enabled in the respective mode registers.

NOTE 3 The tSLXFC time is required to be observed only if the WCK frequency in MR12 OP[8:4] has changed.

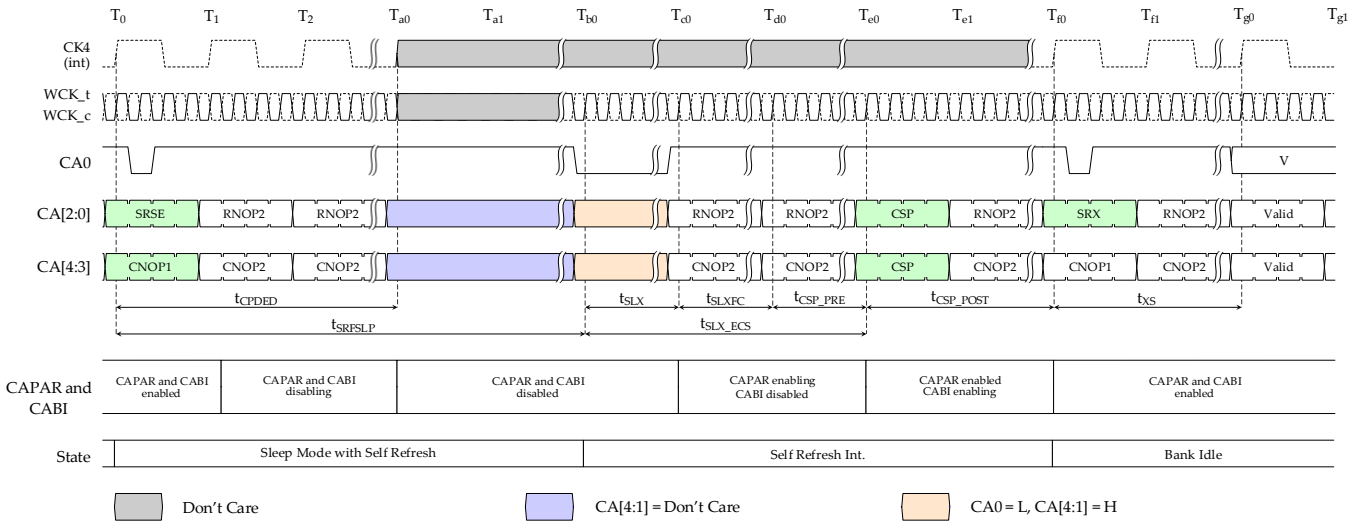
NOTE 4 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

NOTE 5 If RCK always on mode is enabled, RCK will restart asynchronously during tRCK_AON_SLX. See *AC TIMINGS* for details.

NOTE 6 The driving of CA0 low to signal Sleep exit is asynchronous. tSLEEP is counted from the SLE command to the first WCK_t rising edge after CA0 is sampled low as WCK must be stable when CA0 is pulled low. tSLX is referenced from the same first WCK_t rising edge.

Figure 112 — Sleep Mode Entry and Exit

6.15 Sleep Modes (cont'd)



NOTE 1 CA0 is part of the row command bus CA[2:0] but broken out and shown separately to illustrate how CA0 assumes the function of a CKE input during Sleep Modes.

NOTE 2 The states of CAPAR and CABI calculations are applicable when the functions are enabled in the respective mode registers.

NOTE 3 The tSLXFC time is required to be observed only if the WCK frequency in MR12 OP[8:4] has changed.

NOTE 4 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

NOTE 5 If RCK always on mode is enabled, RCK will restart asynchronously during tRCK_AON_SLX. See *AC TIMINGS* for details.

NOTE 6 The driving of CA0 low to signal Sleep exit is asynchronous. tSRFSLP is counted from the SRSE command to the first WCK_t rising edge after CA0 is sampled low as WCK must be stable when CA0 is pulled low. tSLX is referenced from the same first WCK_t rising edge.

NOTE 7 tSLX_ECS is the time from the SLX to the CSP command when Auto ECS is enabled (MR22 OP11=1_B). tSLX_ECS is shown = tSLX+tSLXFC+tCSP_PRE for illustration purposes. See *AC TIMINGS* for details.

NOTE 8 When tSLX_ECS has expired, the ECS logs will not be guaranteed to be updated. The ECS logs will be guaranteed to be updated after tSRSX_ECSLOG_UPD has expired.

Figure 113 — Self Refresh Sleep Mode Entry and Exit

Table 101 — Signal States During Sleep Modes

Signal	State	Signal	State
WCK_t, WCK_c	ODT state (1)	DQ, DQE	High-Z
CA0	Active	RCK_t, RCK_c	High-Z
CA[4:1]	ODT state (1)	ERR	High-Z
NOTE 1 The signals terminate with the programmed ODT setting, which could be terminating or High-Z.			

6.15 Sleep Modes (cont'd)

6.15.1 Hibernate Self Refresh Sleep Mode

Hibernate Self Refresh Sleep mode is a special mode that provides the same data retention as in the regular Self Refresh Sleep mode but allows the channel to disable additional circuits to achieve an even lower power consumption at the expense of a significantly extended period to return to normal operation. The full power savings can only be achieved when all 2 or 4 channels of the device (depending on the selected device configuration) are in Hibernate Self Refresh Sleep mode.

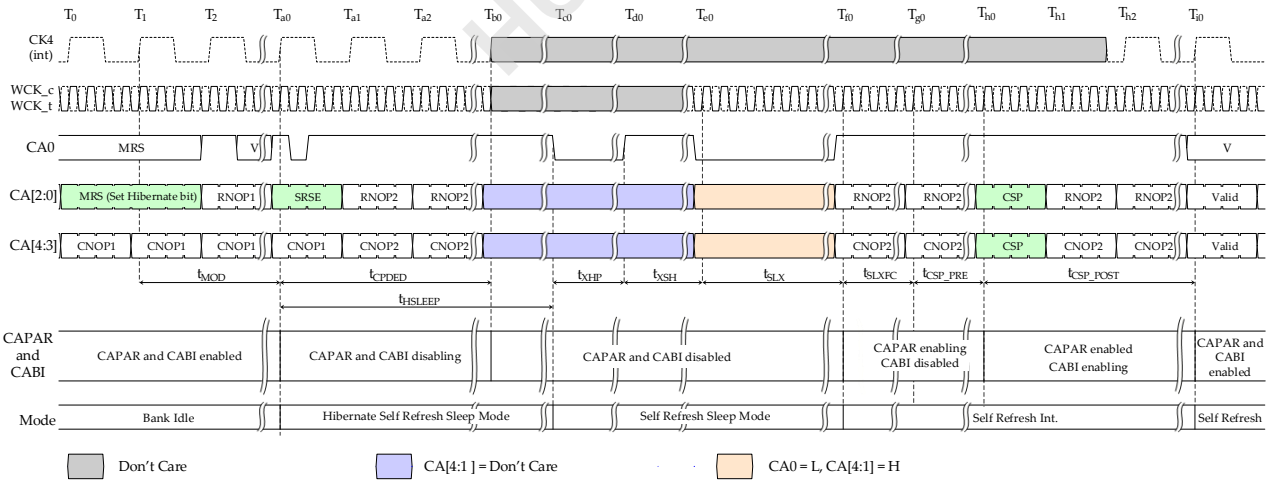
The Hibernate bit in MR8 OP11 is associated with Hibernate Self Refresh Sleep. The bit is self-clearing, meaning that an MRS command must set this bit any time the device shall enter Hibernate Self Refresh Sleep using the SLE or the SRSE command. See above for other conditions associated with these commands.

CA0 must be held High to keep the channel in Hibernate Self Refresh Sleep mode while the channel's other external signals are "Don't care". CA[4:1] and WCK inputs are in ODT state. DQ and DQE signals as well as RCK_t, RCK_c and ERR outputs are in High-Z state. All power supplies (VDD, VDDQ, VPP) must be maintained at valid levels. Refer to *TABLE 101*.

The minimum time that the channel must remain in Hibernate Self Refresh Sleep mode is tHSLEEP.

Exiting Hibernate Self Refresh Sleep requires a sequence of events as shown in *FIGURE 114*.

- At first, CA0 must be pulled to Low for at least tXHP period and then be pulled High again.
- The channel is now in Self Refresh Sleep mode, and must be held in this mode for at least tXSH period, to retain data during the extended exit time from Hibernate Self Refresh Sleep mode.
- After tXSH period the sequence for Self Refresh Sleep mode exit shall be followed to return to normal operation.



NOTE 1 The channel enters Hibernate Self Refresh Sleep mode when the channel is in bank idle state and an SRSE command is issued as shown with the Hibernate bit set before. The channel also enters Hibernate Self Refresh Sleep mode when the channel is in Self Refresh mode and an SLE command is issued instead of the SRFSLE command.

NOTE 2 CA0 is part of the row command bus CA[2:0] but broken out and shown separately to illustrate how CA0 assumes the function of a CKE input during Hibernate Self Refresh Sleep Mode.

NOTE 3 The states of CAPAR and CABI calculations are applicable when the functions are enabled in the respective mode registers.

NOTE 4 The tSLXFC time is required to be observed only if the WCK frequency in MR12 OP[8:4] has changed.

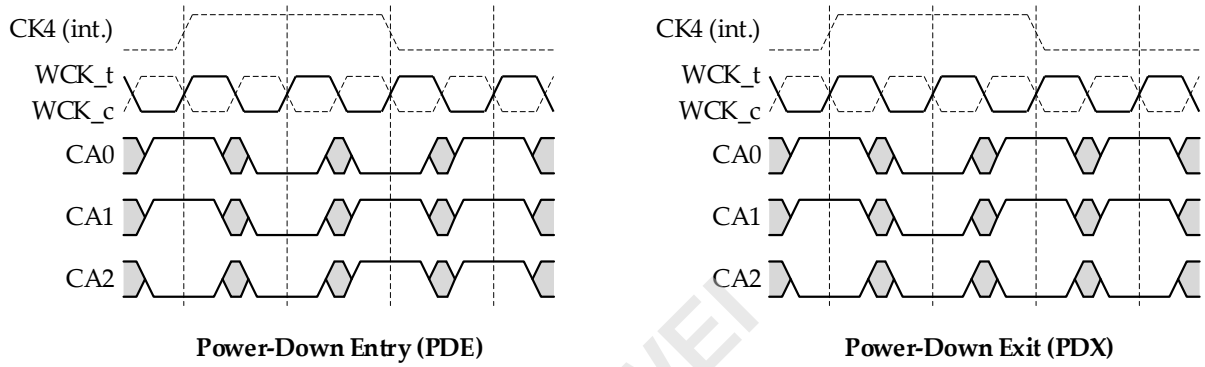
NOTE 5 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 114 — Hibernate Self Refresh Sleep Mode Entry and Exit

6.16 Power-Down

Entering Power-Down mode can reduce the operating power of the GDDR7 SGRAM during idle phases when all previous Read or Write operations have completed. For Reads, a burst completion is defined as when the last data element including CRC has been transmitted on the data outputs (tRDPDE timing); for Writes, a burst completion is defined as when the last data element has been written to the memory array and the result of the write CRC check has been returned to the host (tWRPDE timing).

Power-Down is entered with the Power-Down Entry (PDE) command as shown in *FIGURE 115*. The command must be paired with a CNOP1 command on the column command inputs CA[4:3]. If Power-Down entry occurs when all banks are idle, this mode is referred to as Precharge Power-Down; if Power-Down entry occurs when there is a row active in at least one bank, this mode is referred to as Active Power-Down.



NOTE 1 WCK and CA are shown aligned (tWCK2CA = 0) for illustration purposes. CA training determines the needed offset between WCK and CA.

Figure 115 — Power-Down Entry and Exit Commands

Power-Down mode can be entered while other operations such as row activation, precharge (including auto precharge), refresh or mode register set are in progress. The respective tACTPDE, tPREPDE, tREFPDE, and tMRSPDE timings must be met before entering Power-Down. The Power-Down related IDD2P or IDD3P specifications will not apply until such operations are complete.

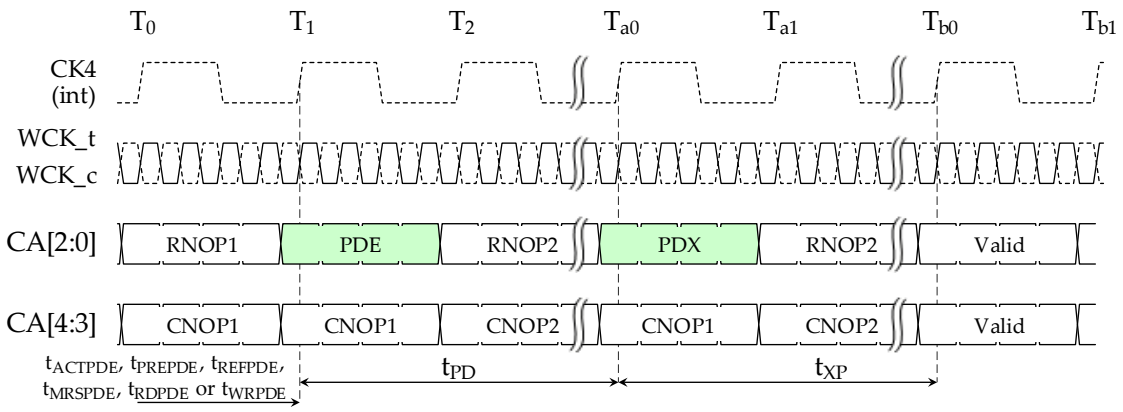
Entering Power-Down deactivates the data input and output buffers while the CA and WCK input buffers and the ERR output remain enabled, awaiting a synchronous Power-Down exit. RCK_t and RCK_c outputs may be active or in High-Z state; they will keep their state while in power-down.

A stable WCK clock must be maintained and RNOP2 and CNOP2 commands must be continuously issued while in Power-Down. The channel must be held in Power-Down for at least tPD time. The maximum Power-Down duration is limited by the refresh requirements of the device, as no Refresh commands can be issued during Power-Down. Refer to the Refresh section for detail on refresh.

Power-Down mode is synchronously exited with the Power-Down Exit (PDX) command as shown in *FIGURE 116*. RNOP2 and CNOP2 commands must be issued for at least tXP cycles following the Power-Down Exit command before the channel is back in either bank idle or bank active state and ready to receive subsequent access commands.

CA parity is evaluated with the PDE and PDX commands as well as all commands issued while in Power-Down mode when the parity calculation is enabled in MR15 OP0.

6.16 Power-Down (cont'd)



NOTE 1 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 116 — Power-Down Entry and Exit

Table 102 — Signal States During Power-Down

Signal	State	Signal	State
WCK_t, WCK_c	Active	RCK_t, RCK_c	Active or High-Z
CA	Active		
DQ, DQE	Off (ODT state)	ERR	Active

6.17 Frequency Change Sequence (fWCK)

GDDR7 SGRAMs supporting multiple WCK frequencies allow the WCK frequency to be changed during the course of normal operation following the WCK Frequency Change Sequence. By following the sequence the device's data rate can be modified whenever only a fraction of the maximum available bandwidth is required by the current work load.

WCK frequency changes can take place after entering self refresh sleep or sleep mode using the standard WCK Frequency Change procedure.

The sequence requires the device to be properly given MR10 to MR15 to prepare the setting for the new WCK frequency before self refresh sleep or sleep mode entry.

The host must issue FD_FLAG enable (MR0 OP11 = 0_B) and MR10 to MR15 respect to the new fWCK for the device to suspend MR10 to MR15 setting before self refresh sleep or sleep entry and to have MR10 to MR15 changed after self refresh sleep exit or sleep mode exit.

The sequence requires the device to be properly placed into self refresh sleep or sleep before the frequency is changed from the existing stable frequency, Foriginal, to the new desired frequency, Fnew. The WCK Change Frequency Sequence procedure also requires changes to the Frequency Range mode register using MR12 bits OP[8:4], depending on whether the feature is supported.

The DRAM vendor's datasheet shall be consulted regarding the supported frequencies for WCK Frequency Change Sequence, and any dependencies of AC timing parameters on the selected frequency.

Step 1) Wait until all commands have finished, all banks are idle or Self Refresh.

Step 2) Issue FD_FLAG (MR0 OP11=0) & MR10 to MR15 MRS commands to suspend frequency related MRS change.

If the new desired WCK clock frequency requires termination, enable the CA termination before switching to the new frequency if it is not already enabled.

Step 3) Issue SRSE or SLE command followed by NOP until tCPDED is met.

Step 4) Change the WCK clock frequency and wait until clock is stabilized.

Step 5) Exit Self Refresh Sleep or Sleep and wait for tSLX + tSLXFC + tSLX_CAT or tSLX + tSLXFC + tCSP_PRE.

Step 6) Perform CA training and issue CATX command wait for tCATX to reach intermediate state if SRSLX2CAT or SLX2CAT is enabled (MR0 OP5 or OP6 = 0).

After tCATX or tSLX + tSLXFC + tCSP_PRE, issue CSP command and wait tCSP_POST.

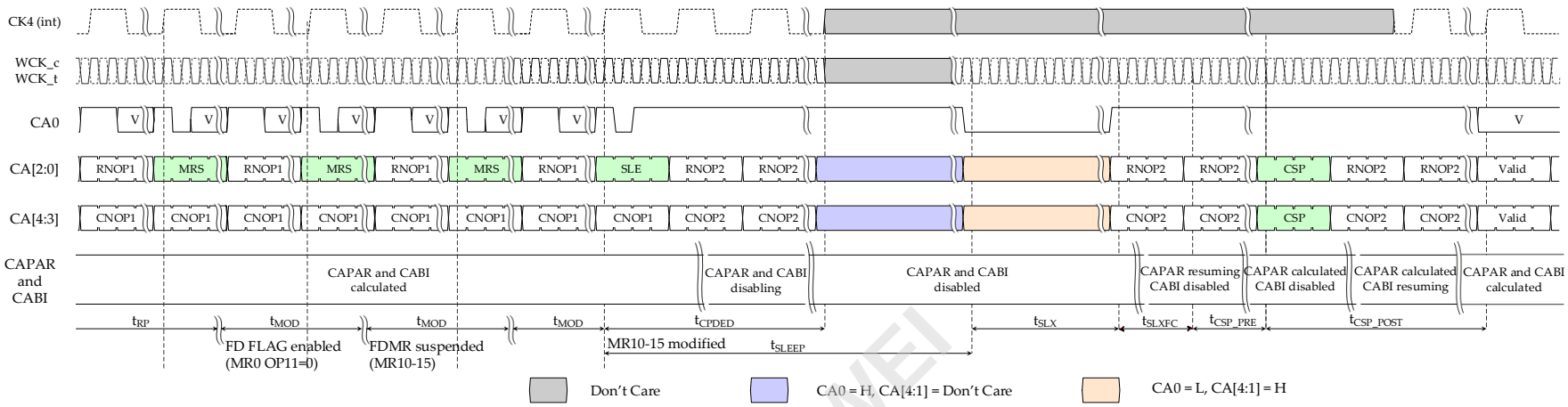
Step 7) Perform READ and WRITE training, if required.

Issue SRX command and wait for tXS if self refresh sleep mode was entered at Step 3. The self refresh exit procedure must be met.

Step 8) Device is ready for normal operation after any necessary interface training.

NOTE: Either in Step 2 or Step 7 other Mode Register settings, for example, RLmrs, WLmrs, or etc. that need to change, should be set for the new frequency before any training and the device is ready for normal operation. Otherwise, proper operation cannot be guaranteed.

6.17 Frequency Change Sequence (fWCK) (cont'd)



NOTE 1 WCK frequency change at SLEEP mode and SLX2CAT disabled for illustration

NOTE 2 The channel enters Self Refresh Sleep mode when the channel was in bank idle state and an SRFSLE command is issued instead of the SLE command.

NOTE 3 CA Training and Data Training must be executed if required after WCK frequency change.

NOTE 4 CA0 is part of the row command bus CA[2:0] but broken out and shown separately to illustrate how CA0 assumes the function of a CKE input during Sleep Mode.

NOTE 5 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 117 — Example of WCK Frequency Change

6.18 Dynamic Voltage Sequence (DVS)

GDDR7 SGRAMs supporting multiple supply voltages allow the supply voltage to be changed during the course of normal operation using the Dynamic Voltage Switching (DVS) feature. By using DVS the device's power consumption can be reduced whenever the operating frequency is supported at a lower voltage.

DVS requires the device to be properly placed into self refresh sleep or sleep mode before the voltage is changed from the existing stable voltage, Voriginal, to the new desired voltage, Vnew. The DVS procedure may also require changes to the VDD Range mode register using MR12 bits OP[1:0] and the VDDQ Range mode register using MR12 bits OP[3:2], depending on whether the feature is supported. The DRAM vendor's datasheet shall be consulted regarding the supported supply voltages for DVS, and any dependencies of AC timing parameters on the selected supply voltage. WCK frequency changes can also take place after entering self refresh sleep or sleep mode using the standard WCK Frequency Change procedure. A WCK frequency change in conjunction with DVS is required if tWCK is less than tWCKmin supported by Vnew. In this case normal device operation including self refresh sleep or sleep exit is not guaranteed without a frequency change.

Once self refresh sleep or sleep is entered, tCPDED must be met before the supply voltage is allowed to transition from Voriginal to Vnew. After VDD and VDDQ are stable at Vnew, tVS must be met to allow for internal voltages in the device to stabilize before self refresh sleep or sleep mode may be exited. During the voltage transition the voltage must not go below Vmin of the lower voltage of either Voriginal or Vnew in order to prevent false chip reset. Vmin is the minimum voltage allowed by VDD or VDDQ in the DC operating conditions table.

Step 1) Complete all operations and precharge all banks or self refresh on all active channels (either 2 or 4 depending on the configuration) on the device.

Step 2) Issue FD FLAG (MR0 OP11=0) & MR10 to MR15 MRS commands on all active channels to suspend voltage related MRS change.

Also issue other MRS commands to set VDD Range and VDDQ Range to proper values for Vnew. This step is only required when the VDD Range and VDDQ Range mode register field is supported by the device. The DRAM vendor's datasheet should be consulted to verify if the feature is supported.

Step 3) Enter self refresh sleep or sleep mode on all active channels. Self refresh sleep and Sleep entry procedure must be met.

Step 4) Wait required time tCPDED before changing voltage to Vnew.

Step 5) Change VDD and VDDQ to Vnew.

Step 6) Wait required time tVS for voltage stabilization.

Step 7) Exit sleep mode and wait for tSLX or tSLX + tCATE. The sleep exit procedure must be met.

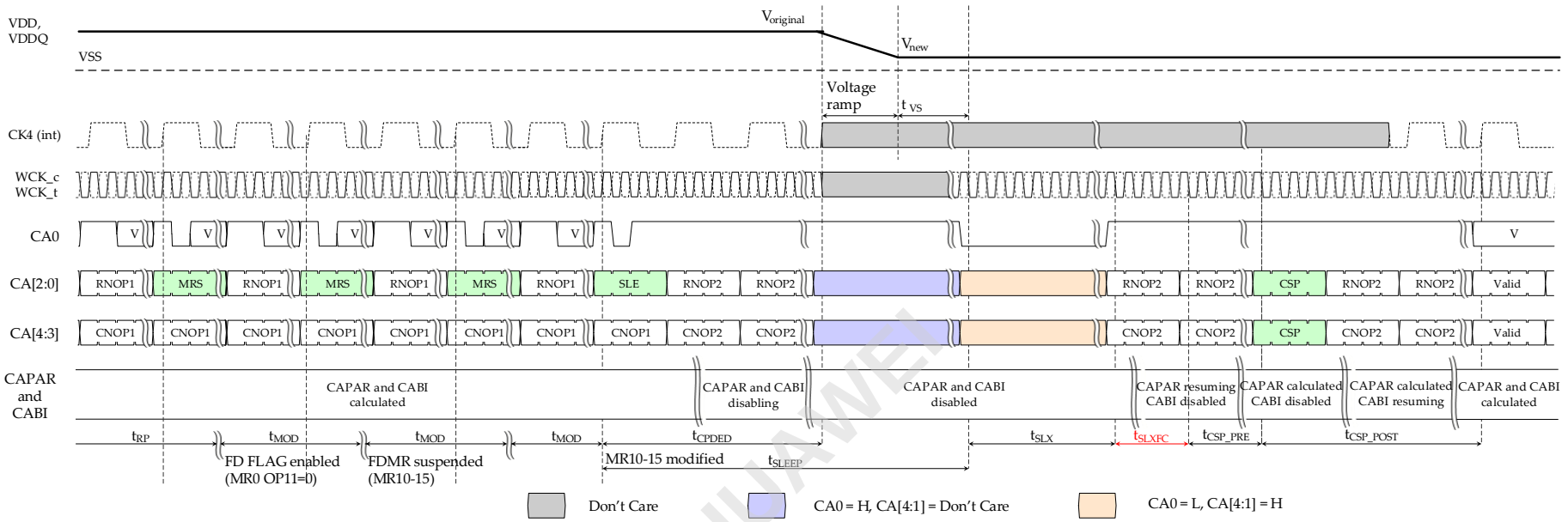
Perform CA training and issue CATX command wait for tCATX to reach intermediate state if SRSLX2CAT or SLX2CAT is enabled (MR0 OP5 or OP6 = 0). After tCSP_PRE, issue CSP command on all active channels and wait until tCSP_POST.

Step 8) Issue MRS commands to adjust mode register settings as desired (e.g., latencies, CRC on/off).

Step 9) Perform any interface training as required.

Step 10) Issue SLX command on all active channels and the sleep exit procedure must be met if self refresh sleep mode was entered at Step 3. Continue normal operation

6.18 Dynamic Voltage Sequence (DVS) (cont'd)



NOTE 1 DVS at SLEEP mode and SLX2CAT disabled for illustration purposes.

NOTE 2 The channel enters Self Refresh Sleep mode when the channel was in bank idle state and an SRFSLE command is issued instead of the SLE command.

NOTE 3 CA Training and Data Training must be executed if required after DVS.

NOTE 4 CA0 is part of the row command bus CA[2:0] but broken out and shown separately to illustrate how CA0 assumes the function of a CKE input during Sleep Mode.

NOTE 5 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

NOTE 6 The tSLXFC time is required to be observed only if the WCK frequency in MR12 OP[8:4] has changed.

Figure 118 — Example of Dynamic Voltage Change

7 Data Integrity

GDDR7 SGRAMs include several features to ensure the integrity of the data either in the DRAM cells or during Read and Write operations.

Table 103 — Data Integrity Related Features

FEATURE	ABR	Description
Cyclic Redundancy Check	CRC	Separate CRC for Read and Write data transfers
On-Die Error Checking and Correcting	OD-ECC	DRAM based ECC for integrity of the data in the core
Severity	SEV	Real- time reporting of errors from OD-ECC
Poison	PSN	Host flags data packet as corrupted data
Error Check and Scrub	Auto ECS	DRAM operates in background to detect and correct errors in the core
ECC Engine Test Mode		Test engine for OD-ECC
Command Address Parity	CAPAR	DRAM checks parity of the CA packet and notifies host if parity does not match
Command Address Parity with Command Blocking	CAPARBLK	DRAM hold commands until parity is checked
CSP Feedback (optional)	CSP FB	DRAM acknowledges CSP command after exit from sleep, self refresh sleep or CA bus training
Hard Post Package Repair	hPPR	Replace row(s) in a bank with a spare row
Refresh Management	RFM	Supplemental Refresh
Adaptive Refresh Management	ARFM	Additional thresholds for supplemental Refresh
Direct Refresh Management	DRFM	Host directed supplemental Refresh

7.1 Read and Write CRC

GDDR7 SGRAMs support Cyclic Redundancy Check (CRC) data link protection for read and write operations. The WRCRC bit in MR0 OP4 controls CRC for Writes, and the RDCRC bit in MR0 OP5 controls CRC for Reads. Both bits default to disabled and must be enabled if desired prior to a read or write operation.

The GDDR7 CRC is computed for each burst transfer on the data and associated meta data (Severity and Poison). During reads the GDDR7 SGRAM calculates and sends the CRC on the DQE signal to the host controller, that calculates the CRC on the received data and compares the result to validate the transfer. During writes the host calculates and sends the CRC to the DRAM, together with the write data, the DRAM calculates the CRC on the received data and compares it with the received CRC. In case of a mismatch the DRAM returns to the host an error flag (WRCRC2ERR) on the ERR signal if this option is enabled as described before. Refer to the *ERR SIGNAL* section for more details on the ERR signal protocol.

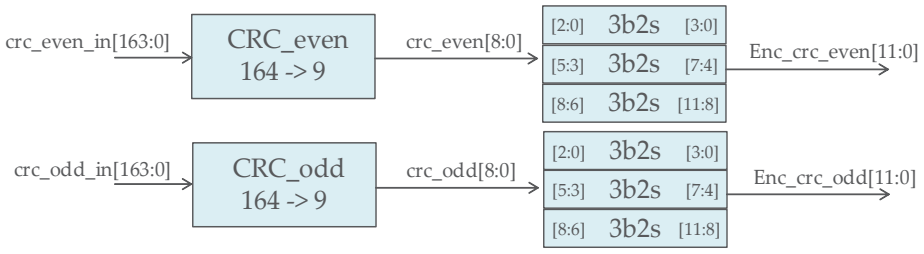
Table 104 — GDDR7 CRC Error Detection Details

Error Type	Detection Ratio[1]
Single Bit	100%
Double Bit	100%
Triple Bit	100%
Quad Bit	~99.95%
Random Odd Count	100%
Random Burst[2]	~99.99%

NOTE 1 Detection ratios are calculated for the CRC-9 (0x14b) polynomial applied on half of the burst (164 bits).

NOTE 2 Even / Odd interleaved CRC generation maximizes burst error detection (up to 18 bits)

CRC is supported in NRZ and PAM3 modes. In PAM3 mode, the CRC is calculated on the PAM3 encoded burst data plus the encoded burst metadata (poison and severity), that amounts for a total of 328 bits per transfer. The GDDR7 SGRAMs allocate 18 bits for the checksum code that are transmitted on the DQE signal, together with other burst data as shown in *FIGURE 120*. The CRC computation is performed by two CRC blocks where each block applies the same CRC-9 ($x^9+x^7+x^4+x^2+x+1$) polynomial on even and odd bits of the burst separately. The checksum code is PAM3 encoded at the output of the CRC blocks, as shown in *FIGURE 119*, in PAM3 mode as well as in NRZ mode.

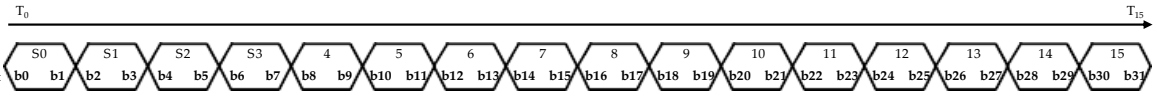


NOTE 1 The 9-bit output of the CRC blocks is always encoded to PAM3 compliant 12-bit wide codes, using 3 x 3b2S encoders as shown in the drawing, either in NRZ or PAM3 modes. For details on the 3b2S encoder please refer to the *PAM3* section.

Figure 119 — CRC-9 Even and Odd Blocks and Output Encoding

7.1 Read and Write CRC (cont'd)

In NRZ mode the same CRC scheme applies, however the calculation is carried on the 256 bits of the data burst without encoding, plus the encoded metadata. Still the CRC computes over a total of 328 bits, where the bits that would correspond to DQ8 and DQ9 (disabled in NRZ mode) and the remainder data bits in DQE, are assumed 1s for the CRC computation, as shown in *FIGURE 121*.



DQ lane to CRC input mapping

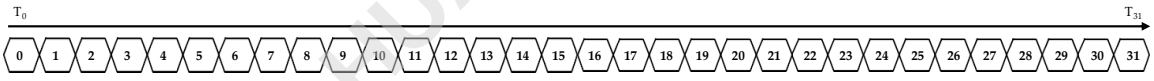
crc_even/crc_odd	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o
DQ0	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15
DQ1	16	16	17	17	18	18	19	19	20	20	21	21	22	22	23	23	24	24	25	25	26	26	27	27	28	28	29	29	30	30	31	31
DQ2	32	32	33	33	34	34	35	35	36	36	37	37	38	38	39	39	40	40	41	41	42	42	43	43	44	44	45	45	46	46	47	47
DQ3	48	48	49	49	50	50	51	51	52	52	53	53	54	54	55	55	56	56	57	57	58	58	59	59	60	60	61	61	62	62	63	63
DQ4	64	64	65	65	66	66	67	67	68	68	69	69	70	70	71	71	72	72	73	73	74	74	75	75	76	76	77	77	78	78	79	79
DQ5	80	80	81	81	82	82	83	83	84	84	85	85	86	86	87	87	88	88	89	89	90	90	91	91	92	92	93	93	94	94	95	95
DQ6	96	96	97	97	98	98	99	99	100	100	101	101	102	102	103	103	104	104	105	105	106	106	107	107	108	108	109	109	110	110	111	111
DQ7	112	112	113	113	114	114	115	115	116	116	117	117	118	118	119	119	120	120	121	121	122	122	123	123	124	124	125	125	126	126	127	127
DQ8	128	128	129	129	130	130	131	131	132	132	133	133	134	134	135	135	136	136	137	137	138	138	139	139	140	140	141	141	142	142	143	143
DQ9	144	144	145	145	146	146	147	147	148	148	149	149	150	150	151	151	152	152	153	153	154	154	155	155	156	156	157	157	158	158	159	159
DQE	160	160	161	161	162	162	163	163																								

DQE burst bit mapping[2]	Enc_MD PSN/SEV		Enc_data_DQE 6 bits pam3 data remainder						Enc_crc_even												Enc_crc_odd											
	0	1	0	1	2	3	4	5	0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5	6	7	8	9	10	11

NOTE 1 The 0 to 163 index numbers in the table illustrate the input bit order for the CRC blocks, as shown in *FIGURE 122*.

NOTE 2 The Enc_MD, Enc_Data_DQE, Enc_crc_even and Enc_crc_odd labels correspond to the input/output variables in *FIGURE 123*, the bit order matches the PAM3 burst bit mapping, please refer to the *PAM3* section for details.

Figure 120 — DQ to CRC Input/Output Bit Assignation and DQE Burst Bit Order in PAM3



DQ lane to CRC input mapping

crc_even/crc_odd	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o	e	o
DQ0	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15
DQ1	16	16	17	17	18	18	19	19	20	20	21	21	22	22	23	23	24	24	25	25	26	26	27	27	28	28	29	29	30	30	31	31
DQ2	32	32	33	33	34	34	35	35	36	36	37	37	38	38	39	39	40	40	41	41	42	42	43	43	44	44	45	45	46	46	47	47
DQ3	48	48	49	49	50	50	51	51	52	52	53	53	54	54	55	55	56	56	57	57	58	58	59	59	60	60	61	61	62	62	63	63
DQ4	64	64	65	65	66	66	67	67	68	68	69	69	70	70	71	71	72	72	73	73	74	74	75	75	76	76	77	77	78	78	79	79
DQ5	80	80	81	81	82	82	83	83	84	84	85	85	86	86	87	87	88	88	89	89	90	90	91	91	92	92	93	93	94	94	95	95
DQ6	96	96	97	97	98	98	99	99	100	100	101	101	102	102	103	103	104	104	105	105	106	106	107	107	108	108	109	109	110	110	111	111
DQ7	112	112	113	113	114	114	115	115	116	116	117	117	118	118	119	119	120	120	121	121	122	122	123	123	124	124	125	125	126	126	127	127
DQ8	128	128	129	129	130	130	131	131	132	132	133	133	134	134	135	135	136	136	137	137	138	138	139	139	140	140	141	141	142	142	143	143
DQ9	144	144	145	145	146	146	147	147	148	148	149	149	150	150	151	151	152	152	153	153	154	154	155	155	156	156	157	157	158	158	159	159
DQE	160	160	161	161	162	162	163	163																								

DQE burst bit mapping[3]	Enc_MD PSN/SEV		Enc_data DQE 6 bits pam3 data remainder						Enc_crc_even												Enc_crc_odd											
	0	1	0	1	2	3	4	5	0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5	6	7	8	9	10	11

NOTE 1 The 0 to 163 index numbers in the table illustrate the input bit order for the CRC blocks, as shown in *FIGURE 122*, and the Enc_crc_even and Enc_crc_odd describe the CRC output bit order in the DQE burst data transfer.

NOTE 2 In NRZ the data burst is not PAM3 encoded, thus only 256 data bits per transfer are transmitted. To keep the same CRC encoding scheme with 328 input bits, the bit positions that would correspond to the data transferred on DQ8 and DQ9 as well as the DQE data remainder in PAM3 mode are considered fix to one for the CRC computation and grayed out in this table.

NOTE 3 The Enc_MD, Enc_Data_DQE, Enc_crc_even and Enc_crc_odd labels correspond to the input/output variables in *FIGURE 123*, the bit order matches the PAM3 burst bit mapping, please refer to the *PAM3* section for details.

Figure 121 — DQ to CRC Input Bit Assignation and DQE Burst Bit Order in NRZ

The GDDR7 SGRAMs implementation of the 0x14b CRC-9 polynomial in RTL is shown in *FIGURE 122*, the input and output bit ordering for the CRC blocks is shown in *FIGURE 120* for PAM3 and in *FIGURE 121* for NRZ.

7.1 Read and Write CRC (cont'd)

```
// This module contains a combinatorial parallel 9-bit CRC implementation

// Polynomial 0x14b: X^9+X^7+X^4+X^2+X^1+1


module crc( din, cout );


input [163:0] din;

output [8:0] cout;

assign cout[0] = din[0] ^ din[2] ^ din[4] ^ din[5] ^ din[6] ^ din[7] ^ din[11] ^ din[12] ^ din[17] ^ din[18] ^ din[19] ^ din[20] ^ din[23] ^ din[26] ^ din[27] ^ din[30] ^ din[32] ^ din[33] ^ din[34] ^ din[39] ^ din[43] ^
din[44] ^ din[45] ^ din[49] ^ din[51] ^ din[52] ^ din[53] ^ din[54] ^ din[55] ^ din[56] ^ din[57] ^ din[63] ^ din[64] ^ din[65] ^ din[67] ^ din[68] ^ din[70] ^ din[72] ^ din[73] ^ din[74] ^ din[76] ^ din[77] ^ din[78] ^
din[80] ^ din[83] ^ din[85] ^ din[87] ^ din[91] ^ din[92] ^ din[94] ^ din[95] ^ din[96] ^ din[99] ^ din[100] ^ din[102] ^ din[103] ^ din[105] ^ din[106] ^ din[110] ^ din[111] ^ din[112] ^ din[113] ^ din[114] ^ din[115] ^
din[117] ^ din[118] ^ din[119] ^ din[120] ^ din[122] ^ din[123] ^ din[126] ^ din[131] ^ din[132] ^ din[135] ^ din[136] ^ din[140] ^ din[147] ^ din[148] ^ din[150] ^ din[153] ^ din[154] ^ din[156] ^ din[158] ^ din[161];

assign cout[1] = din[0] ^ din[1] ^ din[2] ^ din[3] ^ din[4] ^ din[8] ^ din[11] ^ din[13] ^ din[17] ^ din[21] ^ din[23] ^ din[24] ^ din[26] ^ din[28] ^ din[30] ^ din[31] ^ din[32] ^ din[35] ^ din[39] ^ din[40] ^ din[43] ^
din[46] ^ din[49] ^ din[50] ^ din[51] ^ din[58] ^ din[63] ^ din[66] ^ din[67] ^ din[69] ^ din[70] ^ din[71] ^ din[72] ^ din[75] ^ din[76] ^ din[79] ^ din[80] ^ din[81] ^ din[83] ^ din[84] ^ din[85] ^ din[86] ^ din[87] ^
din[88] ^ din[91] ^ din[93] ^ din[94] ^ din[97] ^ din[99] ^ din[101] ^ din[102] ^ din[104] ^ din[105] ^ din[107] ^ din[110] ^ din[116] ^ din[117] ^ din[121] ^ din[122] ^ din[124] ^ din[126] ^ din[127] ^ din[131] ^ din[133]
^ din[135] ^ din[137] ^ din[140] ^ din[141] ^ din[147] ^ din[149] ^ din[150] ^ din[151] ^ din[153] ^ din[155] ^ din[156] ^ din[157] ^ din[158] ^ din[159] ^ din[161] ^ din[162];

assign cout[2] = din[0] ^ din[1] ^ din[3] ^ din[6] ^ din[7] ^ din[9] ^ din[11] ^ din[14] ^ din[17] ^ din[19] ^ din[20] ^ din[22] ^ din[23] ^ din[24] ^ din[25] ^ din[26] ^ din[29] ^ din[30] ^ din[31] ^ din[34] ^ din[36] ^
din[39] ^ din[40] ^ din[41] ^ din[43] ^ din[45] ^ din[47] ^ din[49] ^ din[50] ^ din[53] ^ din[54] ^ din[55] ^ din[56] ^ din[57] ^ din[59] ^ din[63] ^ din[65] ^ din[71] ^ din[74] ^ din[78] ^ din[81] ^ din[82] ^ din[83] ^
din[84] ^ din[86] ^ din[88] ^ din[89] ^ din[91] ^ din[96] ^ din[98] ^ din[99] ^ din[108] ^ din[110] ^ din[112] ^ din[113] ^ din[114] ^ din[115] ^ din[119] ^ din[120] ^ din[125] ^ din[126] ^ din[127] ^ din[128] ^ din[131] ^
din[134] ^ din[135] ^ din[138] ^ din[140] ^ din[141] ^ din[142] ^ din[147] ^ din[151] ^ din[152] ^ din[153] ^ din[157] ^ din[159] ^ din[160] ^ din[161] ^ din[162] ^ din[163];

assign cout[3] = din[1] ^ din[2] ^ din[4] ^ din[7] ^ din[8] ^ din[10] ^ din[12] ^ din[15] ^ din[18] ^ din[20] ^ din[21] ^ din[23] ^ din[24] ^ din[25] ^ din[26] ^ din[27] ^ din[30] ^ din[31] ^ din[32] ^ din[35] ^ din[37] ^
din[40] ^ din[41] ^ din[42] ^ din[44] ^ din[46] ^ din[48] ^ din[50] ^ din[51] ^ din[54] ^ din[55] ^ din[56] ^ din[57] ^ din[58] ^ din[60] ^ din[64] ^ din[66] ^ din[72] ^ din[75] ^ din[79] ^ din[82] ^ din[83] ^ din[84] ^
din[85] ^ din[87] ^ din[89] ^ din[90] ^ din[92] ^ din[97] ^ din[99] ^ din[100] ^ din[109] ^ din[111] ^ din[113] ^ din[114] ^ din[115] ^ din[116] ^ din[120] ^ din[121] ^ din[126] ^ din[127] ^ din[128] ^ din[129] ^ din[132] ^
din[135] ^ din[136] ^ din[139] ^ din[141] ^ din[142] ^ din[143] ^ din[148] ^ din[152] ^ din[153] ^ din[154] ^ din[158] ^ din[160] ^ din[161] ^ din[162] ^ din[163];

assign cout[4] = din[0] ^ din[3] ^ din[4] ^ din[6] ^ din[7] ^ din[8] ^ din[9] ^ din[12] ^ din[13] ^ din[16] ^ din[17] ^ din[18] ^ din[20] ^ din[21] ^ din[22] ^ din[23] ^ din[24] ^ din[25] ^ din[28] ^ din[30] ^ din[31] ^
din[34] ^ din[36] ^ din[38] ^ din[39] ^ din[41] ^ din[42] ^ din[44] ^ din[47] ^ din[53] ^ din[54] ^ din[58] ^ din[59] ^ din[61] ^ din[63] ^ din[64] ^ din[68] ^ din[70] ^ din[72] ^ din[74] ^ din[77] ^ din[78] ^ din[84] ^
din[86] ^ din[87] ^ din[88] ^ din[90] ^ din[92] ^ din[93] ^ din[94] ^ din[95] ^ din[96] ^ din[98] ^ din[99] ^ din[101] ^ din[102] ^ din[103] ^ din[105] ^ din[106] ^ din[111] ^ din[113] ^ din[116] ^ din[118] ^ din[119] ^
din[120] ^ din[121] ^ din[123] ^ din[126] ^ din[127] ^ din[128] ^ din[129] ^ din[130] ^ din[131] ^ din[132] ^ din[133] ^ din[135] ^ din[137] ^ din[142] ^ din[143] ^ din[144] ^ din[147] ^ din[148] ^ din[149] ^ din[150] ^
din[155] ^ din[156] ^ din[158] ^ din[159] ^ din[162] ^ din[163];

assign cout[5] = din[1] ^ din[4] ^ din[5] ^ din[7] ^ din[8] ^ din[9] ^ din[10] ^ din[13] ^ din[14] ^ din[17] ^ din[18] ^ din[19] ^ din[21] ^ din[22] ^ din[23] ^ din[24] ^ din[25] ^ din[26] ^ din[29] ^ din[31] ^ din[32] ^
din[35] ^ din[37] ^ din[39] ^ din[40] ^ din[42] ^ din[43] ^ din[45] ^ din[48] ^ din[54] ^ din[55] ^ din[59] ^ din[60] ^ din[62] ^ din[64] ^ din[65] ^ din[69] ^ din[71] ^ din[73] ^ din[75] ^ din[78] ^ din[79] ^ din[85] ^
din[87] ^ din[88] ^ din[89] ^ din[91] ^ din[93] ^ din[94] ^ din[95] ^ din[96] ^ din[97] ^ din[99] ^ din[100] ^ din[102] ^ din[103] ^ din[104] ^ din[106] ^ din[107] ^ din[112] ^ din[114] ^ din[117] ^ din[119] ^ din[120] ^
din[121] ^ din[122] ^ din[124] ^ din[127] ^ din[128] ^ din[129] ^ din[130] ^ din[131] ^ din[132] ^ din[133] ^ din[134] ^ din[136] ^ din[138] ^ din[143] ^ din[144] ^ din[145] ^ din[148] ^ din[149] ^ din[150] ^ din[151] ^
din[156] ^ din[157] ^ din[159] ^ din[160] ^ din[163];

assign cout[6] = din[2] ^ din[5] ^ din[6] ^ din[8] ^ din[9] ^ din[10] ^ din[11] ^ din[14] ^ din[15] ^ din[18] ^ din[19] ^ din[20] ^ din[22] ^ din[23] ^ din[24] ^ din[25] ^ din[26] ^ din[27] ^ din[30] ^ din[32] ^ din[33] ^
din[36] ^ din[38] ^ din[40] ^ din[41] ^ din[43] ^ din[44] ^ din[46] ^ din[49] ^ din[55] ^ din[56] ^ din[60] ^ din[61] ^ din[63] ^ din[65] ^ din[66] ^ din[70] ^ din[72] ^ din[74] ^ din[76] ^ din[79] ^ din[80] ^ din[86] ^
din[88] ^ din[89] ^ din[90] ^ din[92] ^ din[94] ^ din[95] ^ din[96] ^ din[97] ^ din[98] ^ din[100] ^ din[101] ^ din[103] ^ din[104] ^ din[105] ^ din[107] ^ din[108] ^ din[113] ^ din[115] ^ din[118] ^ din[120] ^ din[121] ^
din[122] ^ din[123] ^ din[125] ^ din[128] ^ din[129] ^ din[130] ^ din[131] ^ din[132] ^ din[133] ^ din[134] ^ din[135] ^ din[137] ^ din[139] ^ din[144] ^ din[145] ^ din[146] ^ din[149] ^ din[150] ^ din[151] ^ din[152] ^
din[157] ^ din[158] ^ din[160] ^ din[161];

assign cout[7] = din[0] ^ din[2] ^ din[3] ^ din[4] ^ din[5] ^ din[9] ^ din[10] ^ din[15] ^ din[16] ^ din[17] ^ din[18] ^ din[21] ^ din[24] ^ din[25] ^ din[28] ^ din[30] ^ din[31] ^ din[32] ^ din[37] ^ din[41] ^ din[42] ^
din[43] ^ din[47] ^ din[49] ^ din[50] ^ din[51] ^ din[52] ^ din[53] ^ din[54] ^ din[55] ^ din[61] ^ din[62] ^ din[63] ^ din[65] ^ din[66] ^ din[68] ^ din[70] ^ din[71] ^ din[72] ^ din[74] ^ din[75] ^ din[76] ^ din[78] ^
din[81] ^ din[83] ^ din[85] ^ din[89] ^ din[90] ^ din[92] ^ din[93] ^ din[94] ^ din[97] ^ din[98] ^ din[100] ^ din[101] ^ din[103] ^ din[104] ^ din[108] ^ din[109] ^ din[110] ^ din[111] ^ din[112] ^ din[113] ^ din[115] ^
din[116] ^ din[117] ^ din[118] ^ din[120] ^ din[121] ^ din[124] ^ din[129] ^ din[130] ^ din[133] ^ din[134] ^ din[138] ^ din[145] ^ din[146] ^ din[148] ^ din[151] ^ din[152] ^ din[154] ^ din[156] ^ din[159] ^ din[162];

assign cout[8] = din[1] ^ din[3] ^ din[4] ^ din[5] ^ din[6] ^ din[10] ^ din[11] ^ din[16] ^ din[17] ^ din[18] ^ din[19] ^ din[22] ^ din[25] ^ din[26] ^ din[29] ^ din[31] ^ din[32] ^ din[33] ^ din[38] ^ din[42] ^ din[43] ^
din[44] ^ din[48] ^ din[50] ^ din[51] ^ din[52] ^ din[53] ^ din[54] ^ din[55] ^ din[56] ^ din[62] ^ din[63] ^ din[64] ^ din[66] ^ din[67] ^ din[69] ^ din[71] ^ din[72] ^ din[73] ^ din[75] ^ din[76] ^ din[77] ^ din[79] ^
din[82] ^ din[84] ^ din[86] ^ din[90] ^ din[91] ^ din[93] ^ din[94] ^ din[95] ^ din[98] ^ din[99] ^ din[101] ^ din[102] ^ din[104] ^ din[105] ^ din[109] ^ din[110] ^ din[111] ^ din[112] ^ din[113] ^ din[114] ^ din[116] ^
din[117] ^ din[118] ^ din[119] ^ din[121] ^ din[122] ^ din[125] ^ din[130] ^ din[131] ^ din[134] ^ din[135] ^ din[139] ^ din[146] ^ din[147] ^ din[149] ^ din[152] ^ din[153] ^ din[155] ^ din[157] ^ din[160] ^ din[163];



endmodule
```

Figure 122 — RTL Implementation of the CRC Calculation Based on the Given CRC-9 Polynomial

7.1 Read and Write CRC (cont'd)

```
assign crc_even_in = CRC_en? PAM3 ? { Enc_data_DQE[4,2,0], Enc_MD[0],
    ,Enc_data_DQ9[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ8[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ7[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ6[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc data DQ5[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ4[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc data DQ3[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ2[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ1[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]
    ,Enc_data_DQ0[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]} :
    {3'b111, Enc_MD[0], 32'hFFFFFFFF
    ,DQ7[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ6[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ5[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ4[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ3[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ2[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ1[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0],
    ,DQ0[30,28,26,24,22,20,18,16,14,12,10,8,6,4,2,0]} : {164{1'b1}};

assign crc_even[8:0] = crc(crc_even_in[163:0]);
assign Enc_crc_even[3:0] = CRC_en ? 3b2s_crc(CRC_even[2:0]) : 4'hF;
assign Enc crc even[7:4] = CRC en ? 3b2s crc(CRC even[5:3]) : 4'hF;
assign Enc_crc_even[11:8] = CRC_en ? 3b2s_crc(CRC_even[8:6]) : 4'hF;

assign crc_odd_in = CRC_en? PAM3 ? {Enc_data_DQE[5,3,1], Enc_MD[1],
    ,Enc_data_DQ9[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ8[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ7[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc data DQ6[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ5[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc data DQ4[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ3[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ2[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ1[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]
    ,Enc_data_DQ0[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]} :
{3'b111, Enc_MD[1], 32'hFFFFFFFF,
    ,DQ7[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ6[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ5[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ4[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ3[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ2[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ1[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1],
    ,DQ0[31,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1]} : {164{1'b1}};

assign crc_odd[8:0] = crc(crc_odd_in[163:0]);
assign Enc crc odd[3:0] = CRC en ? 3b2s crc(CRC odd[2:0]) : 4'hF;
assign Enc_crc_odd[7:4] = CRC_en ? 3b2s_crc(CRC_odd[5:3]) : 4'hF;
assign Enc crc odd[11:8] = CRC en ? 3b2s crc(CRC odd[8:6]) : 4'hF;
```

Figure 123 — RTL Implementation of the CRC Input and Output Bit Assignments

7.2 On-Die ECC

7.2.1 On_Die ECC Overview

The GDDR7 device uses on-die ECC, an error scrubbing mechanism, and a real-time error transparency protocol to achieve a high level of system RAS.

GDDR7 on-die ECC features:

- Correction and detection minimum capabilities on each 256b access
  - 100% 1b error correction
  - 100% 2b error detection
  - 99.3% average of >=3b error detection (value optionally included in vendor datasheet)
- On-die ECC real-time transparency
  - One bit in READ data packet to signal ‘no error’ vs ‘uncorrectable error’
- Automated on-die error scrubbing mechanism
  - Auto-ECS during REFab and SRF operation has MR for enable/disable
  - Errors are only logged during ECS

An overview of an example GDDR7 on-die ECC engine is shown in *FIGURE 124*.

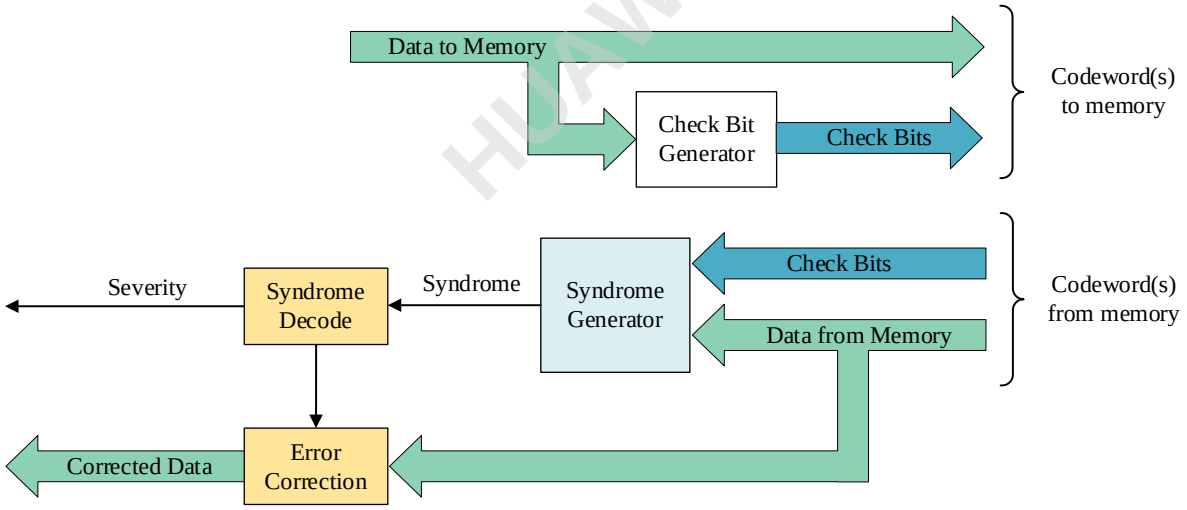


Figure 124 — On-die ECC Overview Diagram Example

7.2.2 On-Die ECC Requirements

On-die ECC Engine:

GDDR7 on-die ECC has a codeword size dependent on the DRAM architecture to achieve the minimum error correction/detection capabilities listed below for each 256b access:

- 100% 1b error correction
- 100% 2b error detection
- 99.3% average of >=3b error detection (value optionally included in vendor datasheet)

There will be a minimum of 16b parity per 256b user data. An example implementation is 256b user data + 9b SEC + 7b CRC for a total codeword size of 272b. The specific ECC H-matrix used and the number of codewords is implementation specific.

On Reads the DRAM corrects all errors that are less than or equal to error correcting capability of the ECC engine before transmitting the corrected data to the memory controller. The DRAM shall not write the corrected data back to the array during a read cycle.

On writes the DRAM computes the parity bits and writes the data and parity bits to the array.

7.2.3 DRAM Fault Isolation Requirements

There are no specific fault boundaries defined.

7.2.4 Poison

GDDR7 supports the ability for a host to flag individual data packet (256b) as poisoned. Host may use this to mark known-bad data packets. Poison support is enabled/disabled using MR0 OP10.

Data packets transferred between host and DRAM on writes or reads include a flag indicating the data packet is poisoned.

For write data packets with this flag set the DRAM stores the information that this data packet is poisoned.

Host must include valid data in the write data packet received by the DRAM. DRAM is not required to store the provided write data when the poison flag is set, though it may choose to do so.

On reads when the DRAM sees valid poison information it will set the poison flag in the read data packet. Valid data must be included as part of the read data packet. It may be a vendor specific fixed pattern or contents from the DRAM.

How poison is stored in the DRAM is vendor dependent. It may be an additional physical bit, a virtual bit, a specific pattern of error, or some other method. Regardless of the method used the poison information must be covered by the on-die ECC. The DRAM must be able to differentiate between good data, correctable data, poison data, and uncorrectable data. Known-bad poisoned data may deteriorate into uncorrectable data whose poison status cannot be relied upon.

If the host disables poison after poisoning addresses, the data and severity returned by the DRAM when reading addresses that were previously poisoned is undefined. The host must write without poison to any previously poisoned addresses to clear any poison flags if valid data is expected at those addresses. The poison flags can be cleared before or after disabling of the poison MRS.

7.2.5 On-Die ECC Transparency Protocol

A GDDR7 device must provide transparency of actions by the on-die ECC engine. The specific information to be conveyed and the method of conveyance is given in *TABLE 105*.

Table 105 — Transparency Attributes and their Access/Control Mechanism

Attribute	Operation	Transparency Mechanism
Real-time severity metadata for UE only	RD/RDA	SEV/PSN bit in Read packet (DQE signal)
Logging of an Auto ECS error	ECS	Info Read (IRD) command

Auto ECS Severity and Poison: During Auto ECS read operations, the DRAM will not record and ignore properly poisoned addresses in either the UE address capture or UE count. The DRAM will check for new CE or UE errors and record them based on the logging configuration mode register settings.

Real-time Severity Metadata: The GDDR7 device includes a mode register, MR0 OP9, to enable or disable the severity signaling by the GDDR7 device in real-time to the host. The severity of an error (SEV) denotes the outcome of the on-die ECC processing over a codeword(s) during a READ operation. Uncorrected Error (UE) and No Error (NE) are the valid severity the DRAM will provide the host in real-time.

During Reads or Reads with Autoprecharge the DRAM will check to see if the data was poisoned by the host and/or will check for severity if the options are enabled. In PAM3 mode, Severity and Poison are encoded on DQE using one PAM3 symbol with UE taking precedent in case of both a UE and PSN event. Severity and Poison transmission in PAM3 mode will use the encoding shown in *TABLE 106* for each transaction.

In NRZ mode, Severity data (read only) and Poison data (read and write) is transmitted in the DQE packet as the binary representation of the encoded symbol from *TABLE 106.*

Table 106 — DQE Severity/Poison (SEV/PSN) Bit Truth Table

Poison	Severity	Value (Trit/Binary)
0	0	‘+1 (11)’ = No Error (NE)
1	0	‘0 (01)’ = Poisoned (PSN)
X	1	‘-1 (00)’ = Uncorrected Error (UE)
NOTE MR0 OP9 (Severity) shall be turned on when MR0 OP10 (Poison) is enabled.		

7.2.6 Error Check and Scrub (Auto ECS)

The GDDR7 device will implement an Auto ECS function. Auto ECS will use on-die ECC and operate in the background during REFab (with TR=L), Self Refresh (with TR=L) and Self Refresh Sleep periods when enabled using the ECS_ON field in MR22 OP11. Auto ECS allows the DRAM to internally read, detect errors, correct errors, and write back corrected data bits to the array (scrub errors). Any errors corrected by on-die ECC during Auto ECS must be logged in the transparency registers according to the rules described in this section.

During Auto ECS, the internal Read-Modify-Write cycle will:

1. Read the entire code-word(s) from the DRAM array.
2. If the ECC engine detects a correctable error, the error will be corrected, and code-word(s) will be written back to DRAM.
3. If an error is detected in the code-word(s) and is uncorrectable, the bits in the code-word(s) will not be modified. The code-word(s) must not be written back to DRAM.
4. If the ECC engine detects no error, the DRAM may choose to write the resultant code-word(s) back to DRAM or not.

The DRAM can only guarantee valid ECS operations if array bits are written to prior to enabling ECS operations, thus enabling DRAM to calculate the proper initial parity bits.

Auto ECS related MR control:

Table 107 — ECS Mode Registers

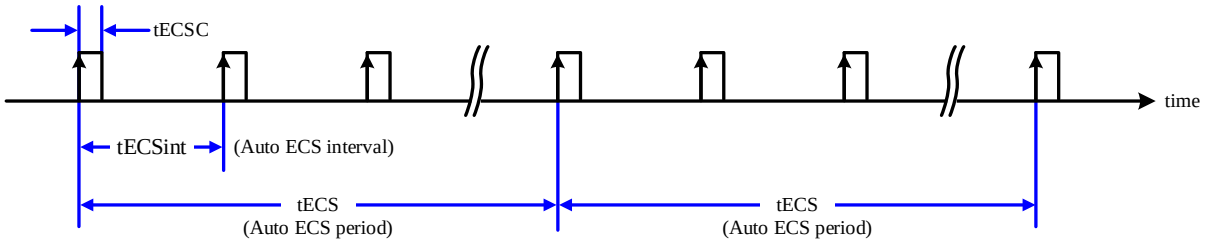
ECS Mode	Mode Register Value (Default all Disabled)
Auto ECS (ECS_ON)	MR22 OP11: 1_B = enabled, 0_B = disabled
Auto ECS reset (ECS_RESET)	MR22 OP10: 1_B = reset (self-clearing), 0_B = disabled
Auto ECS error flag reset (ECS_FLAG_RESET)	MR22 OP9: 1_B = reset (self-clearing), 0_B = disabled
Auto ECS error log overwrite rules (ECS_LOG_RULES)	MR22 OP8: 1_B = overwrite, 0_B = maintain
NOTE 1 REFab commands with TR = L used for ECS will count toward refresh credit. NOTE 2 When ECS enabled, the host must issue REFab commands with TR = L at an average rate of tECSint.	

Auto ECS shall be programmed during DRAM initialization and shall not be changed once the first ECS operation occurs unless followed by an ECS reset, otherwise an unknown operation could result during subsequent ECS operations.

7.2 On-Die ECC (cont'd)

Auto ECS related timing parameters:

The ECS operation timing is shown in *FIGURE 125*.



tECSC: Max time for GDDR7 to complete ECS operation
tECSint: Average ECS interval to cover all codewords in a specified period of tECS (e.g., 24h)
tECS: Period of time to complete ECS on all codeword

Figure 125 — ECS Operation Timing

In order to complete a full Error Check and Scrub within the recommended tECS (e.g., 24 hours), the average periodic interval of ECS operations (tECSint) is 86,400 seconds divided by the total number of codewords as described in *TABLE 108*. The number of ECS operations is density dependent.

Table 108 — tECSint per Channel

Gb per channel	4 Gb	6 Gb	8 Gb	12 Gb	16 Gb	24 Gb	32 Gb
272b code-words per channel	2^24	2^24*1.5	2^25	2^25*1.5	2^26	2^26*1.5	2^27
tECSint [ms] per channel	5.15	3.43	2.57	1.72	1.29	0.86	0.64

For the GDDR7 device to perform ECS operations when Auto ECS is enabled, the host needs to either put the DRAM into i) Self Refresh with the Training flag set to Low (TR=L), ii) Self Refresh Sleep or iii) issue periodic REFab commands with the Training flag set to Low (TR=L). ECS operations will not be performed when the host issues a REFab command with TR= H, or in Self Refresh with TR=H.

In the case of REFab with TR=L, the maximum average spacing between REFab commands with TR=L for the DRAM to complete the automatic scrub is tECSint. Meeting this REFab requirement allows the DRAM to perform the ECS operations without placing additional restrictions on refresh mode usage, i.e., mix of REFab and REFpb. REFab commands with TR=L issued in excess of required by the DRAM for ECS operations (one per tECSint) may also be used by the DRAM for normal refresh operation. tECSint interval timing for the maximum spacing between REFab commands with TR=L or another Self Refresh entry is allowed to restart upon Self Refresh exit. Auto ECS operations that are triggered with a REFab with TR=L must be completed by the device within tECSC. tECSC must be less than or equal to tRFCab.

7.2 On-Die ECC (cont'd)

Expiration of tECS period:

For each Auto ECS operation, ECS Address Counters increment the column address after each internal ECS WR command such that the next code word and check bits are selected. Once the column counter wraps (all code words and check bits on the row have been accessed), the row counter will increment until all code words on each of the rows within a bank are accessed. When the row counter wraps (all rows within the bank have been accessed), the bank counter will increment, and the next bank will repeat the process of accessing each code word. After all the code words within the DRAM are read, corrected, and written once, the bank counter will wrap, and the process begins again with the next Auto ECS operation.

Auto ECS Error Types, Logging, and Info Read:

GDDR7 Auto ECS supports three ECS Error types as outlined in *TABLE 109*. The ECS Error Logs consist of seven Info Register Addresses (IRA [24:18]), that are read out using Info Read command. ECS Error Logs consist of ECS Error Flags, Address Logs and Count Logs. See the *INFO READ* section for details on the Info Read protocol.

Table 109 — ECS Error Types

ECS Error Type	ECS Error Flag	ECS Error Registers
Uncorrected Error Address Logging (UEAL)	ECS UE FLAG	1. UE Address Bank (IRA 18 DQ[3:0]) 2. UE Address Row (IRA [22:21]) 3. UE Address Column (IRA 23) (Vendor specific)
Uncorrected Error Count Logging (UECL)	ECS UE COUNT FLAG	1. UE Count (IRA 24 DQ[3:0]) (Vendor specific)
Corrected Errors per row Address Logging (CEAL)	ECS CE FLAG	1. Max CE per ROW Address Bank (IRA 18 DQ[7:4]) 2. Max CE per ROW Address Row (IRA [20:19])

UECL and UEAL Column Address Logging are vendor specific features. The host can determine if the features are supported by reading out the UE Count Support bit and UE COL Support bit in Vendor ID2 (IRA 2).

ECS FLAG, located in IRA 16, provides a way for the host to poll whether the ECS mechanism has a new ECS error and therefore new ECS error log. The ECS FLAG and all three ECS Error Flags in *TABLE 109* default to 1'b0 and change to a 1'b1 in case of an error.

ECS FLAG equation:

The ECS FLAG is the logical OR of the three ECS Error Flags.

ECS FLAG (IRA 16) = ECS CE FLAG (IRA 24) | ECS UE FLAG (IRA 24) | ECS UE COUNT FLAG (IRA 24)

When the ECS FLAG field indicates a new Auto ECS event has been logged, the expected behavior is the host will read out the three ECS Error flags in IRA 24 to identify which type of error event or events has occurred and proceed accordingly. The seven ECS Error logs [7:0] located in IRA [24:18] can be read out in any order.

The ECS Error Log registers associated with UEAL and CEAL and the UECL UE count register have no defined default upon power up, device reset or ECS reset. The logs are qualified as new with the associated ECS Error flag being set to a 1'b1. The logs remain unchanged when read out, however all three ECS Error Flags are reset to 1'b0 when the host reads out IRA 24.

7.2 On-Die ECC (cont'd)

Uncorrected Error Address Logging (UEAL)

1) When the on-die ECC detects an Uncorrected Error (UE), the device first determines if the log can be updated based on the rules programmed in the ECS_LOG_RULES mode register (MR22 OP8).
    a. For UE errors, if the ECS_LOG_RULES register is set to maintain (0_B) the DRAM will not update the log if the ECS UE FLAG = 1'b1, otherwise the log will be updated.
    b. If the MR is set to overwrite (1_B) the log will always be updated.
2) If the log can be updated, the DRAM address of the error is logged in the form of the Bank Address in IRA 18 and the Row Address in IRA [22:21].
3) If supported, the device will update the Column Address in IRA 23.
    a. If not supported the Column Address in IRA 23 will default to 8'b11111111
4) The ECS UE FLAG in IRA 24 is updated to 1'b1.
5) The ECS FLAG is updated to 1'b1 per the ECS FLAG equation.
6) The error is logged within tECSC.

Uncorrected Error Count Logging (UECL)

1) During a tECS period, if UECL is supported, the device will count the number of rows that have at least one Uncorrected Error up to a max of 15 rows.
2) At the end of the tECS period the device will update the count in the UE Count register in IRA 24.
3) The ECS UE COUNT FLAG in IRA 24 is updated to 1'b1 at the expiration of every tECS period.
4) The ECS FLAG is updated to 1'b1 per the ECS FLAG equation.
5) The error is logged within tECSC.
    a. The device will reset the internal UE counter with the start of each tECS period after the UE Count register and flag are updated.
    b. The UE Count register is not reset when the flag and log are read out and remains unchanged until the expiration of the next tECS period.
    c. If UE count is not supported, the ECS UE COUNT FLAG will default to 1'b0, the UE Count will default to 4'b1111 and both will never update when tECS expires.

Corrected Errors per Row Address Logging (CEAL)

1) During Auto ECS operation the on-die ECC will scrub a row by reading every column address in the row under scrub.
2) The device will track the number of Corrected Errors in the row.
3) Once the full row has been scrubbed the device first determines if the log can be updated.
    a. If the ECS_LOG_RULES mode register is set to maintain (0_B), the log can only be updated if the number of errors in the row is greater than 8 errors in the individual row and the number of errors is greater than the previous errors logged
    b. If the ECS_LOG_RULES mode register is set to overwrite (1_B), the log can be updated if the number of errors in the row is greater than 8 errors
4) If the log can be updated:
    a. the DRAM address of the error is logged in the form of Bank Address in IRA 18 and Row Address in IRA [20:19].
    b. The ECS CE FLAG in IRA 24 is updated to 1'b1.
    c. The ECS FLAG is updated to 1b'1 per the ECS FLAG equation.
    d. The error is logged within tECSC.

7.2 On-Die ECC (cont'd)

Reset of ECS Error Flags:

There are four independent methods for clearing the ECS Error Flags:

1. The host reads IRA 24
2. The host may issue a ECS Flag reset using MR22 OP9
3. The host may issue an ECS reset according to MR22 OP10
4. The host may issue device RESET

Reset of ECS Function:

The host may issue an ECS reset using MR22 OP10 to reset all aspects of the ECS function including but not limited to the current scrub address counter in the DRAM, internal running UE count, UE Count log, and the ECS FLAGS.

Reading of Error logs and Auto ECS operation:

GDDR7 DRAMs avoid situations where the host is reading out the logs while the DRAM is in the process of updating the logs by not allow both operations in a state. *FIGURE 126* illustrates the states and their associated commands for the Auto ECS operation as well as that states and their associated commands that the host can use to read the ECS logs.

When Auto ECS operation is enabled, and the host enters SELF Refresh Sleep the Auto ECS operation will continue until the Self refresh exit command is received by the DRAM. When the Self-Refresh Sleep exit is direct to Self Refresh state with TR=H (SRSLX2CAT MR0 OP6=1_B) the ECS logs will not be guaranteed to be updated until tSRSX_ECSLOG_UPD has expired. Reading the log before tSRSX_ECSLOG_UPD has expired may result in existing or invalid data being read out. The ECS logs include IRA16 DQ7 (ECS Error) and IRA[24:18] (ECS Error Log 1-7).

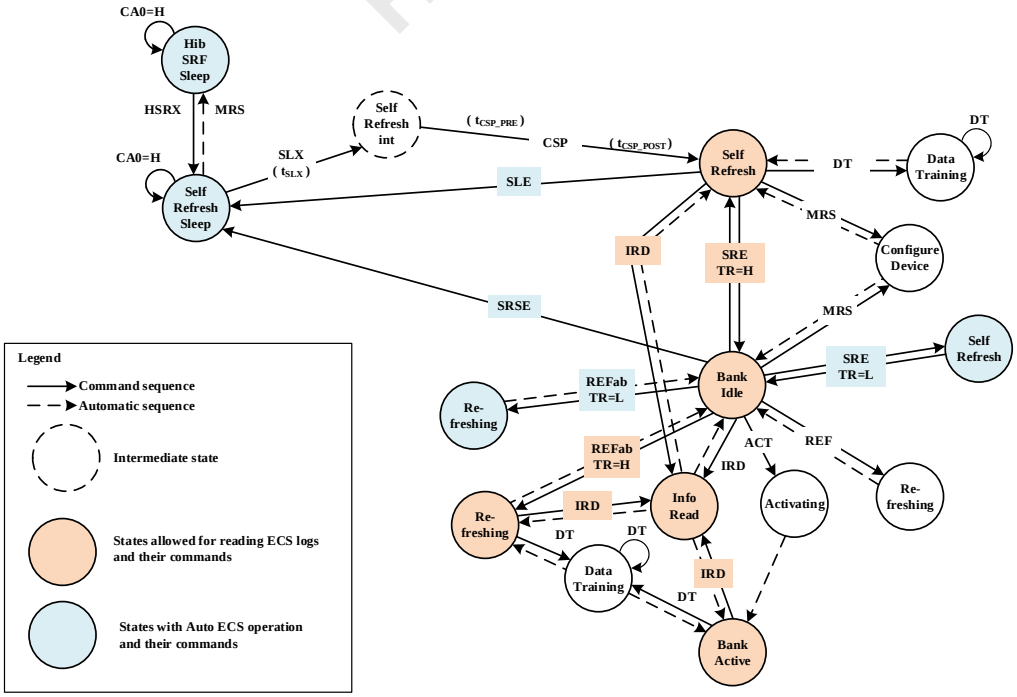
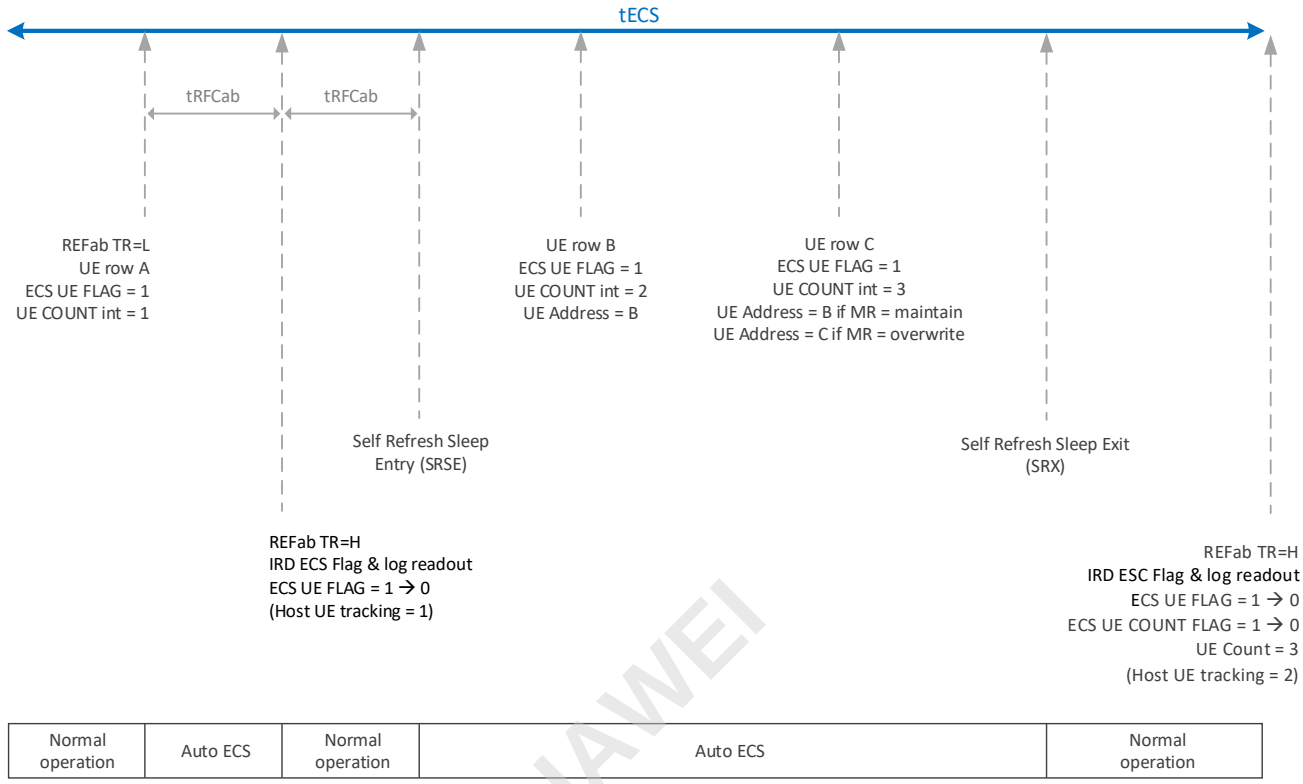


Figure 126 — Auto ECS and ECS Log Read Out State Diagram

7.2 On-Die ECC (cont'd)



NOTES:

1. ECS_LOG_RULES mode register (MR10 OP8) determines whether to maintain the UE Address when ECS UE FLAG = 1 and or to overwrite the UE Address.
2. Rows A, B and C = arbitrary rows with Uncorrected Errors (UE).

Figure 127 — Example ECS Operation – UEAL and UECL

7.3 ECC Engine Test Mode

GDDR7 devices provide ECC engine testing method of the on-die ECC engine only, not error access into the core. The outcome of the error injection is reported according to the transparency protocol.

GDDR7 devices will report results of the ECC engine test on the DQE using the coding in *TABLE 111*. GDDR7 devices may optionally provide 0 (01) for CE if Info Read Register Address (IRA) 2 DQ 2 indicates that the optional 0 (01) = CE is supported. If Info Read Register Address 2 DQ 2 indicates that optional 0 (01) = CE is not supported, GDDR7 devices provide +1 (11) for any CE in the ECC engine test mode on the DQE. GDDR7 devices in 2 channel mode may require two passes to complete the ECC engine test. IRA3 DQ4 indicates if the device requires two passes or just one pass. If the device requires two passes, the ECC_2CH mode register (MR30 OP8) can be set to either first pass with MR30 OP8=0_B or to second pass with MR30 OP8=1_B and must be set to 1_B to check the second test result after confirmation of the first pass. If the device does not require two passes, the ECC_2CH mode register is not required to be implemented by the device.

Table 110 — ECC Engine Test Modes

Selection by MRS	ECC Engine Test Mode
ECC Engine Test Mode (ECC_TM, MR30 OP0)	0_B – Normal Operation (Default) 1_B – ECC Engine Test Mode
Error Vector Patterns (ECC_VEC, MR30 OP1)	0_B – CW0 (Codeword0) Data ‘1’ means error bit and Data ‘0’ means non-error bit 1_B – CW1 (Codeword1) Data ‘0’ means error bit and Data ‘1’ means non-error bit
ECC Parity Bit Error Injection (ECC_PAR_SEL, MR30 OP[7:2]	ECC parity bit error injection 00 $XXXX_B$: No error injection 01 0000_B: Error injected on parity bit 0 01 0001_B: Error injected on parity bit 1 … 01 1110_B: Error injected on parity bit 14 01 1111_B: Error injected on parity bit 15 10 $XXXX_B$: RFU

While in the ECC engine test mode in *TABLE 110*,

1. WR will function as an error injection command, Write DQ data is error injection pattern (CW0 or CW1 by MR30 OP1)
2. RD will function as an outcome output command, Read DQ/SEV data is the outcome of ECC engine test

Table 111 — Severity Report According to IRA 2 DQ 2 (ECC_ENG_CE) Status

Severity (Trit/Binary)	IRA 2 DQ 2 = 1_B (Default) (ECC_ENG_CE)	IRA 2 DQ 2 = 0_B (Optional) (ECC_ENG_CE)
+1 (11)	NE, CE	NE
0 (01)	N/A	CE
-1 (00)	UE	UE

7.3 ECC Engine Test Mode (cont'd)

The following sequence must be satisfied to perform a functional On-die ECC engine test mode of GDDR7 DRAM. See *FIGURE 128* and *TABLE 112*.

1. The GDDR7 device registers Mode Register Set command (MRS) by MR30 OP[7:0] for the entry of On-die ECC engine test mode in *TABLE 110*. Error severity reporting must be enabled via the SEVERITY in MR0 OP9. The Poison in MR0 OP10 must be disabled.

2. As an example, in *TABLE 112 — EXAMPLE OF ERROR VECTORS, PARITY BIT ERROR INJECTION AND CORRESPONDING SEVERITY* for the engine test, write "1" as Error and "0" as NE(No Error) in the case of CW0 mode. The symbol boundary is vendor specific, and output and severity information are determined according to the error type injected by the host and the parity error type injection by MR30 OP[7:2]. The error injection to parity bits are controlled by MR30 OP[7:2] and only one-bit error injection in the parity bits is possible for ECC engine test.

   CW0[255:0] is a 256-bit all '0' vector and CW1[255:0] is a 256-bit all '1' vector used to generate parity bits of ECC engine. Parity bits according to CW0 or CW1 are generated by on-die ECC engine. If CW0 is selected, parity bits for CW0 is prepared for ECC engine test. If CW1 is selected, parity bits for CW1 is prepared for ECC engine test.

   Memory controller is allowed to inject only one-bit error to the parity bits generated by on-die ECC engine to test the ECC engine and the location of on-bit error in parity bits is controlled by MR30 OP[7:2].

   Write data[255:0] received by DQ's are transferred to ECC engine without generating new corresponding parity bits. Therefore, 1's in write data is considered as error bits if CW0 is selected to generate parity bits. In the other case, 0's in write data is considered as error bits because CW1 is selected to generate parity bits.

3. To check the result of engine test, read the output after tWTRSB.
   A. The DQs will show the correction data as ALL "0" when the DATA is NE or SBE(CE) in the case of CW0. Also, the host shall ignore read data in case of UE.
   B. The Severity will indicate NE or UE (optionally indicating CE). Severity information is real-time signaling (refer to severity and poison encoder/decoder in *PAM3 BURST ENCODING* section for more details)

4. Repeat the 2, 3, 4 sequence and the operation for the engine test after tRTW.
   A. e.g.) Mode entry - WR-RD - WR-RD - WR-RD - …
      in this case, a single WR must be followed by a single RD.

The mapping between DQ and DATA[255:0] is vendor specific. When the MRS bit is enabled, the core is not accessed, and the data pattern is interpreted as an error vector. When GDDR7 is in the ECC Engine Test Mode, it does not guarantee data retention and the only allowed commands are WR, RD and MRS to disable this test mode.

There are many implementations of OD-ECC according to DRAM architecture. So it is recommended to vary bank address BA[3:0] in engine test mode.

7.3 ECC Engine Test Mode (cont'd)

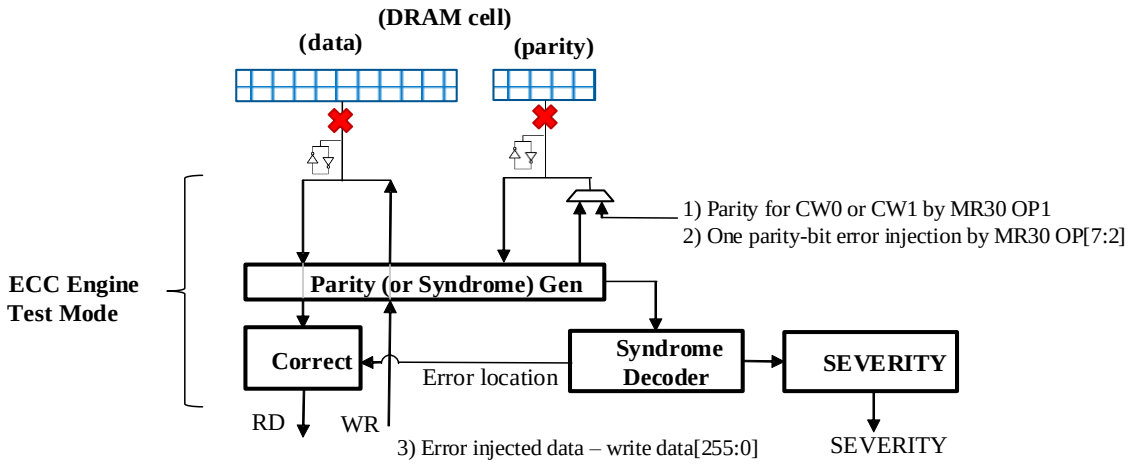


Figure 128 — Example of the Block Diagram of On-die ECC Engine and Path for ECC Engine Test Mode

Table 112 — Example of Error Vectors, Parity Bit Error Injection and Corresponding Severity IRA 2 DQ2 = 1_B

Expected Severity	Error Vector Pattern (MR30 OP1)	Error Vector Input[255:0] (Write data)	Parity Bit Error Injection [15:0] (by MR30 OP[7:2])	Error Vector Output[255:0] (Read data)	Severity Result IRA 2 DQ2 = 1_B	Note
NE	CW0	000…000000	000…000	000…000000	NE	1
	CW1	111…111111	000…000	111…111111		2
NE	CW0	100…000000	000…000	000…000000	NE	1
		000…000000	100…000	000…000000		1
	CW1	011…111111	000…000	111…111111		2
		111…111111	100…000	111…111111		2
UE	CW0	100…000000	100…000	100…0000000	UE	1
		110…000000	000…000	110…0000000		1
	CW1	011…111111	100…000	011…111111		2
		001…111111	000…000	001…111111		2

NOTE 1 CW0 indicates that 1 means the error bit and 0 means normal bit.
NOTE 2 CW1 indicates that 0 means the error bit and 1 means normal bit.

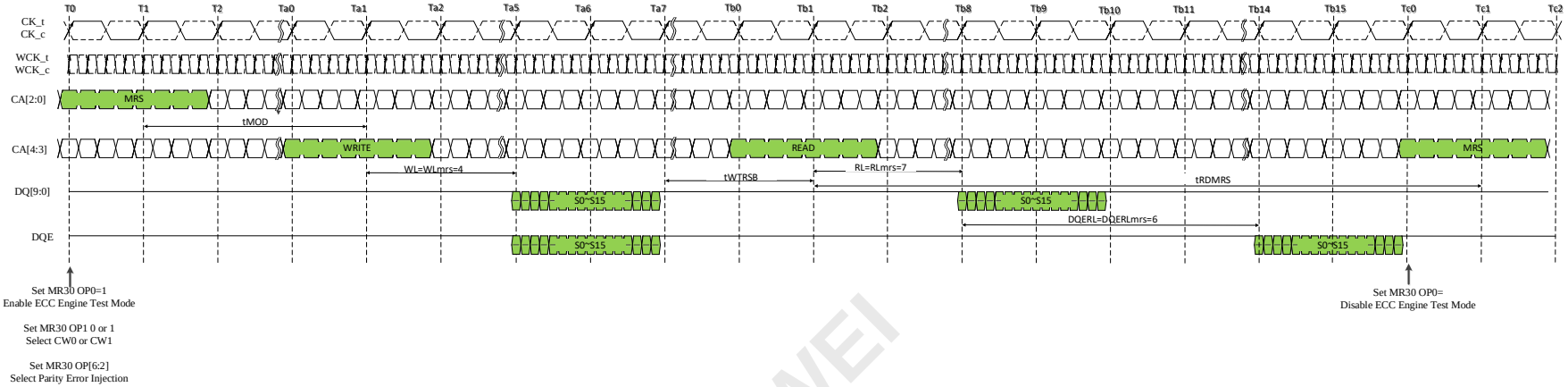
7.3 ECC Engine Test Mode (cont'd)

Table 113 — Example of Error Vectors, Parity Bit Error Injection and Corresponding Severity IRA 2 DQ2 = 0_B (Optional)

Expected Severity	Error Vector Pattern (MR30 OP1)	Error Vector Input[255:0] (Write data)	Parity Bit Error Injection [15:0] (by MR30 OP[7:2])	Error Vector Output[255:0] (Read data)	Severity Result IRA 2 DQ2 = 0_B	Note
NE	CW0	000…000000	000…000	000…000000	NE	1
	CW1	111…111111	000…000	111…111111		2
NE	CW0	100…000000	000…000	000…000000	CE	1
		000…000000	100…000	000…000000		1
	CW1	011…111111	000…000	111…111111		2
		111…111111	100…000	111…111111		2
UE	CW0	100…000000	100…000	100…0000000	UE	1
		110…000000	000…000	110…0000000		1
	CW1	011…111111	100…000	011…111111		2
		001…111111	000…000	001…111111		2

NOTE 1 CW0 indicates that 1 means the error bit and 0 means normal bit.
NOTE 2 CW1 indicates that 0 means the error bit and 1 means normal bit.

7.3 ECC Engine Test Mode (cont'd)



NOTES:

1. WRITE and READ address must be the same for ECC Engine Test Mode.
2. WRITE and READ commands don't require a preceding ACT command for ECC Engine Test Mode.
3. The only allowed commands are WR, RD, and MRS to disable ECC Engine Test Mode.
4. WL = 4 and RL = 7 are shown as an example.
5. Da, ..., Da+ 15 = data-in for WRITE command in PAM3 mode. Db, ..., Db+15 = data-out for READ command in PAM3 mode.
6. tWTR should be tWTRSB by both WRITE and READ access the same bank for ECC Engine Test Mode.
7. SEVERITY on is mandatory to verify on-die ECC transparency.
8. If CRC is enabled during READ or WRITE, CRC calculation result will be transferred via DQE pin.

Figure 129 — Timing Diagram of ECC Engine Test Mode

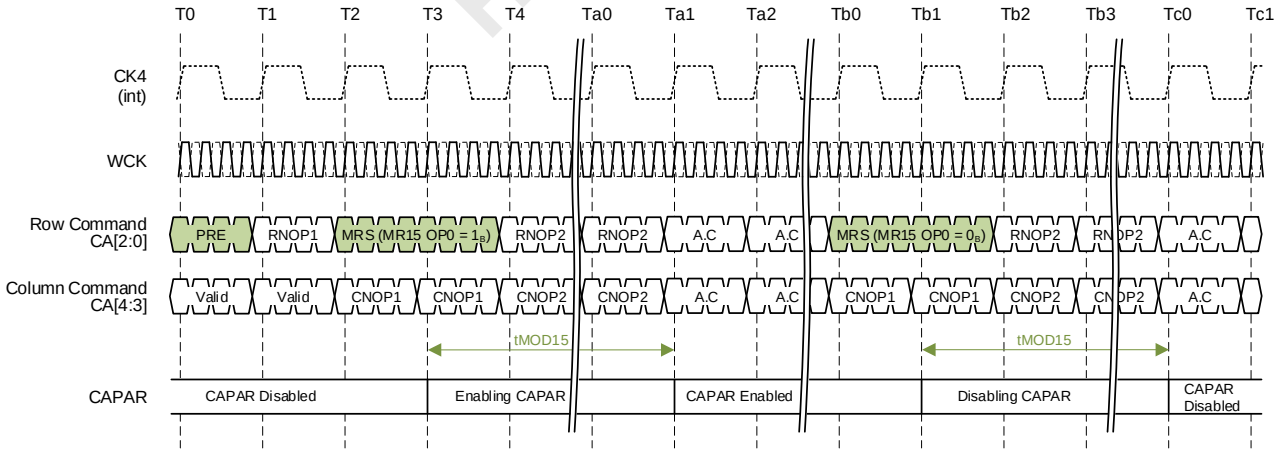
7.4 Command Address Parity (CAPAR) Protocol

GDDR7 SGRAMs support Command Address Parity (CAPAR) to improve the integrity of the Command Address (CA) bus.

If enabled, CAPAR is checked on all commands expect CATX and SLX, and the DRAM will signal the host of any CAPAR errors. The CAPAR bit is sent on CA4 as part of the column command but covers both the row (CA[2:0]) and column(CA[4:3]) command buses. See the COMMAND TRUTH TABLE and CABI AND CAPAR sections for more details on which commands include the CAPAR bit and the encode/decode of CAPAR. With CAPAR enabled and Command blocking (CAPARBLK) disabled, the DRAM will continue to run and execute any command it decodes even after a CAPAR error. See *Figure 132* and *Figure 133* for the behavior of the ERR signal with multiple CAPAR errors.

CAPAR is enabled using MR15 OP0. The default state of CAPAR is disabled. Since CAPAR is part of the FDMR registers, the FD_FLAG (MR0 OP11) must be set to 1_B if the CAPAR is to be enabled before Sleep entry. In this case, either before or at the same time as CAPAR is enabled, the CAPAR2ERR mode register (MR15 OP[11:8]) must be programmed as there is no default for CAPAR2ERR. The DRAM may begin to check parity on the next CK4 cycle following the MRS command that enables CAPAR. The DRAM will have the checking enabled latest when tMOD15 has expired after that MRS command.

Any changes to CAPAR registers in MR15 (CAPAR, CAPARBLK and CAPAR2ERR) require tMOD15. The only legal commands during tMOD15 are RNOP2 and CNOP2 to ensure no toggle on the CA bus as shown in *FIGURE 130*. The definition of RNOP2 and CNOP2 meet the condition for even CA parity. CAPAR2ERR cannot be programmed to a new value when using FD_FLAG=1_B while CAPAR is enabled. In the case of the FD_FLAG = 0_B, the checking may begin to check parity on the next CK4 cycle following the CSP command but will have the checking enabled after tCSP_POST has expired.

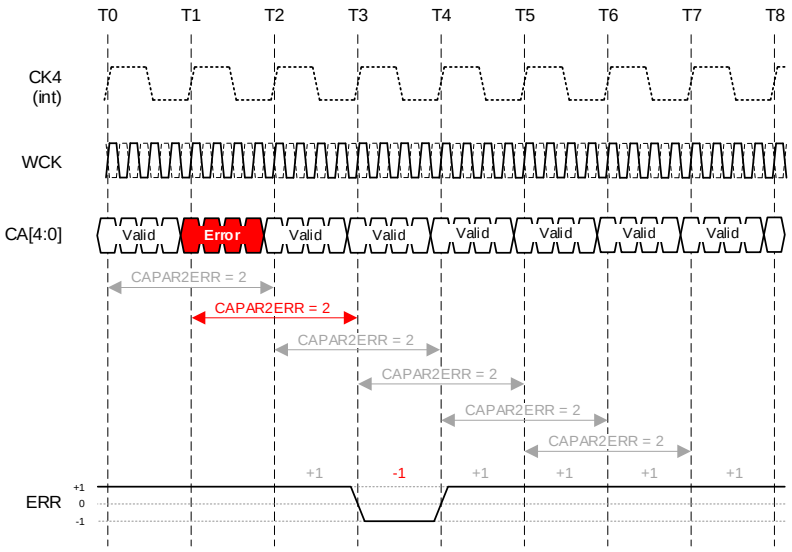


NOTES:

1. A Precharge command as shown in the figure shall not be issued when the channel is in Self Refresh state.
2. A.C. = Any command allowed in bank idle or Self Refresh state.
3. Any changes to CAPAR related registers in MR15 when FD_FLAG = 1_B require RNOP2 and CNOP2 for tMOD15.
4. Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

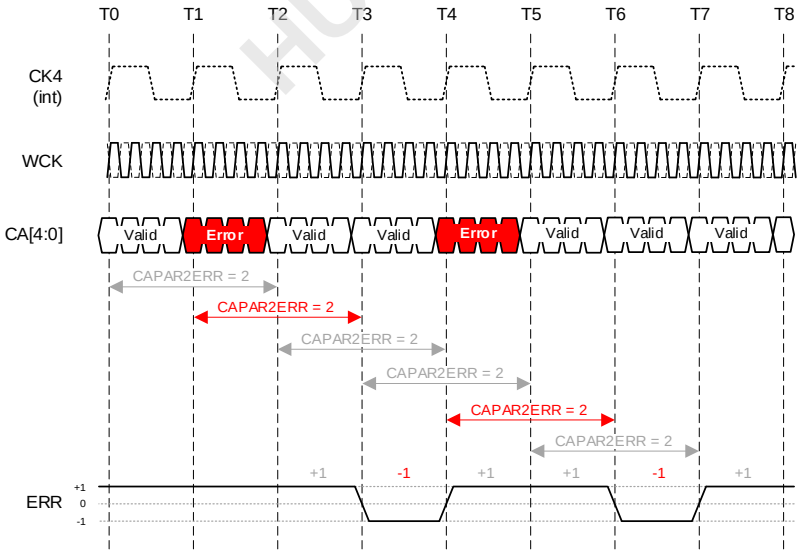
Figure 130 — Enabling and Disabling Command Address Parity (CAPAR) with FD_FLAG = 1_B

7.4 Command Address Parity (CAPAR) (cont'd)



NOTES:
1. CAPAR2ERR = 2 is shown as an example for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. tWCK2CA and tWCK2ERRO = 0 are shown for illustration purposes.

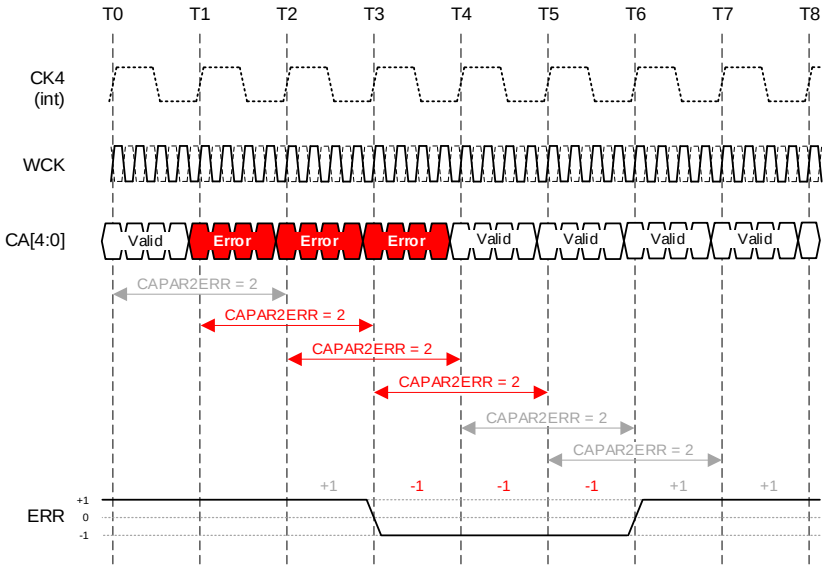
Figure 131 — Single Command Address Parity Error with Command Blocking Disabled



NOTES:
1. CAPAR2ERR = 2 is shown as an example for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. tWCK2CA and tWCK2ERRO = 0 are shown for illustration purposes.

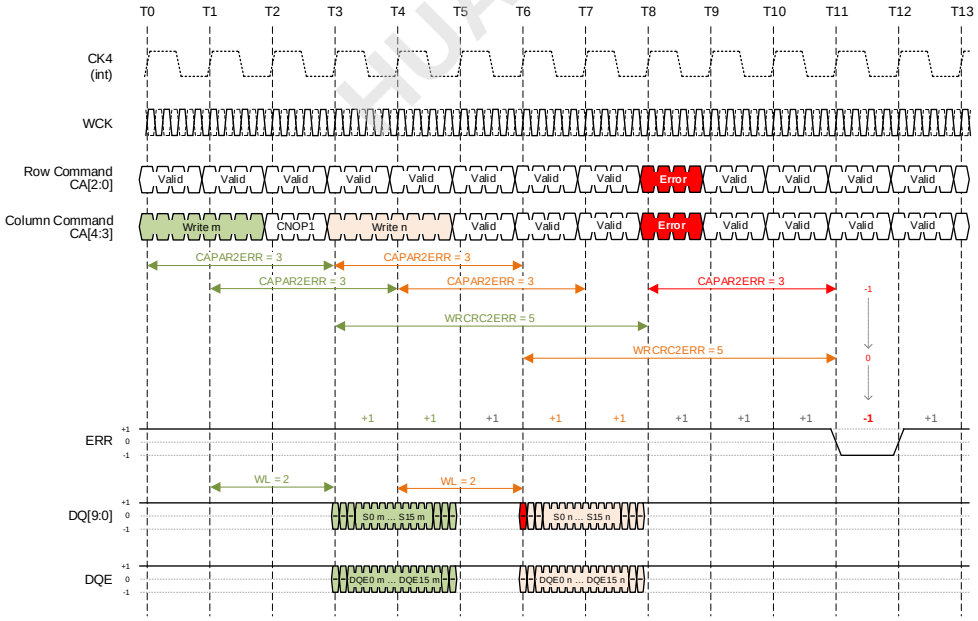
Figure 132 — Multiple Command Address Parity Errors with Command Blocking Disabled

7.4 Command Address Parity (CAPAR) (cont'd)



NOTES:
1. CAPAR2ERR = 2 is shown as an example for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. tWCK2CA and tWCK2ERRO = 0 are shown for illustration purposes.

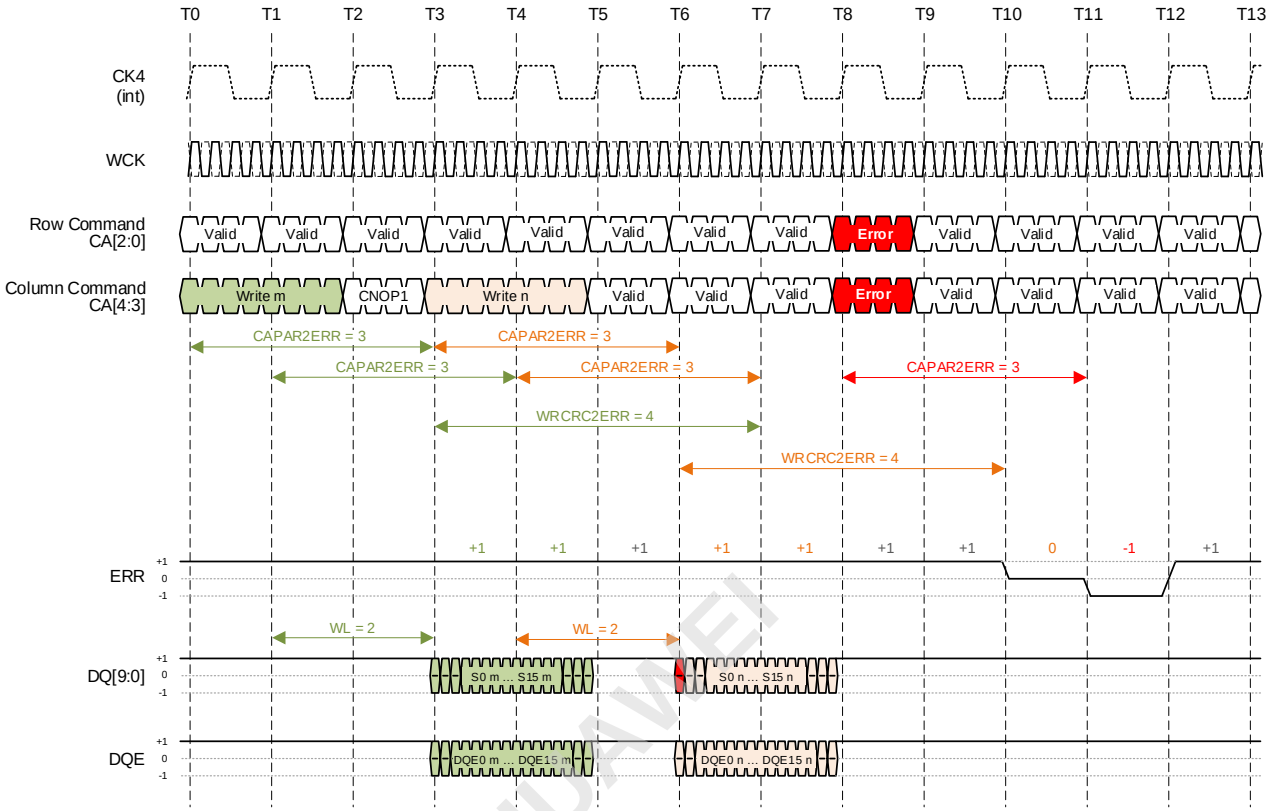
Figure 133 — Persistent Command Address Parity Errors with Command Blocking Disabled



NOTES:
1. WL = 2, CAPAR2ERR = 3 and WRCRC2ERR = 5 are shown as examples for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO.
4. tWCK2CA, tWCK2DQI and tWCK2ERRO = 0 are shown for illustration purposes.
5. WRCRC error on the first symbol of the second Write is shown for illustration purposes.

Figure 134 — WRCRC and CAPAR ERR Signaling (Same Cycle)

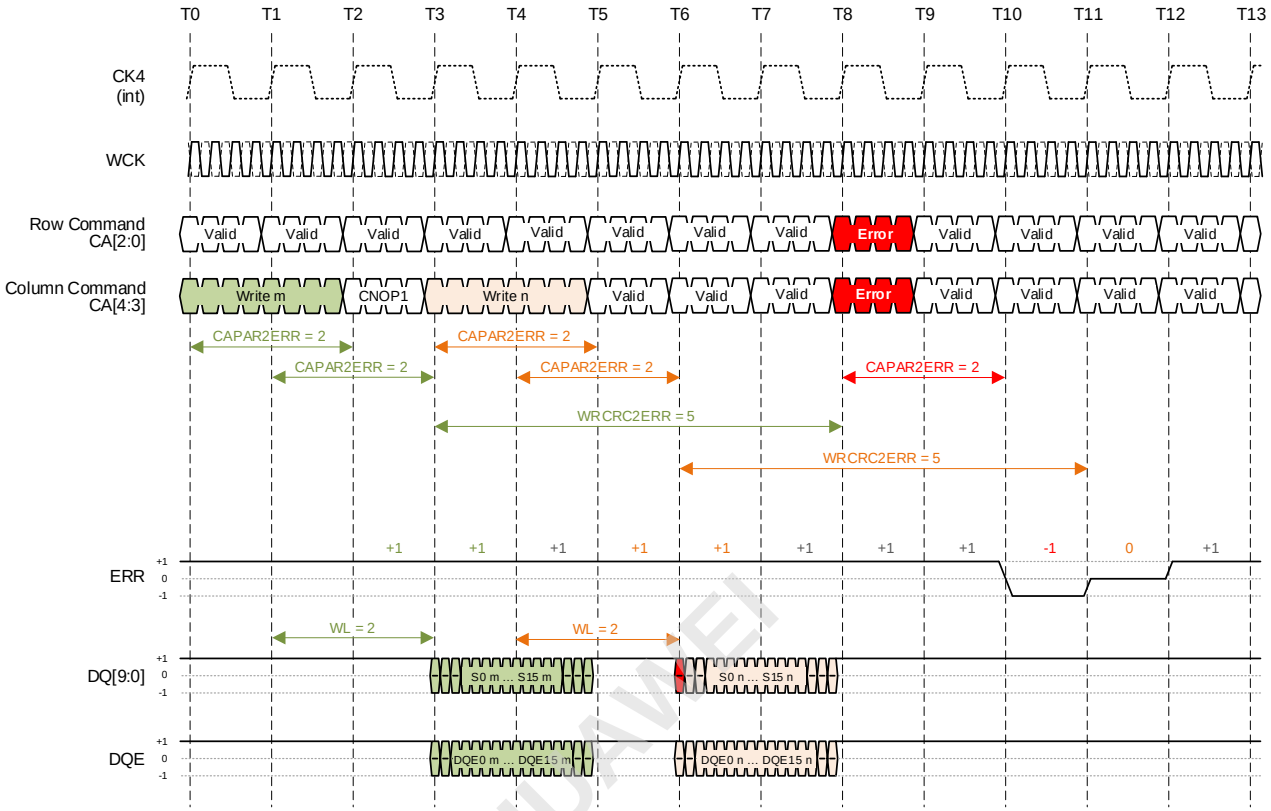
7.4 Command Address Parity (CAPAR) (cont'd)



NOTES:
1. WL = 2, CAPAR2ERR = 3 and WRCRC2ERR = 4 are shown as examples for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO.
4. tWCK2CA, tWCK2DQI and tWCK2ERRO = 0 are shown for illustration purposes.
5. WRCRC error on the first symbol of the second Write is shown for illustration purposes.

Figure 135 — WRCRC and CAPAR ERR Signaling (WRCRC before CAPAR)

7.4 Command Address Parity (CAPAR) (cont'd)



NOTES:
1. WL = 2, CAPAR2ERR = 2 and WRCRC2ERR = 5 are shown as examples for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. Write CRC Error Latency = WRCRC2ERR * tCK4 + tWCK2ERRO.
4. tWCK2CA, tWCK2DQI and tWCK2ERRO = 0 are shown for illustration purposes.
5. WRCRC error on the first symbol of the second Write is shown for illustration purposes.

Figure 136 — WRCRC and CAPAR ERR Signaling (CAPAR before WRCRC)

7.5 CAPAR with Command Blocking (CAPARBLK)

Once the host enables Command blocking using the CAPARBLK mode register (MR15 OP1 = 1_B, then the GDDR7 DRAM must ensure that there is no parity error before executing the received command.

CAPAR must be enabled either before or at the same time CAPARBLK is enabled. CAPARBLK must be disabled (MR15 OP1 = 0_B) either before or at the same time as CAPAR is disabled (MR15 OP0 = 0_B). CAPARBLK is disabled by default. *Figure 139* illustrates enabling CAPARBLK after CAPAR has been enabled and disabling CAPARBLK but leaving CAPAR enabled. Since CAPARBLK is part of the FDMR registers, the FD_FLAG must be set to 1_B if the CAPARBLK is to be enabled without the need for Sleep entry. In the case of enabling/disabling CAPARBLK with FD_FLAG = 1_B, tMOD15 is required and the only legal commands in tMOD15 are RNOP2 and CNOP2.

There is a delay before the command is executed as the command is held while CAPAR is being checked. Once parity has been verified to have no error then the command is released and executed inside the DRAM. The AC timings and Mode registers in *Table 114* may have different values depending on whether command blocking is enabled or disabled. Any Mode Register that requires a different value programmed when command blocking is enabled must be programmed after tMOD15 or tCSP_POST has expired depending on the state of the FD_FLAG.

GDDR7 devices support either implicit (IRA3 DQ[3:2] = 11_B); explicit (IRA3 DQ[3:2] = 10_B); or both implicit and explicit CAPARBLK_LAT (IRA3 DQ[3:2] = 00_B). If the DRAM supports both, the host has the flexibility to program the CAPARBLK_LAT control register (MR15 OP6) to support either implicit or explicit. For implicit only or explicit only, the CAPARBLK_LAT control register (MR15 OP6) is ignored and only required to be programmed with both. Refer to vendor datasheets to see which method is supported.

FIGURE 137 illustrates the case where the device supports implicit CA parity blocking latency or implicit is programmed into the CAPARBLK_LAT control register (MR15 OP6 = 0_B) when both supported. In the case of implicit CAPARBLK_LAT the device specification will include *TABLE 142* for the timings that may have different values depending on whether command blocking is enabled or disabled.

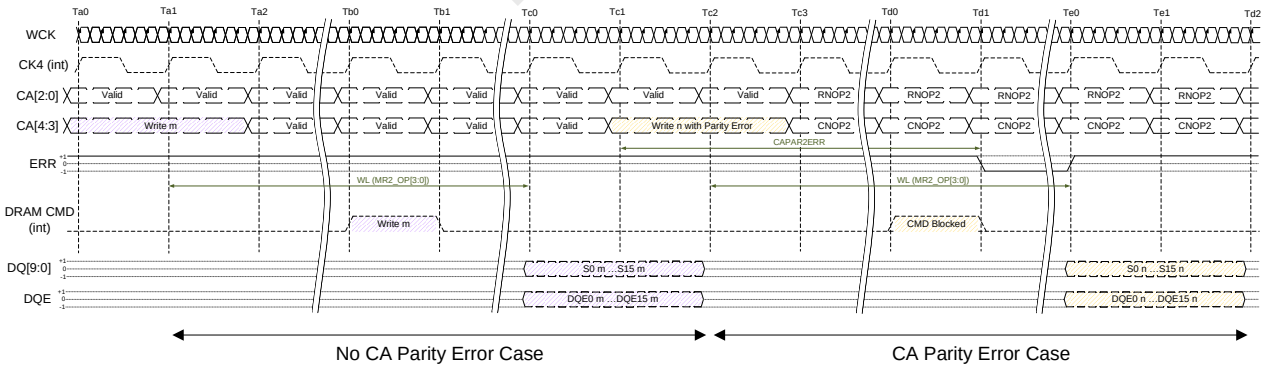


Figure 137 — Implicit CAPARBLK_LAT

FIGURE 138 illustrates the case where the device supports explicit CA parity blocking latency or explicit is programmed into the CAPARBLK_LAT control register (MR15 OP6 = 1_B) when both supported. In the case of explicit CAPARBLK_LAT the device specification will include *TABLE 143* for the timings that may have different values depending on whether command blocking is enabled or disabled.

If the device supports both, the device specification will include both tables and the host must use the corresponding table based on how the CAPARBLK_LAT control register (MR15 OP6) is programmed.

7.5 CAPAR with Command Blocking (CAPARBLK) (cont'd)

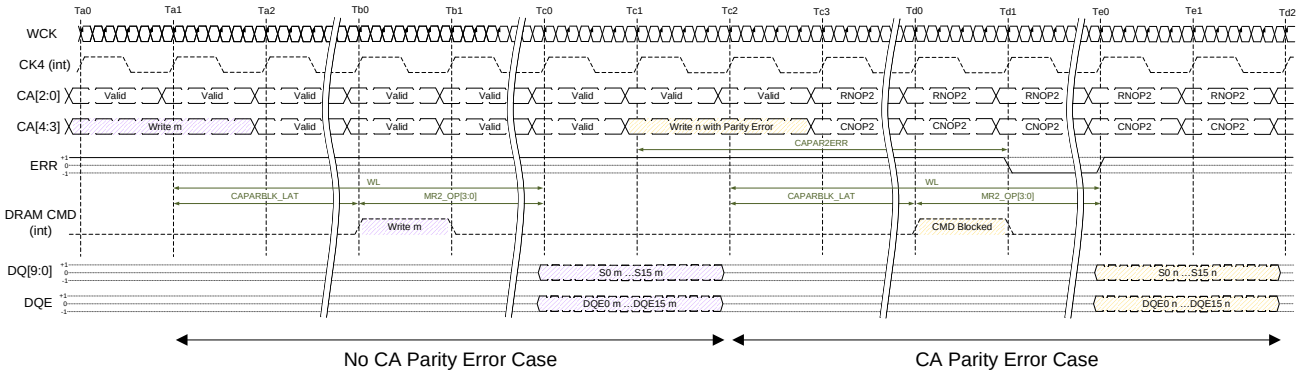


Figure 138 — Explicit CAPARBLK_LAT

Table 114 — Command Blocking Impacted AC Timings

Parameter		Symbol	Min	Max	Units	Notes
Read latency	Command blocking disabled	RL			nCK4	1
	Command blocking enabled				nCK4	1
Write latency	Command blocking disabled	WL			nCK4	1
	Command blocking enabled				nCK4	1
NOP commands required upon Sleep mode entry	CAPAR disabled or CAPAR enabled with CAPAR2ERR (MR15 OP[11:8] = 1-15	tCPDED	MAX (10, CAPAR2ERR +2)		nCK4	2
	CAPAR enabled with CAPAR2ERR (MR15 OP[11:8] = 0		MAX (10, RU{tWCK2ERRINT/tCK4} +2)		nCK4	
RCK Enable time	Command blocking disabled	RCKEN			nCK4	1
	Command blocking enabled				nCK4	1
RCKSTOP command Latency	Command blocking disabled	RCKSTOP_LAT			nCK4	1
	Command blocking enabled				nCK4	1

NOTE 1 See the *AC TIMINGS* section for min/max values.

NOTE 2 With CAPAR disabled, CAPAR2ERR is assumed to be 0 nCK4 in the equation MAX (10, CAPAR2ERR +2), resulting in 10 nCK4.

7.5 CAPAR with Command Blocking (CAPARBLK) (cont'd)

Once a Command Address Parity Error is detected the following actions will happen when CAPAR with Command Blocking is enabled as illustrated in *FIGURE 140*.

1) The command with the parity error is not executed, nor is the command on the other command bus as the CAPAR bit covers both command buses.

2) The commands in the tCAPAR_UNKNOWN and tCAPAR_UNKNOWN_WR periods on both the row and column bus prior to the erroneous command, are not guaranteed to be executed. tCAPAR_UNKNOWN and tCAPAR_UNKNOWN_WR are vendor specific. Consult vendor data sheets for the supported value.

3) The DRAM will block all subsequent commands on both the row and column bus until CA training with Self Refresh is automatically entered.

4) The DRAM will signal the host of the CAPAR error by driving the ERR signal to PAM3 level "-1" for 1 nCK4 cycle according to the value programmed into CAPAR2ERR (MR15 OP[11:8]) in both NRZ and PAM3 modes. See the *ERR SIGNAL* section for more details on ERR signal latency and timings.

5) The DRAM will complete any operation in progress and close any open pages before automatically entering CA Training with Self Refresh. Upon entering CA training with Self Refresh the bank counter for REFpb will be reset by the DRAM.

6) CA training with Self Refresh mode is automatically entered after tCAPAR_UNLOCK. The value of tCAPAR_UNLOCK is defined to allow the DRAM to complete any operation that is underway. When DRFM is enabled using MR8 OP2 = 1_B, the DRAM will wait RU (tDRFM / tCK4) + RU (tRP / tCK4) cycles before entering CA Training with Self Refresh mode. tDRFM depends on the programmed Bounded Refresh Configuration (BRC) in MR8 OP[4:3]. When DRFM is disabled using MR8 OP2 = 0_B, the DRAM will wait RU (tRFCab / tCK4) + RU (tRP / tCK4) cycles before entering CA Training with Self Refresh mode.

Table 115 — tCAPAR_UNLOCK

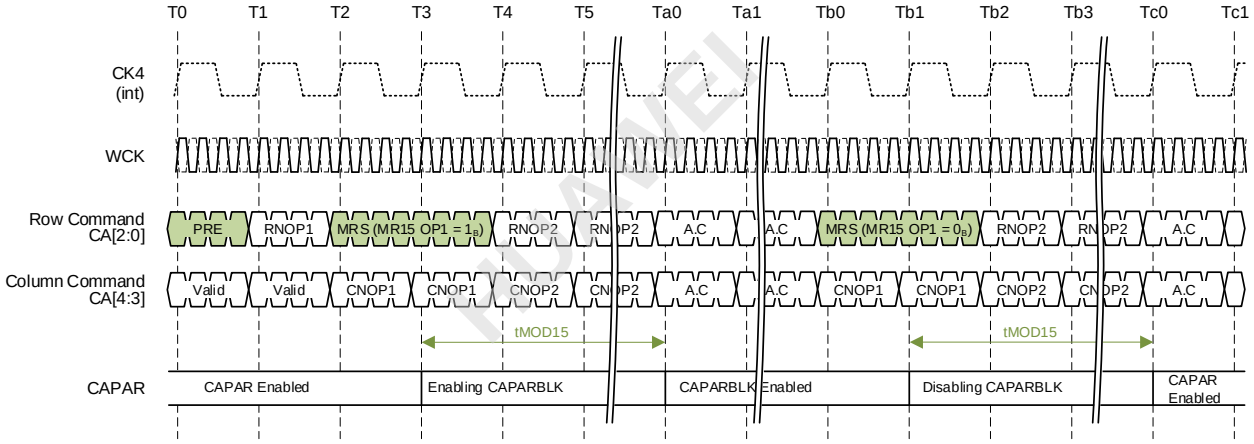
Parameter		Value		Unit
		Min	Max	
tCAPAR_UNLOCK	DRFM enabled	-	RU (tDRFM / tCK4) + RU (tRP / tCK4)	nCK4
	DRFM disabled	-	RU (tRFCab / tCK4) + RU (tRP) / tCK4)	nCK4

7) The DRAM may continue to check CAPAR during the tCAPAR_UNLOCK period before CA Training is entered. Any subsequent CAPAR error(s) may be reported, but the DRAM is not required to report subsequent CAPAR errors throughout the tCAPAR_UNLOCK period. The DRAM must signal any WRCRC errors during the tCAPAR_UNLOCK period from any valid Writes that were in progress prior to the CAPAR error as shown in *Figure 141*.

7.5 CAPAR with Command Blocking (CAPARBLK) (cont'd)

8) If RCK is running either with the always on mode or one of the RCK Start/Stop modes, then the RCK will asynchronously transition from toggling at WCK rate to toggling at CK4 rate no later than tCATERCK. The asynchronous transition may include stopping the RCK. First nCK4 pulse may be incomplete. The RCK may also stop during tCAPAR_UNLOCK if the Valid command before the tCAPAR_UNKNOWN is a RCKSTOP command when in one of the RCK Start/Stop modes. In this case the RCK will stop and transition to High-Z per the RCKSTOP procedure as outlined in the *READ CLOCK (RCK)* section before transitioning to CK4 rate no later than tCATE2RCK.

9) Once in CA Training with Self Refresh mode, the host has the option to performing CA training or exit using the CATX command. The host is required to follow the standard procedure for exiting CA Training with Self Refresh mode to go back to the Self Refresh state. The DRAM will also exit the CA Training to go back to Self Refresh in the standard procedure including starting RCK if the always on mode is programmed. See the *COMMAND ADDRESS BUS TRAINING* section for more details.

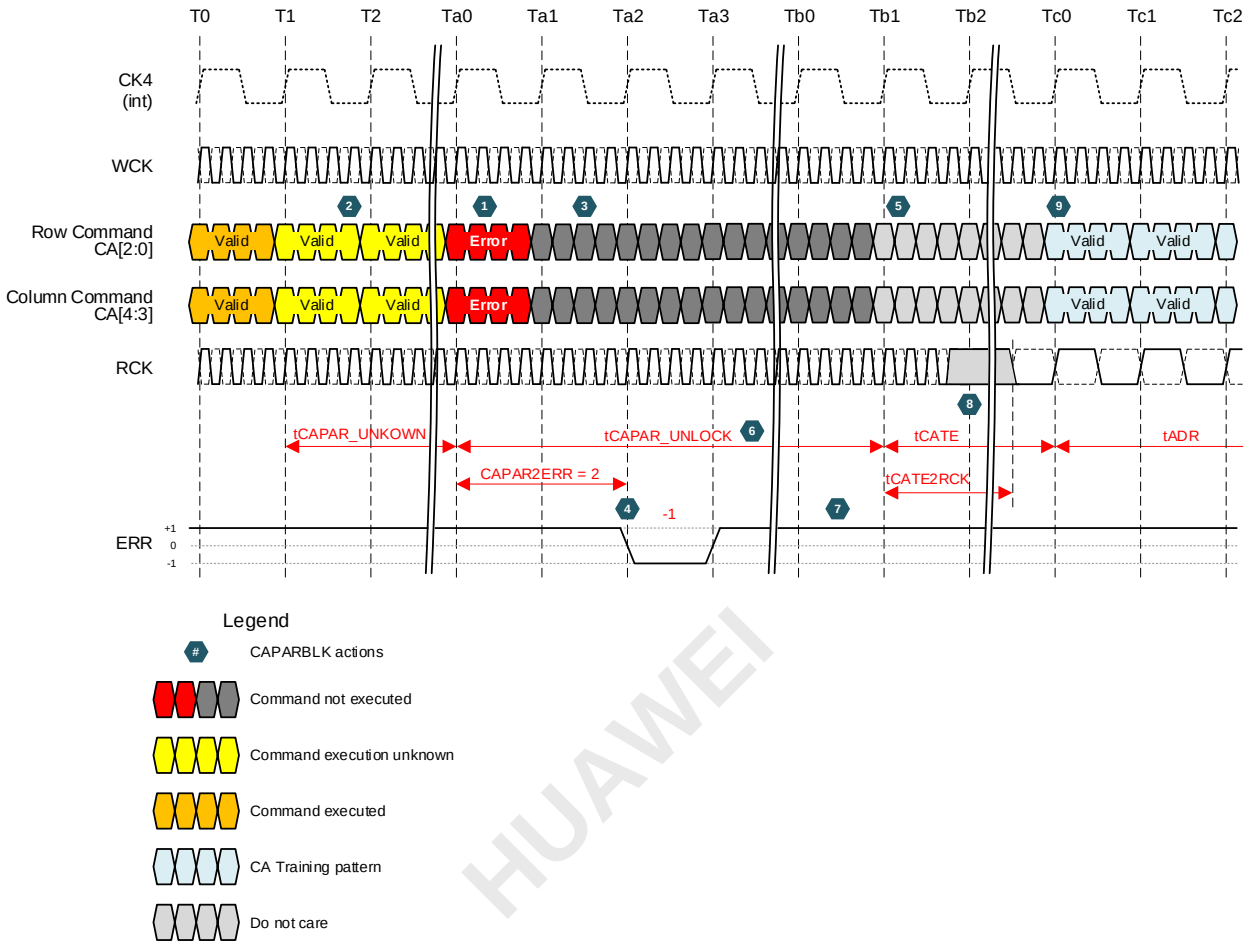


NOTES:

1. A Precharge command as shown in the figure shall not be issued when the channel is in Self Refresh state.
2. A.C. = Any command allowed in bank idle or Self Refresh state.
3. Any changes to CAPAR related registers in MR15 when FD_FLAG = 1_B require RNOP2 and CNOP2 for tMOD15.
4. Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 139 — Enabling and Disabling Command Address Parity Command Blocking (CAPARBLK) with FD_FLAG = 1_B

7.5 CAPAR with Command Blocking (CAPARBLK) (cont'd)

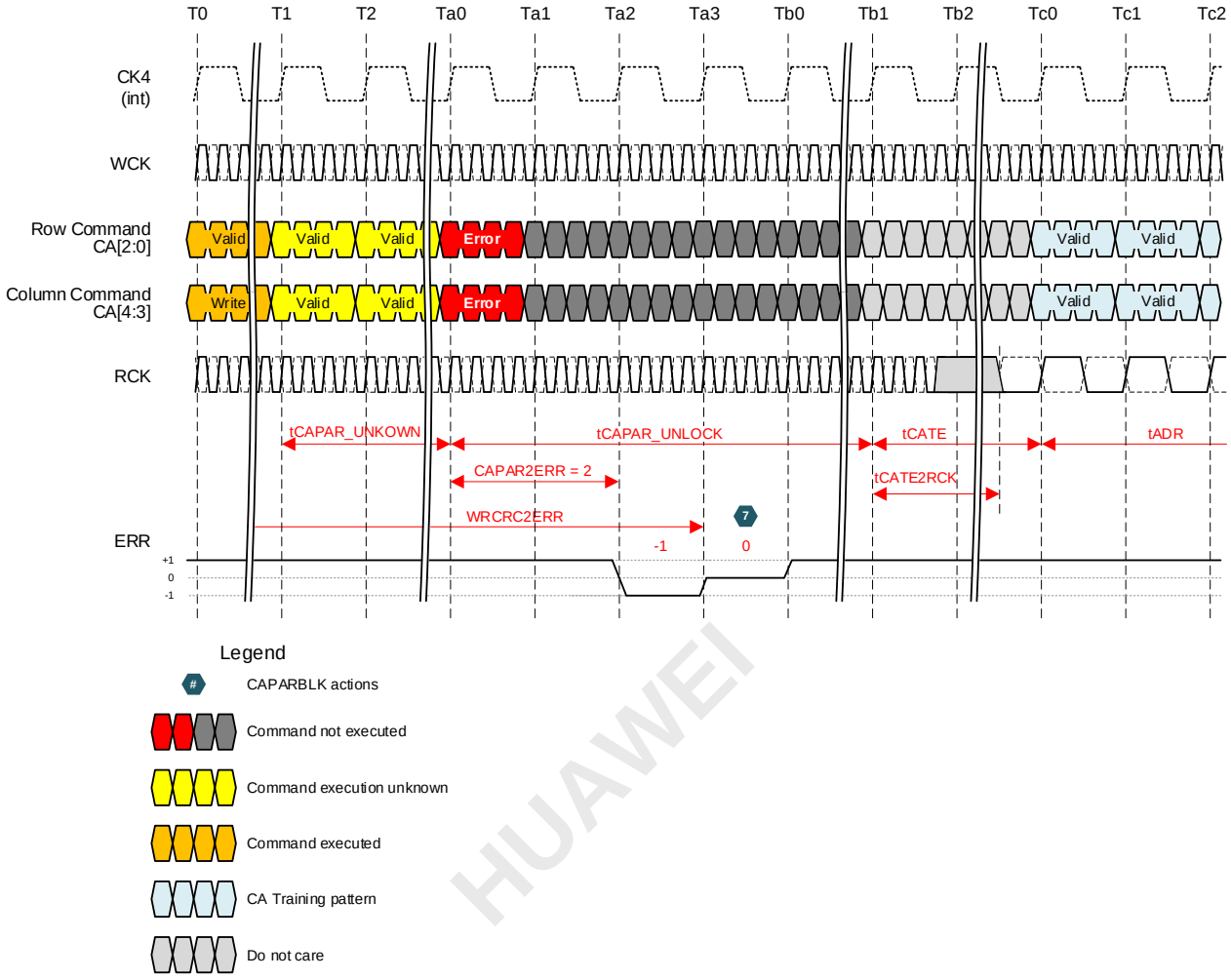


NOTES:

1. CAPAR2ERR = 2 is shown as an example for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. tWCK2CA and tWCK2ERRO = 0 are shown for illustration purposes.
4. RCK is shown toggling at WCK rate before the CAPAR Error at Ta0. Either during tCAPAR_UNLOCK or tCATE2RCK the DRAM will asynchronously transition from WCK rate to toggling at CK4 rate. First nCK4 pulse may be incomplete.
5. If the Valid command at or before T0 is a RCKSTOP command then the RCK will stop and transition to High-Z per the RCKSTOP procedure as outlined in the *READ CLOCK (RCK)* section before transitioning to CK4 rate no later than tCATE2RCK.
6. tCAPAR_UNKNOWN is shown for illustration purposes. tCAPAR_UNKNOWN_WR is the period where WR and WRA commands are not guaranteed to be executed and tCAPAR_UNKNOWN is for all other row and column commands.

Figure 140 — Command Blocking

7.5 CAPAR with Command Blocking (CAPARBLK) (cont'd)



NOTES:

1. CAPAR2ERR = 2 is shown as an example for illustration purposes. Actual supported values are found in the *MODE REGISTERS* and *AC TIMINGS* sections.
2. CA Parity Error Latency = CAPAR2ERR * tCK4 + tWCK2ERRO.
3. tWCK2CA and tWCK2ERRO = 0 are shown for illustration purposes.
4. RCK is shown toggling at WCK rate before the CAPAR Error at Ta0. Either during tCAPAR_UNLOCK or tCATE2RCK the DRAM will asynchronously transition from WCK rate to toggling at CK4 rate. First nCK4 pulse may be incomplete.
5. If there is a Valid command before the Write at T0 that is a RCKSTOP command then the RCK will stop and transition to High-Z per the RCKSTOP procedure as outlined in the *READ CLOCK (RCK)* section before transitioning to CK4 rate no later than tCATE2RCK.
6. tCAPAR_UNKNOWN is shown for illustration purposes. tCAPAR_UNKNOWN_WR is the period where WR and WRA commands are not guaranteed to be executed and tCAPAR_UNKNOWN is for all other row and column commands.

Figure 141 — Command Blocking (WRCRC Error after CAPAR Error)

7.6 CSP Feedback

GDDR7 DRAM supports CSP feedback feature, as an optional feature, to acknowledge the CSP command after exit from sleep mode, self refresh sleep mode or CA bus training. Voltage or temperature change during sleep or self refresh sleep mode may cause shift in WCK clock tree delay relative to CA, resulting in failure of CSP command acknowledgement. Error in CA training may also cause failure in CSP acknowledgement. Once GDDR7 DRAM fails to capture the CSP command, GDDR7 DRAM is unable to capture subsequent commands and may fall into an irrecoverable state. CSP feedback, an optional feature, explicitly informs the host whether the CSP was properly detected, and it is safe to continue with normal operation. CSP feedback allows the host to retry the CSP command if CSP is sent and not detected by GDDR7 DRAM. In this case, the host may also choose to initiate CA training by sending intentional CA parity error pattern if CAPARBLK (MR15_OP1) is enabled.

The CSP feedback is transferred to host by ERR signal. If the optional CSP feedback feature is supported and enabled (MR15 OP2 = 1_B), the ERR signal will transition to “0” level after tCSP_ACK_PRE and will transition to “+1” level after tCSP_ACK_POST if CSP is decoded successfully. If CSP feedback is not enabled (MR15 OP2 = 0_B), the ERR signal continues to be driven at the “+1” level. *FIGURE 142* shows CA training exit without CSP error case when CSP feedback feature is enabled. *FIGURE 143* shows sleep mode exit without CSP error case when CSP feedback feature is enabled.

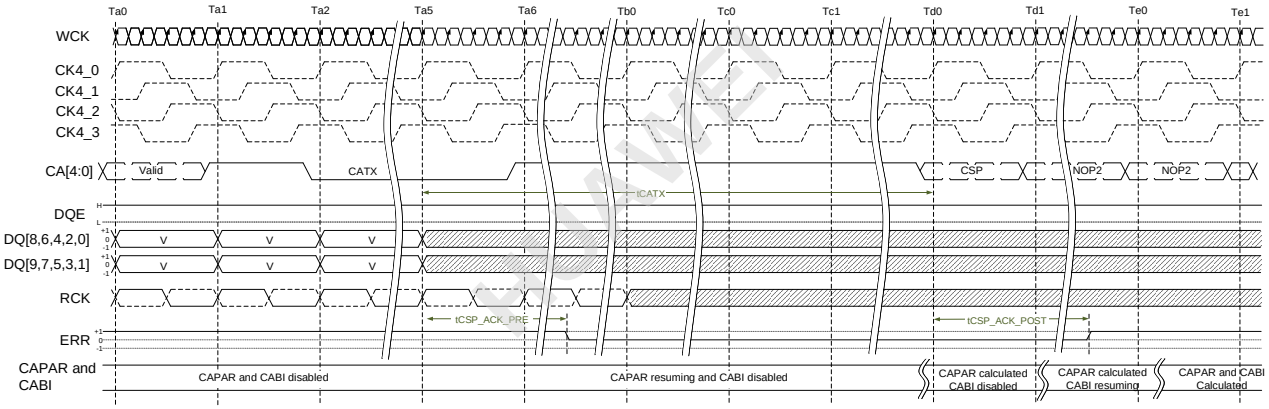


Figure 142 — CA Training Exit without CSP Error

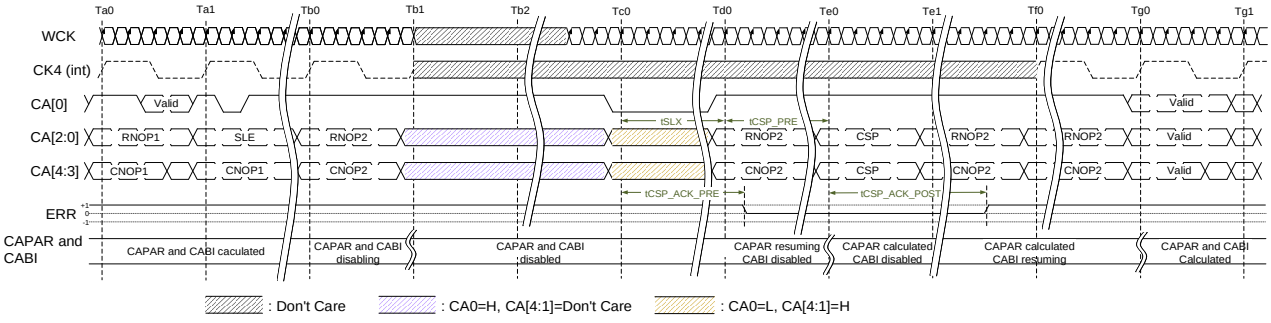


Figure 143 — Sleep Mode Entry and Exit without CSP Error

7.6 CSP Feedback (cont'd)

If a CA parity error occurs while the host is transmitting the CSP command when CAPAR enabled (MR15 OP0=1_B), the output of the ERR signal will transition to "-1" after CAPAR2ERR regardless of CSP feedback support. The CA parity error has higher priority over CSP feedback information. In *FIGURE 144*, CA parity error occurs with CSP command, making the ERR signal transition to "-1". If optional CSP feedback is supported and enabled, the CSP command with CA parity error is not acknowledged as a CSP command and host is allowed to retry CSP command on condition that CAPARBLK (MR15 OP1) = 0_B. If CAPARBLK (MR15 OP1) = 1_B, GDDR7 DRAM will enter CA training automatically.

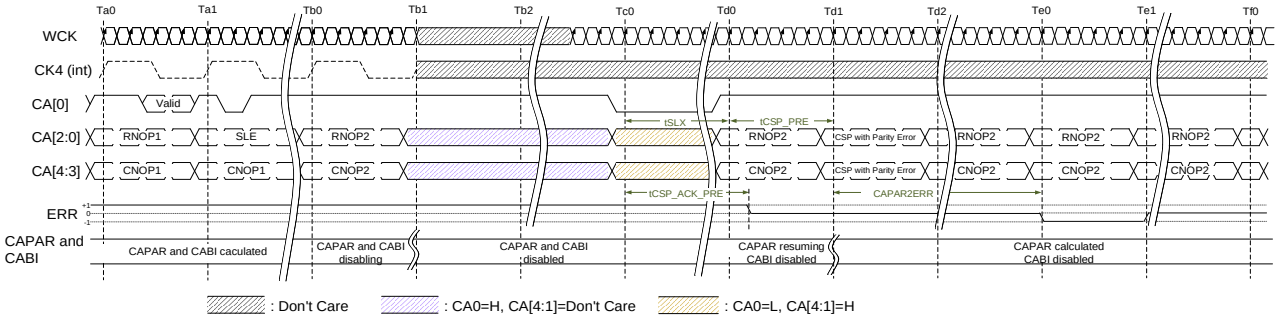


Figure 144 — Sleep Mode Entry and Exit with CSP Parity Error

If a CA parity error occurs within 1CK4 before CSP command when CSP feedback is enabled, the CSP command may not be recognized by GDDR7 DRAM. *FIGURE 145* shows this case and the ERR signal will transition to 0 after -1 of CA parity error report at Te2. If CAPARBLK is enabled, GDDR7 DRAM will enter CA training automatically, otherwise host may retry CSP command until DRAM recognizes CSP command to set ERR signal to +1.

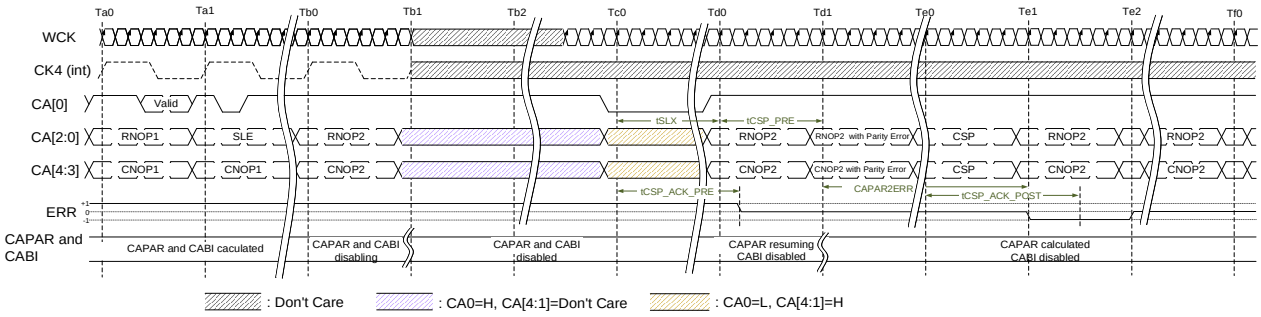


Figure 145 — Sleep Mode Entry and Exit with CSP Parity Error within 1CK4 before CSP Command

FIGURE 146 shows the case that first CSP command after sleep mode exit was not detected by DRAM without creating CA parity error when CSP feedback is enabled. ERR signal was kept in "0" state after the first CSP command from host because DRAM failed to capture the first CSP command. In this case, host retried CSP command and the second CSP command was successfully captured by GDDR7 DRAM and ERR signal returned to "+1" state after tCSP_ACK_POST elapsed from the second CSP command. To retry CSP command, both CAPAR2ERR+1tCK4 and tCSP_ACK_POST must be satisfied because there may exist CSP command fail or CA parity error. CSP command retry is allowed only when CAPARBLK is disabled.

7.6 CSP Feedback (cont'd)

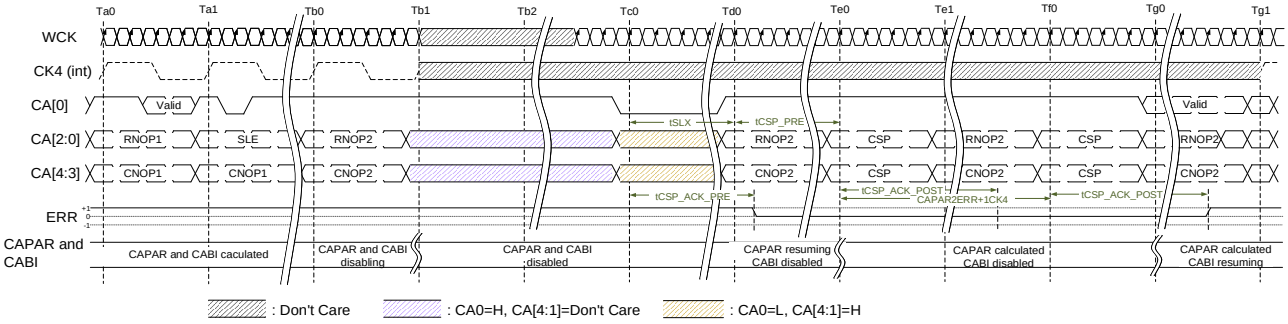


Figure 146 — Sleep Mode Entry and Exit with CSP Retry Pass

FIGURE 147 shows the case when there is a CA parity error in CSP command when both CSP feedback and CAPARBLK are enabled. GDDR7 DRAM will enter CA training automatically after tCAPAR_UNLOCK. ERR pin will also return to +1, once DRAM enters CA training.

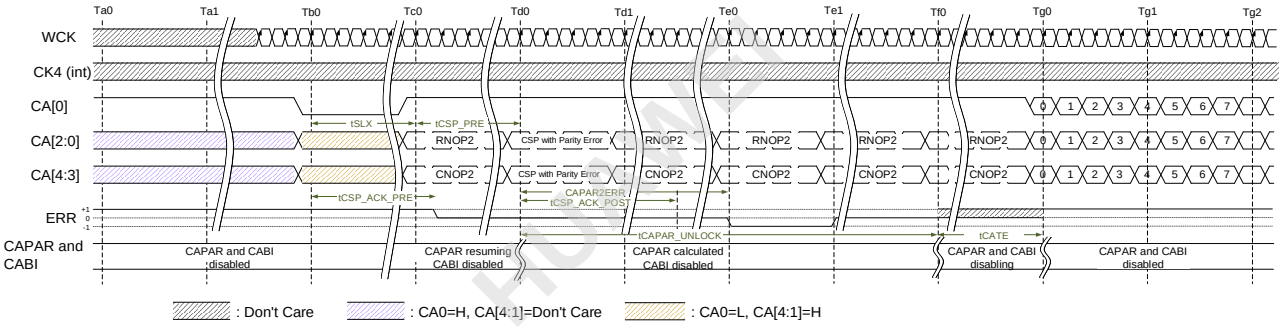


Figure 147 — Sleep Mode Exit with CSP Command with Parity Error when CAPARBLK is Enabled

GDDR7 DRAM will indicate the support of CSP feedback in Info Read register 3 bit 1 (see *TABLE 80*). To enable CSP feedback, CAPAR must also be enabled (MR15 OP0=1_B) before next sleep, self refresh sleep or CA training entry when GDDR7 DRAM supports optional CSP feedback feature.

Table 116 — CSP Feedback Support Definition in Info Read Address 3

IRA	Bit	Field	Description
3	DQ1	CSP feedback	CSP Feedback 0_B : CSP feedback is supported 1_B : CSP feedback is not supported

7.6 CSP Feedback (cont'd)

Table 117 — Mode Register Definition for CSP Feedback

Mode Register	Operand	Data	Notes
MR15	OP2	CSP Feedback (Optional) 0_B : Disable (default) 1_B : Enable	1, 2

NOTE 1 Enabling CSP feedback is allowed only when IRA3 DQ1 indicates that CSP feedback optional feature is supported.
NOTE 2 To enable CSP feedback feature, CAPAR must also be enabled (MR15 OP0=1_B) before next sleep, self refresh sleep or CA training entry.

The ERR signal status after CSP command is listed in *TABLE 118*. In the ERR signal, CA parity error information higher priority than CSP feedback. Therefore, ERR pin will transition to -1, if GDDR7 DRAM needs to send CA parity error and CSP failure simultaneously.

Table 118 — ERR Pin Status after CSP

Host (User) Setting		ERR Sig after CSP	
MR15 OP2 (CSP Feedback)	MR15 OP0 (CA Parity)	CSP Pass	CSP Error
0_B (Disable)	0_B (Disable)	+1	+1
0_B (Disable)	1_B (Enable)	+1	-1 (CAPAR Error)
1_B (Enable)	0_B (Disable)	+1	+1 [1]
1_B (Enable)	1_B (Enable)	+1	0 (CSP failure) -1 (CAPAR error)

NOTE 1 In normal operation CA Parity must be enabled with CSP feedback being enabled, therefore the CSP error will not be signaled on ERR.

The CSP feedback feature allows host to confirm the operation of only CA[1] and CA[3] because CSP command has toggle only in those signals. If host needs sanity check of CA[0],CA[2] and CA[4] signals, it is recommended to use CA link stress test pattern illustrated in *TABLE 119*. Host is allowed to send CA link stress test pattern before CSP command, once tCATX or tCSP_PRE has elapsed to allow GDDR7 DRAM to enable CA parity calculation circuits. To prevent GDDR7 DRAM from unintentionally recognizing CSP command during CA link stress pattern, host must avoid CA1=CA3=HHHL, HHLH, HLHH, LHHH with CA0=CA2=CA4=HHHH. Moreover, the CA stress pattern must be rolling window CA parity compliant because DRAM internal CK4 is not defined before recognition of CSP command. Rolling window means that calculation is performed for every 4 WCK cycles that starts from all WCK cycles, WCK0, WCK1, WCK2 and WCK3. Only RNOP2+CNOP2 commands are allowed during tPRECSP_NOP2 in *TABLE 119*.

If CA parity calculation is enabled and a CA parity error occurs during CA link stress test pattern, ERR signal will transition to -1. This CA bus sanity check with CA link stress pattern works with all GDDR7 DRAM's regardless of CSP feedback support.

7.6 CSP Feedback (cont'd)

A GDDR7 DRAM may have four CA sampling circuits utilizing rising edge of WCK0, WCK1, WCK2 and WCK3 clock cycles. Therefore, WCK cycle based pattern shift as shown in *TABLE 119* (b) may help to check the status of all four samplers. Instead of WCK cycle based shifting, the host may extend the length of CA link stress pattern as shown in *TABLE 119* (c) to check all CA samplers in GDDR7 DRAM.

Table 119 — Reference Example of CA Link Stress Test Pattern

Reference Example	WCK t0	WCK t1	WCK t2	WCK t3	WCK t4	WCK t5	WCK t6	WCK t7	WCK t8	WCK t9	WCK t10	WCK t11	WCK t12	WCK t13	WCK t14	WCK t15	WCK t16	WCK t17	WCK t18	WCK t19	WCK t20	WCK t21	WCK t22	WCK t23	WCK t24	WCK t25	WCK t26	WCK t27	WCK t28	WCK t29	WCK t30	WCK t31	WCK t32	WCK t33	WCK t34	WCK t35
CA 0	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	L	H	H	L	L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H
CA 1	H	H	H	H	H	H	H	H	H	H	H	L	H	L	H	L	L	L	L	H	L	H	H	H	L	H	L	H	H	H	H	L	H	H	H	H
CA 2	H	H	H	H	H	H	H	H	H	H	H	H	L	H	L	H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
CA 3	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	L	H	H	H	L	L	L	H	L	H	L	H	H	H	H	H	L	H	H	H	H
CA 4	H	H	H	H	H	H	H	H	H	H	H	L	L	L	L	H	L	H	H	H	H	H	H	H	L	H	L	H	H	H	H	H	H	H	H	H
	tCATX or tCSP_PRE			CA link stress test pattern																								tPRECSP_NOP2				CSP Command				
	CAPAR parity check enabled																																			

(a) Reference example of CA link stress test pattern.
Single high pulse after multi-cycle low or single low pulse after multi-cycle high are marked in purple color.

	WCK t0	WCK t1	WCK t2	WCK t3	WCK t4	WCK t5	WCK t6	WCK t7	WCK t8	WCK t9	WCK t10	WCK t11	WCK t12	WCK t13	WCK t14	WCK t15	WCK t16	WCK t17	WCK t18	WCK t19	WCK t20	WCK t21	WCK t22	WCK t23	WCK t24	WCK t25	WCK t26	WCK t27	WCK t28	WCK t29	WCK t30	WCK t31	WCK t32	WCK t33	WCK t34	WCK t35
CA 0	H	H	H	H	H	H	L	H	H	H	H	L	H	H	L	L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
CA 1	H	H	H	H	H	H	L	H	L	H	L	L	L	L	H	L	H	H	H	L	H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H
CA 2	H	H	H	H	H	H	H	L	H	L	H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
CA 3	H	H	H	H	H	H	L	H	H	H	L	H	H	H	L	L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
CA 4	H	H	H	H	H	H	L	L	L	L	H	L	H	H	H	H	H	H	H	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	tCATX or tCSP_PRE			CA link stress test pattern																								tPRECSP_NOP2				CSP Command				
	CAPAR parity check enabled																																			

(b) CA link stress test pattern with WCK clock cycle based shift
Host is allowed to test healthiness of WCK0/90/180/270 sampling timing by WCK based pattern shift.

	WCK t0	WCK t1	WCK t2	WCK t3	WCK t4	WCK t5	WCK t6	WCK t7	WCK t8	WCK t9	WCK t10	WCK t11	WCK t12	WCK t13	WCK t14	WCK t15	WCK t16	WCK t17	WCK t18	WCK t19	WCK t20	WCK t21	WCK t22	WCK t23	WCK t24	WCK t25	WCK t26	WCK t27	WCK t28	WCK t29	WCK t30	WCK t31	WCK t32	WCK t33	WCK t34	WCK t35
CA 0	H	H	H	L	H	H	H	L	H	L	H	H	H	L	H	L	H	H	H	L	H	L	H	H	L	L	L	H	H	H	H	H	H	H	H	H
CA 1	H	H	H	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	H	H	H	H	H	H	H	L	H	H	H	H
CA 2	H	H	H	H	H	L	H	L	H	H	H	L	H	L	H	H	H	L	H	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
CA 3	H	H	H	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	H	H	H	H	H	H	H	L	H	H	H	H
CA 4	H	H	H	L	H	L	H	H	H	L	H	H	H	H	H	L	H	L	H	H	H	L	H	L	L	L	L	H	H	H	H	H	H	H	H	H
	tCATX or tCSP_PRE			CA link stress test pattern																								tPRECSP_NOP2				CSP Command				
	CAPAR parity check enabled																																			

(c) Reference example of CA link stress pattern with extended length

Before successful CSP command recognition, the internal CK4 inside GDDR7 DRAM is not defined. Therefore, there are four different cases that the DRAM internal CK4 is temporally aligned with WCK of host. Host is allowed to issue CSP command only after tPRECSP_NOP2. GDDR7 DRAM may not recognize the CSP command for a period same or shorter than tCAPAR2CSP after the CA parity error is found. The CA parity error information is returned to the host after CAPAR2ERR. If CA parity error is found in the CA link stress pattern when CAPARBLK is enabled, GDDR7 DRAM will enter CA training automatically. See the CA parity section for more details.

7.6 CSP Feedback (cont'd)

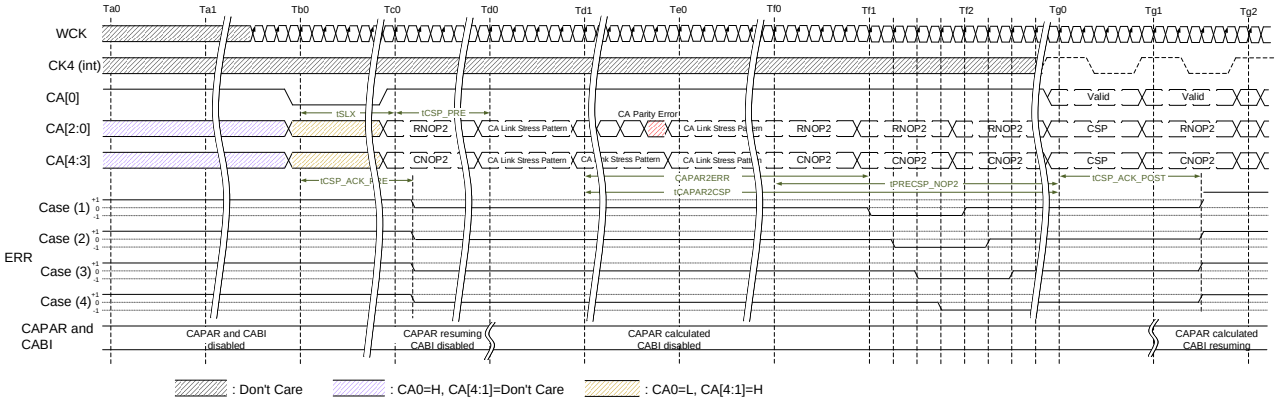


Figure 148 — Sleep Mode Exit with CA Link Stress Pattern

An intentional CA parity error pattern is defined in *TABLE 120*, as a robust method to force the GDDR7 DRAM to enter CA training when CAPAR_BLK is enabled and the GDDR7 DRAM failed to acknowledge a CSP command without a CA parity error. A GDDR7 DRAM must be able to regard this intentional CA parity error pattern as CA parity error regardless of CA timing alignment with WCK if all AC timings for a CSP command (i.e., tCATE or tSLX+tCSP_PRE) are met and GDDR7 DRAM not yet detected CSP command.

Table 120 — Intentional Parity Error Pattern

Function	Symbol	WCK Clock Cycle	CA0	CA1	CA2	CA3	CA4	Notes
Intentional CA Parity Error	ICAPAR_ERR	n	H	H	H	L [2]	H	1
		n+1	H	H	H	L [2]	H	
		n+2	H	H	H	L [2]	H	
		n+3	H	H	H	H	H	

NOTE 1 GDDR7 DRAM with optional CSP feedback support (IRA3 DQ1 = 1_B) will recognize ICAPAR_ERR pattern as CA parity error when all AC timings required CSP commands are met, and DRAM is waiting for CSP command. If CAPAR(MR15 OP0)=1_B, DRAM will output CA Parity Error to ERR signal ("-1"). If CAPARBLK(MR15_OP1) =1_B GDDR7 DRAM will automatically enter CA training, because ICAPAR_ERR pattern is recognized as a CA parity error.

NOTE 2 Three consecutive L on cycles n, n+1, and n+2 are shown on CA3 for the intentional parity error. It is pointed out that three cycles of L on any CA signal provides a robust method for the GDDR7 DRAM to enter CA training. Other methods of generating CA parity error are also legal but may not provide a robust method of entering CA training.

7.6 CSP Feedback (cont'd)

Table 121 — AC Parameters in CSP Feedback

Parameter	Symbol	Min	Max	Unit	Notes
CATX to CSP command delay	tCATX		-	ns	
Sleep mode exit to CSP command delay	tCSP_PRE		-	ns	1
CA0 low pulse width to exit from sleep mode	tSLX		-	ns	
Waiting time from CATX or SLX with SLX2CAT=1_B or Self refresh sleep mode exit with SRSLX2CAT=1_B to ERR = "0"	tCSP_ACK_PRE		-	ns	2
Waiting time from Successful CSP to ERR="+1"	tCSP_ACK_POST		-	ns	3
Minimum time for host to wait until sending CSP command after CA link stress pattern	tPRECSP_NOP2		-	CK4	
Minimum time for host to wait to send a CSP command that DRAM guarantees recognition after CA parity error	tCAPAR2CSP	CAPAR2ERR + 2		CK4	4

NOTE 1 tCSP_PRE is frequency dependent. See *TABLE 145* for more details.
NOTE 2 The timing means minimum time host to wait before observing ERR="0".
NOTE 3 The timing means minimum time host to wait before observing ERR="+1".
NOTE 4 This timing starts from occurrence of any parity error prior to CSP after tCATX or tCSP_PRE has expired.

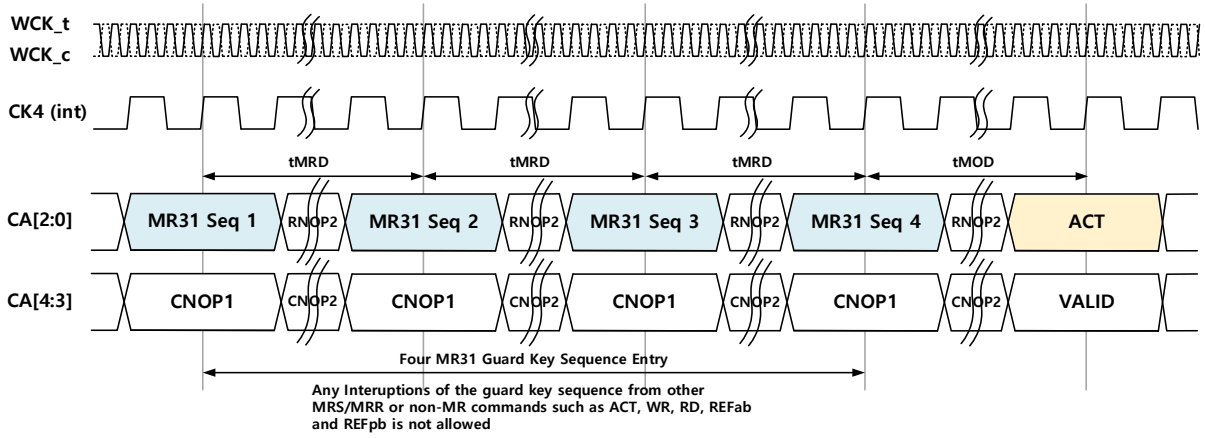
7.7 Hard Post Package Repair (hPPR)

GDDR7 supports hard Fail Row address repair, hPPR which allows a simple and easy repair method in a system. Entry into hPPR is protected through a sequential MRS guard key to prevent unintentional hPPR programming. The hPPR guard key requires a sequence of four MRS commands as shown in *FIGURE 149*. The guard key sequence must be entered in the specified order as shown in the figure and in *TABLE 122*. If the guard key is not entered in the required order or an incorrect guard key is entered unintentionally, the SGRAM will capture the wrong guard key and the device will not enter hPPR mode. Only RNOP2/CNOP2 commands are allowed between the four MRS commands.

A guard key of more than four MRS commands to MR31 may be considered an incorrect guard key even if the greater than four MRS commands includes the MR31 Seq1 thru MR31 Seq4 in the proper order separated only by RNOP2/CNOP2 commands. The entry into hPPR mode in this case is vendor specific. It is noted that regardless of the vendor implementation, if the host is aware that sequence of MRS commands it started to enter is incorrect, the host can issue a MRS command to MR31 with OP11=0 to reset the hPRR mode entry sequence and re-issue the correct guard key sequence after tMOD expires as illustrated in *FIGURE 150*. For GDDR7 SGRAMs that treat the greater than four MRS commands as an incorrect guard key, this is the only case where more than four MRS commands to MR31 are allowed for successful entry into hPPR mode.

If a command other than RNOP2/CNOP2 is entered between MRS commands, the offending command will terminate the hPPR mode entry process and may or may not execute correctly however the offending command will not cause the SGRAM to lock up. The SGRAM does not provide an error indication if an incorrect hPPR guard key sequence is entered, nor if the hPPR mode entry sequence was terminated. The subsequent ACT, PRE and MRS to MR31 OP[11] associated with the hPPR operation will not cause the SGRAM to lock up.

Once hPPR mode is successfully entered and hPPR operation started, if the hPPR operation is prematurely terminated, whether unforeseen or not, the repair may or may not complete and the resource may or may not be preserved. The SGRAM does not inform the host if the repair was started successfully, if completed or unsuccessful/terminated. The only way to verify if the repair was successfully completed or not and resource(s) are still available is to perform a reset and read the repair resource as well as check the status of the failing address using reads and writes for each repair that the host attempted before the required reset.



NOTES:
1. Only NOP2 commands are allowed during tMRD and tMOD.
2. Prior to entering hPPR mode, RCK must be stopped.

Figure 149 — Guard Key Timing Diagram

7.7 Hard Post Package Repair (hPPR) (cont'd)

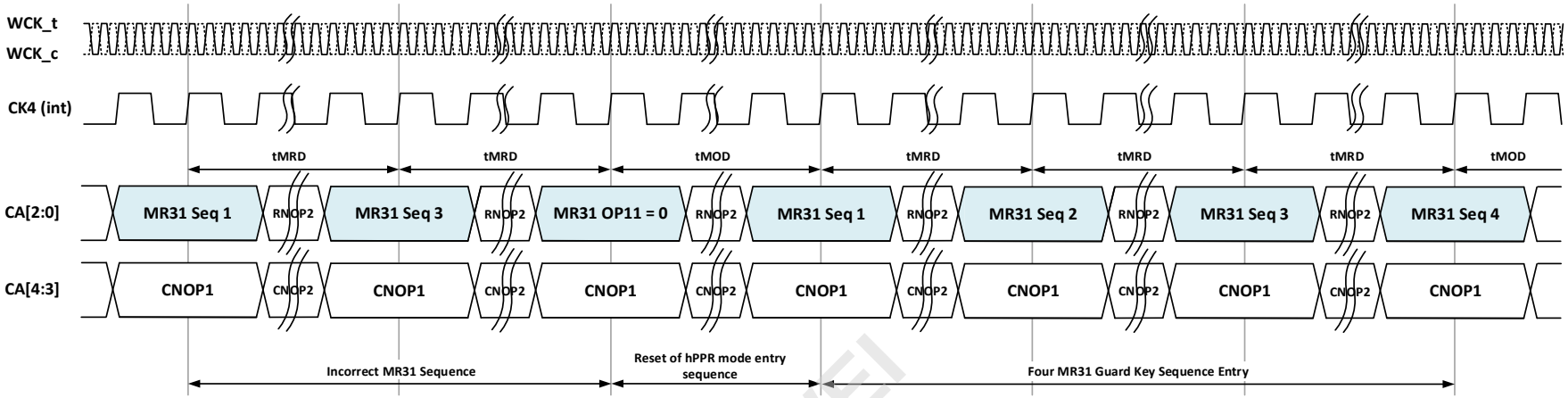


Figure 150 — Incorrect Guard Key Sequence and hPPR Mode Reset

7.7 Hard Post Package Repair (hPPR) (cont'd)

Table 122 — Guard Key Encoding for MR31

Command	OP[11]	OP[7]	OP[6]	OP[5]	OP[4]	OP[3]	OP[2]	OP[1]	OP[0]
MR31 Seq1	1	1	1	0	0	1	1	1	1
MR31 Seq2	1	0	1	1	1	0	0	1	1
MR31 Seq3	1	1	0	1	1	1	0	1	1
MR31 Seq4	1	0	0	1	1	1	0	1	1

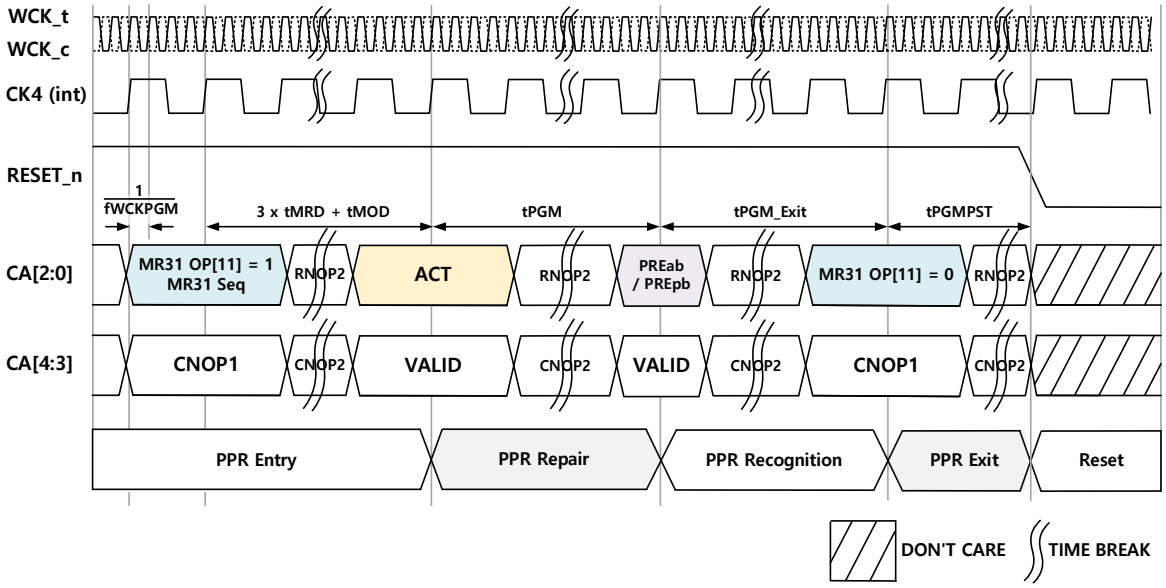
7.7.1 hPPR Fail Row Address Repair

With hPPR, GDDR7 SGRAMs can correct at least one row address per Bank. The hPPR resource designation (SGRAM Info address[25:26]) will indicate the hPPR resource availability and can be read/checked prior to implementing a repair. It is important to note that hPPR repairs are permanent; the Electrical-fuse cannot be switched back to un-fused states once it is programmed. The controller should prevent unintended hPPR mode entry and repair. (i.e., During the Command/Address training period). Entry into hPPR is through a register enable, ACT command is used to transmit the bank and row address of the row to be replaced in SGRAM. After program time, and PRE, the hPPR mode can be exited. During hPPR, other channels shall be in idle state and RD/WR commands are not allowed on any channel (i.e., All banks shall be Precharged and other channels are not allowed to be issued except for NOP2 command).

1. Since the SGRAM Info allows the user to execute hPPR resource, SGRAM Info of hPPR resource designation (field [25:26]) needs to be read. After user's checking the hPPR resource availability of each bank from SGRAM Info, hPPR mode can be entered. If the SGRAM Info of hPPR resource designation (field [25:26]) appears to not be available, the host controller should not issue hPPR mode.
2. Before entering hPPR mode, all banks are required to be Precharged and idle state on all channels, and RCK shall be stopped.
3. Issue the guard key as four consecutive MR31 MRS commands each with a unique address field OP[7:0] and OP[11]. Each MRS command should be spaced by tMRD and tMOD must be observed after the last guard key MRS command. Only RNOP2/CNOP2 commands are allowed in the tMRD period between MRS commands as well as during tMOD from the last MR31 of the guard key sequence to the ACT command as shown in *FIGURE 149*.
4. Issue an ACT command with the Bank and Row fail address.
5. Wait tPGM to allow the SGRAM repair target Row Address internally, then issue PREab/PREpb command.
6. Wait tPGM_Exit after PRE command which allows the SGRAM to recognize repaired Row address RAn.
7. Exit hPPR by setting MR31 OP[11]=0 and wait tPGMPST
8. In case of repairing one failed address on different banks and/or different channels subsequently, repeat 3 to 8.
9. Assert Reset_n, and then to do the reset and initialization procedure.

After entering hPPR mode, do not deviate from the above-mentioned procedure. The hPPR procedure is required to be performed at WCK clock frequency range from 200 MHz to 2000 MHz defined as fWCKPGM and the WCK period for hPPR mode is defined as 1/fWCKPGM as shown in *FIGURE 151*. Once hPPR mode is exited, the controller can verify that the target row was repaired by writing data into the target row and reading it back after reset and initialization procedure.

7.7 Hard Post Package Repair (hPPR) (cont'd)



NOTES:
1. With one hPPR command, only one row can be repaired at one time per die
2. RESET is required at the end of every hPPR procedure
3. During hPPR, memory contents are not refreshed and may be lost
4. Assert Reset_n: Refer to *INITIALIZATION* for details on reset, power-up, initialization and power off procedures
5. During hPPR, other channels are not allowed to enter hPPR; it shall be in idle state.
6. During hPPR, RD/WR commands are not allowed on any channels.
7. Prior to entering hPPR mode, RCK shall be stopped.

Figure 151 — hPPR Fail Row Repair Timing

Table 123 — Mode Register Bits for hPPR

Mode Register	OP[11]	OP[10]	OP[9]	OP[8]	OP[7]	OP[6]	OP[5]	OP[4]	OP[3]	OP[2]	OP[1]	OP[0]
MR31	hPPR	RFU	RFU	RFU	hPPR Guard Key							

7.7.2 hPPR Required Timing Parameters

Repair requires additional time to repair Fail Row Address into Spare Row Address and the following timing parameters are required for hPPR.

Table 124 — hPPR Timing Parameters

Parameter	Symbol	Min	Max	Unit	Notes
hPPR Programming Clock	fWCKPGM	200	2000	MHz	
hPPR Programming Time	tPGM	2000	-	ms	
hPPR Exit Recognition Time	tPGM_Exit	tRP	-	ns	
New Address Setting Time	tPGMPST	500	-	us	

8 Operating Conditions

8.1 Absolute Maximum Ratings

Stresses greater than those listed may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may adversely affect reliability.

Table 125 — Absolute Maximum Ratings

Parameter [1]	Symbol	Min	Max	Unit	Notes
Voltage on V_{DD} pin relative to V_{SS}	V_{DD}	-0.3	1.5	V	1, 2
Voltage on V_{DDQ} pin relative to V_{SS}	V_{DDQ}	-0.3	1.5	V	1, 2
Voltage on V_{PP} pin relative to V_{SS}	V_{PP}	-0.3	2.3	V	3
Voltage on any pins relative to V_{SS}	V_{IN}/V_{OUT}	-0.3	1.5	V	2
Storage temperature	T_{STG}	-55	+125	°C	4

NOTE 1 V_{DD} and V_{DDQ} must be within 300 mV of each other at all times the device is powered-up for devices that require $V_{DD} = V_{DDQ}$.
NOTE 2 Maximum values apply to GDDR7 SGRAMs supporting nominal V_{DD}, V_{DDQ} levels of 1.20V. Vendor datasheets should be consulted for devices only supporting V_{DD}, V_{DDQ} levels less than 1.20V as maximum values may be lower.
NOTE 3 VPP must be equal or greater than V_{DD} and V_{DDQ} at all times the device is powered-up.
NOTE 3 Storage temperature is the case surface temperature on the center/top side of the DRAM. For the measurement conditions, please refer to the JESD51-2 standard.

8.2 Pad Capacitances

Table 126 — Silicon Pad Capacitance

Parameter [1]	Symbol	Min	Max	Unit	Notes
Delta Input/Output Capacitance: DQ, DQE	DC_{IO}	–		pF	2
Delta Output Capacitance: RCK_t, RCK_c	DC_{O1}	–		pF	3
Delta Input Capacitance: CA	DC_{I1}	–		pF	4
Delta Input Capacitance: WCK_t, WCK_c	DC_{I2}	–		pF	5
Input/Output Capacitance: DQ, DQE	C_{IO}			pF	
Output Capacitance: RCK_t, RCK_c	C_{O1}			pF	
Output Capacitance: ERR	C_{O2}			pF	
Input Capacitance: CA	C_{I1}			pF	
Input Capacitance: WCK_t, WCK_c	C_{I2}			pF	

NOTE 1 Silicon pad capacitance values are not subject to production test. They are verified by design and characterization and validated by de-embedding the package L & C parasitics. The capacitance is measured with V_{DD}, V_{DDQ} and V_{SS} applied with all other signals floating.
NOTE 2 $DC_{IO} = C_{IO}$ (Max) – C_{IO} (Min)
NOTE 3 DC_{O1} = Absolute value of C RCK_t – C RCK_c
NOTE 4 $DC_{I1} = C_{I1}$ (Max) – C_{I1} (Min)
NOTE 5 DC_{I2} = Absolute value of C WCK_t – C WCK_c

8.3 Package Electrical Specification

Table 127 — Package Electrical Specification

Parameter [1, 2, 3]	Symbol	Min	Max	Unit	Notes
Input/Output package impedance: DQ, DQE	Z_{IO}			Ω	
Input/Output package delay: DQ, DQE	Td_{IO}			ps	
Output package impedance: RCK_t, RCK_c	Z_{O1}			Ω	
Output package delay: RCK_t, RCK_c	Td_{O1}			ps	
Output package impedance: ERR	Z_{O2}			Ω	
Output package delay: ERR	Td_{O2}			ps	
Input package impedance: CA	Z_{I1}			Ω	
Input package delay: CA	Td_{I1}			ps	
Input package impedance: WCK_t, WCK_c	Z_{I2}			Ω	
Input package delay: WCK_t, WCK_c	Td_{I2}			ps	
Delta output package impedance: RCK_t, RCK_c	DZ_{O1}	–		Ω	4
Delta output package delay: RCK_t, RCK_c	DTd_{O1}	–		ps	5
Delta input package impedance: WCK_t, WCK_c	DZ_{I2}	–		Ω	6
Delta input package delay: WCK_t, WCK_c	DTd_{I2}	–		ps	7

NOTE 1 This parameter is not subject to production test. It is verified by design and characterization. The package L & C parasitics are validated using package only samples. The capacitance is measured with V_{DD}, V_{DDQ} and V_{SS} shorted with all other signals floating. The inductance is measured with V_{DD}, V_{DDQ} and V_{SS} shorted and all other signal balls shorted at the die side (not ball).

NOTE 2 Package only impedance (Z_{PKG}) is calculated based on the L_{PKG} and C_{PKG} total for a given pin where

$$Z_{PKG}(\text{total per pin}) = \sqrt{L_{PKG}/C_{PKG}}$$

NOTE 3 Package only delay (Td_{PKG}) is calculated based on L_{PKG} and C_{PKG} total for a given pin where

$$Td_{PKG}(\text{total per pin}) = \sqrt{L_{PKG}*C_{PKG}}$$

NOTE 4 DZ_{O1} = Absolute value of Z_{O1} RCK_t – Z_{I2} RCK_c

NOTE 5 DTd_{O1} = Absolute value of Td_{O1} RCK_t – Td_{I2} RCK_c

NOTE 6 DZ_{I2} = Absolute value of Z_{I2} WCK_t – Z_{I2} WCK_c

NOTE 7 DTd_{I2} = Absolute value of Td_{I2} WCK_t – Td_{I2} WCK_c

8.4 Package Thermal Characteristics

Table 128 — Package Thermal Characteristics

Parameter [1]	Symbol	Value	Unit	Notes
Thermal resistance from junction to ambient	Theta_JA		°C/W	2, 3
Thermal resistance from junction to board	Theta_JB		°C/W	2, 4
Thermal resistance from junction to case	Theta_JC		°C/W	4
Junction-to-board thermal characterization parameter	Psi_JB		°C/W	5
Junction-to-top thermal characterization parameter	Psi_JT		°C/W	5

NOTE 1 Measurement procedures for each parameter must follow standard procedures defined in the JEDEC JESD51 standard.

NOTE 2 Theta_JA and Theta_JB must be measured with the high effective thermal conductivity test board defined in JESD51-7.

NOTE 3 Airflow information must be documented for Theta_JA.

NOTE 4 Theta_JB and Theta_JC are derived through a package thermal simulation.

NOTE 5 Psi_JB and Psi_JT parameters as defined in JESD51-12 are derived through a package thermal simulation.

8.5 Electrostatic Discharge Sensitivity Characteristics

Table 129 — Electrostatic Discharge Sensitivity Characteristics

Parameter [1]	Symbol	Min	Max	Unit	Notes
Human body model (HBM)	ESD_{HBM}	1000	–	V	2
Charged-device model (CDM)	ESD_{CDM}	250	–	V	3
NOTE 1 State-of-the-art basic ESD control measures must be in place when handling devices.					
NOTE 2 Refer to ESDA / JEDEC Joint Standard JS-001 for measurement procedures.					
NOTE 3 Refer to ESDA / JEDEC Joint Standard JS-002 for measurement procedures.					

8.6 Operating Temperature Range

GDDR7 operating temperature refers to the junction temperature of the GDDR7 SGRAM as being reported via the Temperature Sensor Readout of Info Read. The Info Read (IRD) command shall be issued in regular intervals on the 2 or 4 channels of the device, depending on the selected channel configuration, to verify that the maximum support operating temperature is not exceeded.

GDDR7 SGRAMs supports JESD402-1 operating temperature ranges. For specific ranges available and supported in a device, please refer to vendor datasheets and JESD402-1B or later. *TABLE 130* thru *TABLE 132* show examples of the ranges that could be supported in a device.

Table 130 — Operating Temperature Range Example 1

Parameter	Symbol	Min	Max	Unit	Notes
Operating Temperature Range	TJopr115	0	115	°C	1
NOTE 1 See the *AC TIMINGS* table for details on refresh rate.					

Table 131 — Operating Temperature Range Example 2

Parameter	Symbol	Min	Max	Unit	Notes
Operating Temperature Range Standard	TJopr90	0	90	°C	1
Operating Temperature Range Extended	TJopr100	0	100	°C	1
NOTE 1 See the *AC TIMINGS* table for details on refresh rate.					

Table 132 — Operating Temperature Range Example 3

Parameter	Symbol	Min	Max	Unit	Notes
Operating Temperature Range A105	TJoprA105	-40	105	°C	1
Operating Temperature Range A85	TJoprA85	-40	85	°C	1
Operating Temperature Range Extended	TJopr110	0	110	°C	1
Operating Temperature Range Standard	TJopr95	0	95	°C	1
NOTE 1 See the *AC TIMINGS* table for details on refresh rate.					

8.7 DC Operating Conditions

Initially, all GDDR7 SGRAMs are designed for 1.2 V typical V_{DD} and V_{DDQ} voltage supplies. GDDR7 SGRAMs can also be designed for 1.1 V typical V_{DD} and V_{DDQ} voltage supplies, or be designed to support multiple V_{DD} and V_{DDQ} supply voltages, e.g., 1.2 V and 1.1 V. Vendor datasheets should be consulted for actual V_{DD} and V_{DDQ} voltages supported, as factors such as process technology and supported system voltage(s) many require typical operating voltages to be added, dropped or maintained over time.

GDDR7 SGRAMs can also support the switching of the V_{DD} and V_{DDQ} supply voltages during sleep mode. The Dynamic Voltage Switching (DVS) procedure should be followed when switching between supply voltages. Vendor datasheets should be consulted for details regarding DVS support.

8.7 DC Operating Conditions (cont'd)

Table 133 — DC Operating Conditions

Parameter	Symbol	1.20 V Range			1.10 V Range			Unit	Notes
		Min	Typ	Max	Min	Typ	Max		
Device supply voltage	V_{DD}	1.164	1.2	1.236	1.067	1.1	1.133	V	1, 2
Output supply voltage	V_{DDQ}	1.164	1.2	1.236	1.067	1.1	1.133	V	1, 2
Pump voltage	V_{PP}	1.746	1.8	1.908	1.746	1.8	1.908	V	2
Reference voltages for DQ and DQE inputs in PAM3 mode	V_{REFDL}	–	0.625 × V_{DDQ}	–	–	0.625 × V_{DDQ}	–	V	3, 4
	V_{REFDH}	–	0.875 × V_{DDQ}	–	–	0.875 × V_{DDQ}	–	V	
Reference voltages for DQ and DQE inputs in NRZ mode with ODT off	V_{REFD2}	–	0.5 × V_{DDQ}	–	–	0.5 × V_{DDQ}	–	V	3, 4
Reference voltages for CA inputs with ODT on	V_{REFCA}	–	0.725 × V_{DDQ}	–	–	0.725 × V_{DDQ}	–	V	3, 5
Reference voltages for CA inputs with HalfVREFC with ODT off	V_{REFCA2}	–	0.5 × V_{DDQ}	–	–	0.5 × V_{DDQ}	–	V	3, 5, 6
DC input logic High voltage with V_{REFCA} for CA inputs	V_{IHCA}(DC)	V_{REFCA} + 0.12	–	–	V_{REFCA} + 0.1	–	–	V	
DC input logic Low voltage with V_{REFCA} for CA inputs	V_{ILCA}(DC)	–	–	V_{REFCA} – 0.12	–	–	V_{REFCA} – 0.1	V	
DC input logic High voltage with V_{REFCA2} for CA inputs	V_{IHCA2}(DC)	V_{REFCA2} + 0.24	–	–	V_{REFCA2} + 0.22	–	–	V	
DC input logic Low voltage with V_{REFCA2} for CA inputs	V_{ILCA2}(DC)	–	–	V_{REFCA2} – 0.24	–	–	V_{REFCA2} – 0.22	V	
DC input logic High voltage with V_{REFD2} for DQ and DQE inputs	V_{IHD2}(DC)	V_{REFD2} + 0.24	–	–	V_{REFD2} + 0.22	–	–	V	
DC input logic Low voltage with V_{REFD2} for DQ and DQE inputs	V_{ILD2}(DC)	–	–	V_{REFD2} – 0.24	–	–	V_{REFD2} – 0.22	V	
RESET_n input logic high voltage	V_{IHR}	0.8 × V_{DD}	–	–	0.8 × V_{DD}	–	–	V	
RESET_n input logic low voltage	V_{IHL}	–	–	0.2 × V_{DD}	–	–	0.2 × V_{DD}	V	
CA[4:0] input logic high voltage for device configuration at exit from reset state	V_{IHR2}	0.8 × V_{DDQ}	–	–	0.8 × V_{DDQ}	–	–	V	7

Table 133 — DC Operating Conditions (cont'd)

Parameter	Symbol	1.20 V Range			1.10 V Range			Unit	Notes
		Min	Typ	Max	Min	Typ	Max		
CA[4:0] input logic low voltage for device configuration at exit from reset state	V_{IHL2}	–	–	0.2 × V_{DDQ}	–	–	0.2 × V_{DDQ}	V	7
Input leakage current (any input 0V ≤ V_{IN} ≤ V_{DDQ}; all other signals not under test = 0V)	I_{IL}		–			–		µA	
Output Leakage Current (outputs are disabled; 0V ≤ V_{OUT} ≤ V_{DDQ})	I_{OZ}		–			–		µA	
WCK clock input mid-point voltage	V_{MP}(DC)	V_{REFCA} - 0.1V	–	V_{REFCA} + 0.1V	V_{REFCA} - 0.1V	–	V_{REFCA} + 0.1V	V	8
WCK clock input differential voltage	V_{IDWCK}(DC)	0.175	–	–	0.165	–	–	V	9
External resistor value	ZQ	118.8	120	121.2	118.8	120	121.2	Ω	

NOTE 1 GDDR7 SGRAMs are designed to tolerate PCB designs with separate V_{DD} and V_{DDQ} power regulators.
NOTE 2 DC bandwidth is limited to 20 MHz.
NOTE 3 AC noise in the system is estimated at 50mV pk-pk for the purpose of DRAM design.
NOTE 4 The reference voltage for the DQ and DQE inputs is generated internally, and its values are determined by the PAM3, VREFDH and VREFDL level and offset mode register bits. The typical V_{REFDH} and V_{REFDL} levels depend on the selected data termination value; the values in this table represent the ideal data eye center with 40 Ohm ODT and 40 Ohm host-side driver strength; see Mode Register 16 (MR16) and 17 (MR17) for details.
NOTE 5 The reference voltage for the CA inputs is generated internally, and its values are determined by the Half VREFC and VREFCA mode register bits. The typical V_{REFCA} level depends on the selected CA termination value; the values in this table represent the ideal data eye center with 48 Ohm ODT and 40 Ohm host-side driver strength; see Mode Register 13 (MR13) for details.
NOTE 6 Programmable V_{REFCA} levels are not supported with V_{REFCA2}.
NOTE 7 V_{IHR2} and V_{ILR2} apply at exit from reset state when latching default device configurations.
NOTE 8 The WCK_t and WCK_c input reference level (for timing referenced to WCK_t and WCK_c) is the point at which WCK_t and WCK_c cross. Please refer to the applicable timings in the AC Timings table.
NOTE 9 V_{IDWCK} is the magnitude of the difference between the input level on WCK_t and the input level on WCK_c. The input reference level for signals other than WCK_t and WCK_c is either V_{REFC}, V_{REFC2}, V_{REFDH}, V_{REFDL} or V_{REFD2}.

8.8 AC Operating Conditions

GDDR7 does not specify AC operating conditions as the dynamic behavior of a GDDR7 DRAM is expected to depend a lot on the characteristics of the individual target system. Users therefore are encouraged to work with the DRAM vendors on related topics like signaling, clocking, jitter or power supply noise, to validate the stable operation of the DRAM in the target system.

9 IDD

This chapter defines IDD and IDDQ measurement conditions such as test load and patterns.

- IDD currents are measured as time-averaged currents, with all V_{DD} balls of the device under test tied together. IDDQ and IPP currents are not included in IDD currents.
- IDDQ currents are measured as time-averaged currents with all V_{DDQ} balls of the device under test tied together. IDD and IPP currents are not included in IDDQ currents:
- IPP currents are measured as time-averaged currents with all V_{PP} balls of the device under test tied together. IDD and IDDQ currents are not included in IPP currents:

Attention: IDDQ values cannot be directly used to calculate IO power of the device. They can be used to support correlation of simulated IO power to actual IO power.

For IDD, IPP and IDDQ measurements, the following definitions apply:

- CABI, CAPAR, CAPARBLK, PSN/SEV, RDCRC, WRCRC and PAM3 are enabled;
- DQE_HZ, ECS_ON, CAOSC and Data Scramble are disabled;
- WCK frequency is set with DCC enabled (if DCC supported);
- VDD and VDDQ range are set;
- DQs and DQE are ODT during all cycles without data traffic;
- ODTs are enabled with 40 Ohm for DQs and 48 Ohm for WCKs and CAs; ODT offsets are disabled; ZQ auto calibration is enabled;
- WLmrs, RLmrs, RAS, WR are set to appropriate values;
- CRC is enabled for READs and WRITEs with appropriate WRCRC2ERR and DQERL settings;
- RCKTYPE is set to differential and RCKLEVEL is set to Full swing;
- Command Address (CA) inputs include the CABI bit in CA1
- Each channel consists of ten DQs and one DQE pin;
- RNOP2, CNOP2 and all address inputs pulled HIGH during idle command cycles;
- RCK is always on mode in IDD3NR, IDD4R and IDD7 with differential RCKTYPE and full swing RCKLEVEL; RCK is disabled in all other IDD condition.

- Data pattern (one burst of 32) used with IDD4R, IDD4W and IDD7 pattern:
  - This pattern are specified as un-encoded raw payload data;
  - For IDD4R/W, the pattern are defined based on the polynomial of $X^{14} + X^{13} + X^{12} + X^2 + 1$ with 16383 length.
  - For IDD7, when cycling thru all 16 banks, the same pattern can be replicated 4x.
  - To avoid DQ to DQ correlation, the PRBS will sequentially generate blocks of 256 raw data bits for each transfer, that are distributed to the 8 internal data lanes (pre-encoding). For example: the first 32 consecutive generated bits to the Data lane 0, the next 32 bits on Data lane 1,… , until data lane 7.
  - The PRBS can be initiated with any random seed.
  - DRAM vendors can provide the additional IDD numbers to reflect the worst case scenario outside JEDEC.

9 IDD (cont'd)

- Basic IDD, IPP and IDDQ Measurement Conditions including timings used for IDD Measurement-Loop Patterns are described in *TABLE 134*.
- IDD Measurements are done after properly initializing the device. This includes the pre-load of the memory array with data pattern used with IDD4R and IDD7 measurements.
- The IDD, IPP and IDDQ Measurement-Loop patterns shall be executed at least once before actual measurement is started.
- The measurements shall be taken separately with the device configured to 4-channel mode and 2-channel mode.
- Measurements are taken per device with the same IDD Measurement-Loop Patterns on all active channels.

9 IDD (cont'd)

Table 134 — IDD Specifications and Test Conditions

PARAMETER/CONDITION	SYMBOL	TEMP	4CH Mode			2CH Mode			NOTES
			Bin[11] n	Bin n+1	…	Bin[11] n	Bin n+1	…	
One Bank Activate Precharge Current: tWCK = tWCK(min); DQ, DQE is High; bank and row addresses as defined in *TABLE 136* with ACT command; AC timings as defined in *TABLE 135*.	IDD0	TJopr(max)[10]							
	IDDQ0								
	IPP0								2
Four Bank Activate Precharge Current: tWCK = tWCK(min); DQ, DQE is High; bank and row addresses as defined in *TABLE 137* with ACT command; AC timings as defined in *TABLE 135*.	IDD0F								
	IDDQ0F								
	IPP0F								2
Precharge Power-down Current: tWCK = tWCK(min); all banks idle; device is in precharge power-down state; all other inputs are High; WCK inputs are stable;	IDD2P								
	IDDQ2P								
	IPP2P								2
Precharge Standby Current: tWCK = tWCK(min); all banks idle; all other inputs are High; WCK inputs are stable;	IDD2N								
	IDDQ2N								
	IPP2N								2
Precharge Sleep mode Current: all banks idle; device is in sleep mode;WCK inputs are High; all CA inputs are High; all DQs are High; all other inputs are don't care;	IDD2S								
	IDDQ2S								
	IPP2S								2
Active Power-down Current: tWCK = tWCK(min); one bank active; device is in precharge power-down state; all other inputs are High; WCK inputs are stable	IDD3P								
	IDDQ3P								
	IPP3P								2
Active Standby Current: tWCK = tWCK(min); one bank active; all other inputs are High; WCK inputs are stable;	IDD3N								
	IDDQ3N								
	IPP3N								2
Active Standby Current with RCK enabled : tWCK = tWCK(min); one bank active; all other inputs are High; WCK inputs are stable; RCK enabled;	IDD3NR								
	IDDQ3NR								
	IPP3NR								2
Active Standby Current with 4banks activated: tWCK = tWCK(min); four banks active; all other inputs are High; WCK inputs are stable;	IDD3NF								
	IDDQ3NF								
	IPP3NF								2
Read Burst Current: tWCK = tWCK(min); 4 banks activate; continuous read burst across the activated 4 banks with 50% internal data-bus toggle (PRBS14 generated) on each data transfer, as defined in *TABLE 138*; bank and column addresses as defined in *TABLE 137*; with READ command; RCK enabled;	IDD4R								
	IDDQ4R								
	IPP4R								2

Table 134 — IDD Specifications and Test Conditions (cont'd)

PARAMETER/CONDITION	SYMBOL	TEMP	4CH Mode			2CH Mode			NOTES
			Bin[11] n	Bin n+1	…	Bin[11] n	Bin n+1	…	
Write Burst Current: tWCK = tWCK(min); 4 banks activate; continuous write burst across the activated 4 banks with 50% internal data-bus toggle (PRBS14 generated) on each data transfer, as defined in *TABLE 139* with WRITE command;	IDD4W								
	IDDQ4W								
	IPP4W								2
Refresh All Bank Current: tWCK = tWCK(min); timing reference is in *TABLE 135*; all other inputs are high; WCK inputs are stable;	IDD5B								
	IDDQ5B								
	IPP5B								2
Refresh Per Bank Current: tWCK = tWCK(min); timing reference is in *TABLE 135*; all other inputs are high; WCK inputs are stable;	IDD5P								
	IDDQ5P								
	IPP5P								2
Self Refresh Current: tWCK = tWCK(min); all other inputs are High; WCK inputs are stable	IDD6								
	IDDQ6								
	IPP6								2
Sixteen Bank Interleave Read Current: tWCK = tWCK(min);one 16 banks activated and precharged at tRC(min); continuous read burst across the activated 16banks with 50% data toggle (PRBS14 generated) on each data transfer; bank, row and column addresses as defined in *TABLE 140*; with ACT and READ/READA commands; RCK enabled;	IDD7								
	IDDQ7								
	IPP7								2
Self Refresh Sleep: All other inputs are High; WCK inputs are High. TCSR is disabled.	IDD6S	Tn							3, 5
	IDD6QS								3, 5
	IPP6S								2, 3, 5
	IDD6S	Tn+1 (optional)							3, 5
	IDD6QS								3, 5
	IPP6S								2, 3, 5
	IDD6S	Tn+2 (optional)							3, 5, 9
	IDD6QS								3, 5, 9
	IPP6S								2, 3, 5, 9

Table 134 — IDD Specifications and Test Conditions (cont'd)

PARAMETER/CONDITION	SYMBOL	TEMP	4CH Mode			2CH Mode			NOTES
			Bin[11] n	Bin n+1	…	Bin[11] n	Bin n+1	…	
Self Refresh Sleep TCSR: All other inputs are High; WCK inputs are High. TCSR is enabled.	IDD6S-A	TJopr(min) - Ta							5, 6, 7
	IDDQ6S-A								5, 6, 7
	IPP6S-A								2, 5, 6, 7
	IDD6S-A	Tb - Ty (optional)							5, 6, 7
	IDDQ6S-A								5, 6, 7
	IPP6S-A								2, 5, 6, 7
	IDD6S-A	Tz - TJopr(max)							5, 6, 7
	IDDQ6S-A								5, 6, 7
	IPP6S-A								2, 5, 6, 7
Hibernate Self Refresh Sleep: All other inputs are High; WCK inputs are High. TCSR controls the interval of self refresh.	IDD6H	TJopr(min) - Ta							5, 7, 8, 9
	IDDQ6H								5, 7, 8, 9
	IPP6H								2, 5, 7, 8, 9
	IDD6H	Tb - Ty (optional)							5, 7, 8, 9
	IDDQ6H								5, 7, 8, 9
	IPP6H								2, 5, 7, 8, 9
	IDD6H	Tz - TJopr(max)							5, 7, 8, 9
	IDDQ6H								5, 7, 8, 9
	IPP6H								2, 5, 7, 8, 9

NOTE 1 Min tRC or tRFC for IDD measurements is the smallest multiple of tCK that meets the minimum of the absolute value for the respective parameter.

NOTE 2 IPP currents have the same definition as IDD except that the current on the VPP supply is measured.

NOTE 3 IDD6S is applicable for MR8 OP[10:9]= 00_B, 01_B or 10_B (fixed self refresh interval). Tn is the temperature or temperature range associated with 00_B. Tn+1 is associated with 01_B and Tn+2 is associated with 10_B which are both optional fixed self refresh intervals. See vendor datasheets for more details.

NOTE 4 Max. values for IDD currents considering worst case conditions of process, temperature and voltage.

NOTE 5 Refer to the vendor datasheets for the value specification method (e.g., max, typical) for IDD6S, IDD6S-A, and IDD6H.

NOTE 6 IDD6S-A is only when the Temperature Controlled Self Refresh feature is enabled by MR8 OP[10:9] = 11_B (Temperature compensated self refresh rate)

NOTE 7 The number of discrete temperature ranges supported and the associated Ta - Tz, and the values are supplier/ design specific. Temperature ranges are intended to denote the nominal trip points for the internal temperature sensor to bracket discrete self refresh rates internal to the DRAM. Refer to supplier datasheet for more information.

NOTE 8 IDD6H is for Hibernate Self Refresh Sleep as enabled with MR8 OP11. Hibernate self refresh is allowed to be entered over the full operating temperature range. See vendor datasheets for details.

NOTE 9 IDD6H values are characterized but not necessarily tested, see vendor data sheets for more details.

NOTE 10 TJopr(max) is the max temperature of the supported JESD402-1 ranges. See the Operating Temperature Range Section.

NOTE 11 Bin n, Bin n+1 are the speed bins supported. See vendor datasheets for details on the frequencies and voltages supported.

9 IDD (cont'd)

Table 135 — AC Parameter Condition for IDD Test

Parameter		Value		Unit	Notes
		Min	Max		
tRC		50		ns	
tRAS		30	33	ns	1
tRP		17	20	ns	1
tRCDRD		18		ns	
tRFCab	16, 24, 32Gb	200		ns	
	48, 64Gb	TBD		ns	
tRFCpb	16, 24, 32Gb	120		ns	
	48, 64Gb	TBD		ns	
tREFI		1.9		µs	

NOTE 1 tRAS and tRP are vendor specific, but the sum of tRAS and tRP should be the same as tRC = 50 ns.

Table 136 — IDD0 Measurement-Loop Pattern

Sub-Loop	CK4 Cycle	Row Command	Col Command [1]	Data
0	0	ACT Bank 0 Row E7C7'h	CNOP1	-
	1		CNOP1	
	2	RNOP2	CNOP2	
	…	repeat NOP until cycle (nRAS -1)	CNOP2	
	nRAS	PREab	CNOP1	
	nRAS + 1	RNOP2	CNOP2	
	…	repeat NOP until cycle (nRAS -1)	CNOP2	
1	nRC	repeat sub-loop 0 pattern until cycle (2* nRC -1) with Bank 5 and RA = 03C7h instead;		
2	2*RC	repeat sub-loop 0 pattern until cycle (3* nRC -1) with Bank 10 and RA = 03C7h instead;		
3	3*RC	repeat sub-loop 0 pattern until cycle (4* nRC -1) with Bank 15 and RA = 03C7h instead;		

NOTE 1 CNOP1 in this table is supposed to be set as all "H" except CAPAR bit.

9 IDD (cont'd)

Table 137 — IDD0F Measurement-Loop Pattern

Sub-Loop	CK4 Cycle	Row Command	Col Command [1]	Data
Initialization	0	ACT Bank 0 Row E7C7'h	CNOP1	-
	1			
	…	repeat RNOP2 until cycle (nRC /4-1)	CNOP2	
	nRC/4	ACT Bank 5 Row E7C7'h	CNOP1	
	nRC/4+1			
	…	repeat RNOP2 until cycle (nRC *2/4-1)	CNOP2	
	nRC *2/4	ACT Bank 10 Row E7C7'h	CNOP1	
	nRC *2/4+1			
	…	repeat RNOP2 until cycle (nRC *3/4-1)	CNOP2	
	nRC *3/4	ACT Bank 15 Row E7C7'h	CNOP1	
	nRC *3/4+1			
	…	repeat RNOP2 until cycle (nRC -1)	CNOP2	
0	nRC	PREpb Bank 0	CNOP1	-
	nRC+1			
	nRC+2	ACT Bank 4 Row 03C7'h	CNOP1	
	nRC+3			
	…	repeat RNOP2 until cycle (nRC *5/4 -1)	CNOP2	
	nRC*5/4	PREpb Bank 5	CNOP1	
	nRC*5/4+1			
	nRC*5/4+2	ACT Bank 9 Row 03C7'h	CNOP1	
	nRC*5/4+3			
	…	repeat RNOP2 until cycle (nRC *6/4-1)	CNOP2	
	nRC*6/4	PREpb Bank 10	CNOP1	
	nRC*6/4+1			
	nRC*6/4+2	ACT Bank 14 Row 03C7'h	CNOP1	
	nRC*6/4+3			
	…	repeat RNOP2 until cycle (nRC *7/4-1)	CNOP2	
	nRC*7/4	PREpb Bank 15	CNOP1	
	nRC*7/4+1			
	nRC*7/4+2	ACT Bank 3 Row 03C7'h	CNOP1	
	nRC*7/4+3			
	…	repeat RNOP2 until cycle (nRC *8/4-1)	CNOP2	
1	2*nRC	repeat sub-loop 0 pattern until cycle (3* nRC -1) with Bank 4, Bank 9, Bank 14, Bank 3 for PREpb in order and Bank 8, Bank 13, Bank 2, Bank 7 for ACT in serial with RA = 03C7h instead;		
2	3*nRC	repeat sub-loop 0 pattern until cycle (4* nRC -1) with Bank 8, Bank 13, Bank 2, Bank 7 for PREpb in order and Bank 12, Bank 1, Bank 6, Bank 11 for ACT in serial with RA = 03C7h instead;		
3	4*nRC	repeat sub-loop 0 pattern until cycle (5* nRC -1) with Bank 12, Bank 1, Bank 6, Bank 11 for PREpb in order and Bank 0, Bank 5, Bank 10, Bank 15 for ACT in serial with RA = 03C7h instead;		
NOTE CNOP1 in this table is supposed to be set as all "H" except CAPAR bit.				

9 IDD (cont'd)

Table 138 — IDD4R Measurement-Loop Pattern

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	0	RNOP2	READ Bank 0 Col 00'h	PRBS-14
	1	RNOP2		
	2	RNOP2	READ Bank 5 Col 07'h	
	3	RNOP2		
	4	RNOP2	READ Bank 10 Col 14'h	
	5	RNOP2		
	6	RNOP2	READ Bank 15 Col 20'h	
	7	RNOP2		
	8	RNOP2	READ Bank 0 Col 3C'h	
	9	RNOP2		
	10	RNOP2	READ Bank 5 Col 29'h	
	11	RNOP2		
	12	RNOP2	READ Bank 10 Col 11'h	
	13	RNOP2		
	14	RNOP2	READ Bank 15 Col 32'h	
	15	RNOP2		
	16	RNOP2	READ Bank 0 Col 17'h	
	17	RNOP2		
	18	RNOP2	READ Bank 5 Col 3B'h	
	19	RNOP2		
	20	RNOP2	READ Bank 10 Col 21'h	
	21	RNOP2		
	22	RNOP2	READ Bank 15 Col 2C'h	
	23	RNOP2		
	24	RNOP2	READ Bank 0 Col 3F'h	
	25	RNOP2		
	26	RNOP2	READ Bank 5 Col 0B'h	
	27	RNOP2		
	28	RNOP2	READ Bank 10 Col 17'h	
	29	RNOP2		
	30	RNOP2	READ Bank 15 Col 0E'h	
	31	RNOP2		
	32	RNOP2	READ Bank 0 Col 07'h	
	33	RNOP2		
	34	RNOP2	READ Bank 5 Col 14'h	
	35	RNOP2		
	36	RNOP2	READ Bank 10 Col 20'h	
	37	RNOP2		

Table 138 — IDD4R Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	38	RNOP2	READ Bank 15 Col 3C'h	PRBS-14
	39	RNOP2		
	40	RNOP2	READ Bank 0 Col 29'h	
	41	RNOP2		
	42	RNOP2	READ Bank 5 Col 11'h	
	43	RNOP2		
	44	RNOP2	READ Bank 10 Col 32'h	
	45	RNOP2		
	46	RNOP2	READ Bank 15 Col 17'h	
	47	RNOP2		
	48	RNOP2	READ Bank 0 Col 3B'h	
	49	RNOP2		
	50	RNOP2	READ Bank 5 Col 21'h	
	51	RNOP2		
	52	RNOP2	READ Bank 10 Col 2C'h	
	53	RNOP2		
	54	RNOP2	READ Bank 15 Col 3F'h	
	55	RNOP2		
	56	RNOP2	READ Bank 0 Col 0B'h	
	57	RNOP2		
	58	RNOP2	READ Bank 5 Col 17'h	
	59	RNOP2		
	60	RNOP2	READ Bank 10 Col 0E'h	
	61	RNOP2		
	62	RNOP2	READ Bank 15 Col 00'h	
	63	RNOP2		
	64	RNOP2	READ Bank 0 Col 14'h	
	65	RNOP2		
	66	RNOP2	READ Bank 5 Col 20'h	
	67	RNOP2		
	68	RNOP2	READ Bank 10 Col 3C'h	
	69	RNOP2		
	70	RNOP2	READ Bank 15 Col 29'h	
	71	RNOP2		
	72	RNOP2	READ Bank 0 Col 11'h	
	73	RNOP2		
	74	RNOP2	READ Bank 5 Col 32'h	
	75	RNOP2		
	76	RNOP2	READ Bank 10 Col 17'h	
	77	RNOP2		
	78	RNOP2	READ Bank 15 Col 3B'h	
	79	RNOP2		
	80	RNOP2	READ Bank 0 Col 21'h	
	81	RNOP2		

Table 138 — IDD4R Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	82	RNOP2	READ Bank 5 Col 2C'h	PRBS-14
	83	RNOP2		
	84	RNOP2	READ Bank 10 Col 3F'h	
	85	RNOP2		
	86	RNOP2	READ Bank 15 Col 0B'h	
	87	RNOP2		
	88	RNOP2	READ Bank 0 Col 17'h	
	89	RNOP2		
	90	RNOP2	READ Bank 5 Col 0E'h	
	91	RNOP2		
	92	RNOP2	READ Bank 10 Col 00'h	
	93	RNOP2		
	94	RNOP2	READ Bank 15 Col 07'h	
	95	RNOP2		
	96	RNOP2	READ Bank 0 Col 20'h	
	97	RNOP2		
	98	RNOP2	READ Bank 5 Col 3C'h	
	99	RNOP2		
	100	RNOP2	READ Bank 10 Col 29'h	
	101	RNOP2		
	102	RNOP2	READ Bank 15 Col 11'h	
	103	RNOP2		
	104	RNOP2	READ Bank 0 Col 32'h	
	105	RNOP2		
	106	RNOP2	READ Bank 5 Col 17'h	
	107	RNOP2		
	108	RNOP2	READ Bank 10 Col 3B'h	
	109	RNOP2		
	110	RNOP2	READ Bank 15 Col 21'h	
	111	RNOP2		
	112	RNOP2	READ Bank 0 Col 2C'h	
	113	RNOP2		
	114	RNOP2	READ Bank 5 Col 3F'h	
	115	RNOP2		
	116	RNOP2	READ Bank 10 Col 0B'h	
	117	RNOP2		
	118	RNOP2	READ Bank 15 Col 17'h	
	119	RNOP2		

Table 138 — IDD4R Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	120	RNOP2	READ Bank 0 Col 0E'h	PRBS-14
	121	RNOP2		
	122	RNOP2	READ Bank 5 Col 00'h	
	123	RNOP2		
	124	RNOP2	READ Bank 10 Col 07'h	
	125	RNOP2		
	126	RNOP2	READ Bank 15 Col 14'h	
	127	RNOP2		

NOTE 1 PRBS data patterns are defined based on the polynomial of $X^{14} + X^{13} + X^{12} + X^2 + 1$.

NOTE 2 DRAM vendors can use a vendor specific pattern designed to generate IDD values as close as possible to a PRBS data pattern described in NOTE 1 as a temporary alternative. Also, DRAM vendors can provide a conversion factor that allows the projection of the equivalent IDD values compared to the PRBS pattern.

9 IDD (cont'd)

Table 139 — IDD4W Measurement-Loop Pattern

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	0	RNOP2	WRITE Bank 0 Col 00'h	PRBS-14
	1	RNOP2		
	2	RNOP2	WRITE Bank 5 Col 07'h	
	3	RNOP2		
	4	RNOP2	WRITE Bank 10 Col 14'h	
	5	RNOP2		
	6	RNOP2	WRITE Bank 15 Col 20'h	
	7	RNOP2		
	8	RNOP2	WRITE Bank 0 Col 3C'h	
	9	RNOP2		
	10	RNOP2	WRITE Bank 5 Col 29'h	
	11	RNOP2		
	12	RNOP2	WRITE Bank 10 Col 11'h	
	13	RNOP2		
	14	RNOP2	WRITE Bank 15 Col 32'h	
	15	RNOP2		
	16	RNOP2	WRITE Bank 0 Col 17'h	
	17	RNOP2		
	18	RNOP2	WRITE Bank 5 Col 3B'h	
	19	RNOP2		
	20	RNOP2	WRITE Bank 10 Col 21'h	
	21	RNOP2		
	22	RNOP2	WRITE Bank 15 Col 2C'h	
	23	RNOP2		
	24	RNOP2	WRITE Bank 0 Col 3F'h	
	25	RNOP2		
	26	RNOP2	WRITE Bank 5 Col 0B'h	
	27	RNOP2		
	28	RNOP2	WRITE Bank 10 Col 17'h	
	29	RNOP2		
	30	RNOP2	WRITE Bank 15 Col 0E'h	
	31	RNOP2		
	32	RNOP2	WRITE Bank 0 Col 07'h	
	33	RNOP2		
	34	RNOP2	WRITE Bank 5 Col 14'h	
	35	RNOP2		
	36	RNOP2	WRITE Bank 10 Col 20'h	
	37	RNOP2		

Table 139 — IDD4W Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	38	RNOP2	WRITE Bank 15 Col 3C'h	PRBS-14
	39	RNOP2		
	40	RNOP2	WRITE Bank 0 Col 29'h	
	41	RNOP2		
	42	RNOP2	WRITE Bank 5 Col 11'h	
	43	RNOP2		
	44	RNOP2	WRITE Bank 10 Col 32'h	
	45	RNOP2		
	46	RNOP2	WRITE Bank 15 Col 17'h	
	47	RNOP2		
	48	RNOP2	WRITE Bank 0 Col 3B'h	
	49	RNOP2		
	50	RNOP2	WRITE Bank 5 Col 21'h	
	51	RNOP2		
	52	RNOP2	WRITE Bank 10 Col 2C'h	
	53	RNOP2		
	54	RNOP2	WRITE Bank 15 Col 3F'h	
	55	RNOP2		
	56	RNOP2	WRITE Bank 0 Col 0B'h	
	57	RNOP2		
	58	RNOP2	WRITE Bank 5 Col 17'h	
	59	RNOP2		
	60	RNOP2	WRITE Bank 10 Col 0E'h	
	61	RNOP2		
	62	RNOP2	WRITE Bank 15 Col 00'h	
	63	RNOP2		
	64	RNOP2	WRITE Bank 0 Col 14'h	
	65	RNOP2		
	66	RNOP2	WRITE Bank 5 Col 20'h	
	67	RNOP2		
	68	RNOP2	WRITE Bank 10 Col 3C'h	
	69	RNOP2		
	70	RNOP2	WRITE Bank 15 Col 29'h	
	71	RNOP2		
	72	RNOP2	WRITE Bank 0 Col 11'h	
	73	RNOP2		
	74	RNOP2	WRITE Bank 5 Col 32'h	
	75	RNOP2		
	76	RNOP2	WRITE Bank 10 Col 17'h	
	77	RNOP2		

Table 139 — IDD4W Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	78	RNOP2	WRITE Bank 15 Col 3B'h	PRBS-14
	79	RNOP2		
	80	RNOP2	WRITE Bank 0 Col 21'h	
	81	RNOP2		
	82	RNOP2	WRITE Bank 5 Col 2C'h	
	83	RNOP2		
	84	RNOP2	WRITE Bank 10 Col 3F'h	
	85	RNOP2		
	86	RNOP2	WRITE Bank 15 Col 0B'h	
	87	RNOP2		
	88	RNOP2	WRITE Bank 0 Col 17'h	
	89	RNOP2		
	90	RNOP2	WRITE Bank 5 Col 0E'h	
	91	RNOP2		
	92	RNOP2	WRITE Bank 10 Col 00'h	
	93	RNOP2		
	94	RNOP2	WRITE Bank 15 Col 07'h	
	95	RNOP2		
	96	RNOP2	WRITE Bank 0 Col 20'h	
	97	RNOP2		
	98	RNOP2	WRITE Bank 5 Col 3C'h	
	99	RNOP2		
	100	RNOP2	WRITE Bank 10 Col 29'h	
	101	RNOP2		
	102	RNOP2	WRITE Bank 15 Col 11'h	
	103	RNOP2		
	104	RNOP2	WRITE Bank 0 Col 32'h	
	105	RNOP2		
	106	RNOP2	WRITE Bank 5 Col 17'h	
	107	RNOP2		
	108	RNOP2	WRITE Bank 10 Col 3B'h	
	109	RNOP2		
	110	RNOP2	WRITE Bank 15 Col 21'h	
	111	RNOP2		
	112	RNOP2	WRITE Bank 0 Col 2C'h	
	113	RNOP2		
	114	RNOP2	WRITE Bank 5 Col 3F'h	
	115	RNOP2		
	116	RNOP2	WRITE Bank 10 Col 0B'h	
	117	RNOP2		
	118	RNOP2	WRITE Bank 15 Col 17'h	
	119	RNOP2		
	120	RNOP2	WRITE Bank 0 Col 0E'h	
	121	RNOP2		

Table 139 — IDD4W Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	122	RNOP2	WRITE Bank 5 Col 00'h	PRBS-14
	123	RNOP2		
	124	RNOP2	WRITE Bank 10 Col 07'h	
	125	RNOP2		
	126	RNOP2	WRITE Bank 15 Col 14'h	
	127	RNOP2		

NOTE 1 PRBS data patterns are defined based on the polynomial of $X^{14} + X^{13} + X^{12} + X^2 + 1$.

NOTE 2 DRAM vendors can use a vendor specific pattern designed to generate IDD values as close as possible to a PRBS data pattern described in NOTE 1 as a temporary alternative. Also, DRAM vendors can provide a conversion factor that allows the projection of the equivalent IDD values compared to the PRBS pattern.

9 IDD (cont'd)

Table 140 — IDD7 Measurement-Loop Pattern

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
Initialization	0	ACT Bank 6 Row 4F'h	CNOP1	-
	1		CNOP1	
	2	RNOP2	CNOP1	
	3	RNOP2	CNOP1	
	4	RNOP2	CNOP1	
	5	RNOP2	CNOP1	
	6	RNOP2	CNOP1	
	7	RNOP2	CNOP1	
	8	ACT Bank 7 Row 4F'h	CNOP1	
	9		CNOP1	
	10	RNOP2	CNOP1	
	11	RNOP2	CNOP1	
	12	RNOP2	CNOP1	
	13	RNOP2	CNOP1	
	14	RNOP2	CNOP1	
	15	RNOP2	CNOP1	
	16	ACT Bank 8 Row 4F'h	CNOP1	
	17		CNOP1	
	18	RNOP2	CNOP1	
	19	RNOP2	CNOP1	
	20	RNOP2	CNOP1	
	21	RNOP2	CNOP1	
	22	RNOP2	CNOP1	
	23	RNOP2	CNOP1	
	24	ACT Bank 9 Row 4F'h	CNOP1	
	25		CNOP1	
	26	RNOP2	CNOP1	
	27	RNOP2	CNOP1	
	28	RNOP2	CNOP1	
	29	RNOP2	CNOP1	
	30	RNOP2	CNOP1	
	31	RNOP2	CNOP1	
	32	ACT Bank 10 Row 4F'h	CNOP1	
	33		CNOP1	
	34	RNOP2	CNOP1	
	35	RNOP2	CNOP1	
	36	RNOP2	CNOP1	
	37	RNOP2	CNOP1	
	38	RNOP2	CNOP1	
	39	RNOP2	CNOP1	
	40	ACT Bank 11 Row 4F'h	CNOP1	
	41		CNOP1	

Table 140 — IDD7 Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
Initialization	42	RNOP2	CNOP1	-
	43	RNOP2	CNOP1	
	44	RNOP2	CNOP1	
	45	RNOP2	CNOP1	
	46	RNOP2	CNOP1	
	47	RNOP2	CNOP1	
	48	ACT Bank 12 Row 4F'h	CNOP1	
	49		CNOP1	
	50	RNOP2	CNOP1	
	51	RNOP2	CNOP1	
	52	RNOP2	CNOP1	
	53	RNOP2	CNOP1	
	54	RNOP2	CNOP1	
	55	RNOP2	CNOP1	
	56	ACT Bank 13 Row 4F'h	CNOP1	
	57		CNOP1	
	58	RNOP2	CNOP1	
	59	RNOP2	CNOP1	
	60	RNOP2	CNOP1	
	61	RNOP2	CNOP1	
	62	RNOP2	CNOP1	
	63	RNOP2	CNOP1	
	64	ACT Bank 14 Row 4F'h	CNOP1	
	65		CNOP1	
	66	RNOP2	CNOP1	
	67	RNOP2	CNOP1	
	68	RNOP2	CNOP1	
	69	RNOP2	CNOP1	
	70	RNOP2	CNOP1	
	71	RNOP2	CNOP1	
	72	ACT Bank 15 Row 4F'h	CNOP1	
	73		CNOP1	
	74	RNOP2	CNOP1	
	74	RNOP2	CNOP1	
	76	RNOP2	CNOP1	
	77	RNOP2	CNOP1	
	78	RNOP2	CNOP1	
	79	RNOP2	CNOP1	

Table 140 — IDD7 Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	80	ACT Bank 0 Row 4F'h	READA Bank 6 Col 1F'h	PRBS-14
	81			
	82	RNOP2	READ Bank 10 Col 1A'h	
	83	RNOP2		
	84	RNOP2	READ Bank 9 Col 3F'h	
	85	RNOP2		
	86	RNOP2	READ Bank 8 Col 33'h	
	87	RNOP2		
	88	ACT Bank 1 Row 4F'h	READA Bank 7 Col 1F'h	
	89			
	90	RNOP2	READ Bank 11 Col 1A'h	
	91	RNOP2		
	92	RNOP2	READ Bank 10 Col 3F'h	
	93	RNOP2		
	94	RNOP2	READ Bank 9 Col 33'h	
	95	RNOP2		
	96	ACT Bank 2 Row 4F'h	READA Bank 8 Col 1F'h	
	97			
	98	RNOP2	READ Bank 12 Col 1A'h	
	99	RNOP2		
	100	RNOP2	READ Bank 11 Col 3F'h	
	101	RNOP2		
	102	RNOP2	READ Bank 10 Col 33'h	
	103	RNOP2		
	104	ACT Bank 3 Row 4F'h	READA Bank 9 Col 1F'h	
	105			
	106	RNOP2	READ Bank 13 Col 1A'h	
	107	RNOP2		
	108	RNOP2	READ Bank 12 Col 3F'h	
	109	RNOP2		
	110	RNOP2	READ Bank 11 Col 33'h	
	111	RNOP2		
	112	ACT Bank 4 Row 4F'h	READA Bank 10 Col 1F'h	
	113			
	114	RNOP2	READ Bank 14 Col 1A'h	
	115	RNOP2		
	116	RNOP2	READ Bank 13 Col 3F'h	
	117	RNOP2		
	118	RNOP2	READ Bank 12 Col 33'h	
	119	RNOP2		
	120	ACT Bank 5 Row 4F'h	READA Bank 11 Col 1F'h	
	121			
	122	RNOP2	READ Bank 15 Col 1A'h	
	123	RNOP2		
	124	RNOP2	READ Bank 14 Col 3F'h	
	125	RNOP2		
	126	RNOP2	READ Bank 13 Col 33'h	
	127	RNOP2		

Table 140 — IDD7 Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	128	ACT Bank 6 Row 4F'h	READA bank 12 Col 1F'h	PRBS-14
	129			
	130	RNOP2	READ Bank 0 Col 1A'h	
	131	RNOP2		
	132	RNOP2	READ Bank 15 Col 3F'h	
	133	RNOP2		
	134	RNOP2	READ Bank 14 Col 33'h	
	135	RNOP2		
	136	ACT Bank 7 Row 4F'h	READA Bank 13 Col 1F'h	
	137			
	138	RNOP2	READ Bank 1 Col 1A'h	
	139	RNOP2		
	140	RNOP2	READ Bank 0 Col 3F'h	
	141	RNOP2		
	142	RNOP2	READ Bank 15 Col 33'h	
	143	RNOP2		
	144	ACT Bank 8 Row 4F'h	READA Bank 14 Col 1F'h	
	145			
	146	RNOP2	READ Bank 2 Col 1A'h	
	147	RNOP2		
	148	RNOP2	READ Bank 1 Col 3F'h	
	149	RNOP2		
	150	RNOP2	READ Bank 0 Col 33'h	
	151	RNOP2		
	152	ACT Bank 9 Row 4F'h	READA Bank 15 Col 1F'h	
	153			
	154	RNOP2	READ Bank 3 Col 1A'h	
	155	RNOP2		
	156	RNOP2	READ Bank 2 Col 3F'h	
	157	RNOP2		
	158	RNOP2	READ Bank 1 Col 33'h	
	159	RNOP2		
	160	ACT Bank 10 Row 4F'h	READA Bank 0 Col 1F'h	
	161			
	162	RNOP2	READ Bank 4 Col 1A'h	
	163	RNOP2		
	164	RNOP2	READ Bank 3 Col 3F'h	
	165	RNOP2		
	166	RNOP2	READ Bank 2 Col 33'h	
	167	RNOP2		
	168	ACT Bank 11 Row 4F'h	READA Bank 1 Col 1F'h	
	169			
	170	RNOP2	READ Bank 5 Col 1A'h	
	171	RNOP2		
	172	RNOP2	READ Bank 4 Col 3F'h	
	173	RNOP2		
	174	RNOP2	READ Bank 3 Col 33'h	
	175	RNOP2		

Table 140 — IDD7 Measurement-Loop Pattern (cont'd)

Sub-Loop	CK4 Cycle	Row Command	Col Command	Data
0	176	ACT Bank 12 Row 4F'h	READA Bank 2 Col 1F'h	PRBS-14
	177			
	178	RNOP2	READ Bank 6 Col 1A'h	
	179	RNOP2		
	180	RNOP2	READ Bank 5 Col 3F'h	
	181	RNOP2		
	182	RNOP2	READ Bank 4 Col 33'h	
	183	RNOP2		
	184	ACT Bank 13 Row 4F'h	READA Bank 3 Col 1F'h	
	185			
	186	RNOP2	READ Bank 7 Col 1A'h	
	187	RNOP2		
	188	RNOP2	READ Bank 6 Col 3F'h	
	189	RNOP2		
	190	RNOP2	READ Bank 5 Col 33'h	
	191	RNOP2		
	192	ACT Bank 14 Row 4F'h	READA Bank 4 Col 1F'h	
	193			
	194	RNOP2	READ Bank 8 Col 1A'h	
	195	RNOP2		
	196	RNOP2	READ Bank 7 Col 3F'h	
	197	RNOP2		
	198	RNOP2	READ Bank 6 Col 33'h	
	199	RNOP2		
	200	ACT Bank 15 Row 4F'h	READA Bank 5 Col 1F'h	
	201			
	202	RNOP2	READ Bank 9 Col 1A'h	
	203	RNOP2		
	204	RNOP2	READ Bank 8 Col 3F'h	
	205	RNOP2		
	206	RNOP2	READ Bank 7 Col 33'h	
	207	RNOP2		

NOTE 1 PRBS data patterns are defined based on the polynomial of $X^{14} + X^{13} + X^{12} + X^2 + 1$.

NOTE 2 DRAM vendors can use a vendor specific pattern designed to generate IDD values as close as possible to a PRBS data pattern described in NOTE 1 as a temporary alternative. Also, DRAM vendors can provide a conversion factor that allows the projection of the equivalent IDD values compared to the PRBS pattern.

NOTE 3 Sub-loop 0 can be repeated indefinitely to extend the IDD7 measurement time.

10 AC Timings

Table 141 — AC Timings

PARAMETER [1,2,3]		SYMBOL	VALUES		UNIT	NOTES
			MIN	MAX		
WCK Timings						
WCK clock cycle time		tWCK		5	ns	87
WCK clock frequency		fWCK	200		MHz	87
WCK clock High-level width (average)		tWCKH(avg)			tWCK(avg)	4
WCK clock Low-level width (average)		tWCKL(avg)			tWCK(avg)	4
WCK clock High-level width (absolute)		tWCKH(abs)			tWCK(avg)	5
WCK clock Low-level width (absolute)		tWCKL(abs)			tWCK(avg)	5
CK4 cycle time		tCK4	4 × tWCK		ns	6, 87
WCK clock frequency in NRZ mode		fWCKNRZ	fWCK(min)	2000	MHz	7, 87
WCK clock frequency with single-ended RCK		fWCKSERCK	fWCK(min)		MHz	8, 87
WCK clock frequency with half swing RCK		fWCKHRCK	fWCK(min)		MHz	9, 87
WCK clock frequency for post package repair		fWCKPGM	fWCK(min)	2000	MHz	10, 87
Command / Address (CA) Interface Timings						
WCK to CA input offset		tWCK2CA			ps	11
CA to CA skew		tCA2CA			ps	12
CA input pulse width		tCAPW		–	ps	13
WCK to CA input offset voltage variation		tWCK2CA_ VOLT	–		ps/mV	14
WCK to CA input offset temperature variation		tWCK2CA_ TEMP	–		ps/°C	15
CA parity error to exit from command blocking waiting time	DRFM off	tCAPAR_ UNLOCK	RU(tRFCab(min) / tCK4) + RU(tRP(min) / tCK4)	–	nCK4	
	DRFM on		RU(tDRFM(min) / tCK4) + RU(tRP(min) / tCK4)	–	nCK4	
Period prior to a CA parity error where commands except WR and WRA are not guaranteed to be executed		tCAPAR_ UNKNOWN			ns	
Period prior to a CA parity error where WR and WRA commands are not guaranteed to be executed		tCAPAR_ UNKNOWN_WR			ns	
RCK Timings						
RCK_t/RCK_c pseudo-differential output high level width without DCC (average)		tRCKHnD(avg)			tWCK(avg)	16, 17
RCK_t/RCK_c pseudo-differential output low level width without DCC (average)		tRCKLnD(avg)			tWCK(avg)	16, 17

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]		SYMBOL	VALUES		UNIT	NOTES
			MIN	MAX		
RCK Timings (continued)						
RCK_t/RCK_c pseudo-differential output high level width with DCC (average)		tRCKH(avg)			tWCK(avg)	16, 17
RCK_t/RCK_c pseudo-differential output low level width with DCC (average)		tRCKL(avg)			tWCK(avg)	16, 17
RCK_t/RCK_c pseudo-differential output high level width (absolute)		tRCKH(abs)			tWCK(avg)	16
RCK_t/RCK_c pseudo-differential output low level width (absolute)		tRCKL(abs)			tWCK(avg)	16
WCK to RCK offset		tWCK2RCK			ps	18
WCK to RCK offset voltage variation		tWCK2RCK_ VOLT	–		ps/mV	19
WCK to RCK offset temperature variation		tWCK2RCK_ TEMP	–		ps/°C	20
RCK High-Z to static preamble time	CAPARBLK disabled	tRCK2LZ	–		ns	21
	CAPARBLK enabled		–		ns	
RCK static postamble to High-Z time		tRCK2HZ	–		ns	22
RCK static preamble		tRCK_ST	4	–	nCK4	23
RCK high-speed preamble		tRCK_HS		–	nCK4	24
RCK postamble		tRCKPST	2	–	nCK4	
RCKSTRT to RCKSTOP command delay		tRCKST2SP	RCKEN + RCK_LS + tRCK_HS(min)	–	nCK4	25
RD, RDA, RDTR, IRD, RDWTEC to RCKSTOP command delay		tRD2RCKSTOP	RL + DQERL + BL/8 + tRCKPST(min) - RCKSTOP_LAT	–	nCK4	26
RCKSTRT to RD, RDA, RDTR, IRD, RDWTEC command delay		tRCKSTRT2RD	2	–	nCK4	27
RCKSTOP to RCKSTRT or RD, RDA, RDTR, IRD, RDWTEC command delay		tRCKSP2ST		–	ns	28
RCKSTOP to MR9 or MR12 MRS programming delay		tRCKSTOP2MR9_12	RCKSTOP_LAT × tCK4 + 4 x tCK4 + tRCK2HZ(max)	–	ns	29
DQ/DQE Input Timings						
WCK to DQ/DQE offset for Write		tWCK2DQI			ps	30
DQ/DQE input skew		tDQ2DQI			ps	31
WCK to DQ/DQE offset voltage variation for Write		tWCK2DQI_ VOLT	–		ps/mV	32
WCK to DQ/DQE offset temperature variation for Write		tWCK2DQI_ TEMP	–		ps/°C	33
DQ/DQE Output Timings						
WCK to DQ/DQE offset for Read		tWCK2DQO			ps	34
DQ/DQE output skew		tDQ2DQO			ps	35

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]		SYMBOL	VALUES		UNIT	NOTES
			MIN	MAX		
DQ/DQE Output Timings (continued)						
WCK to DQ/DQE offset voltage variation for Read		tWCK2DQO_ VOLT	–		ps/mV	36
WCK to DQ/DQE offset temperature variation for Read		tWCK2DQO_ TEMP	–		ps/°C	37
ERR Signal Timings						
Read command to severity output delay		tSEV2ERR			nCK4	38
Internal ERR output delay with variable CAPARERR latency		tWCK2ERRINT			ns	39
WCK to ERR output delay		tWCK2ERRO			ns	40
WCK to ERR output delay voltage variation		tWCK2ERRO_ VOLT	–		ps/mV	41
WCK to ERR output delay temperature variation		tWCK2ERRO_ TEMP	–		ps/°C	42
Interface Training Timings						
CA training entry to first valid sample delay		tCATE	–		ns	
CA training entry to RCK on delay		tCATE2RCK	–	tCATE(max) + tADR(max) - tCK4	ns	
CA sampling from CK4_0 (int) to data out delay		tADR			ns	
CATX to CSP command delay		tCATX	MAX(10, 10 × tCK4)	–	ns	
CA[4:0] high following DQE transition		tDQE_POST	MAX(10, 10 × tCK4)	–	ns	
RCK rising to DQ valid window start delay		tCAT_RCK2DQ			ps	
DQ to DQ output skew in CA training		tCAT_DQ2DQ			ps	43
After CATX, RCK asynchronous restart when in RCK always on mode		tRCK_AON_CATX	–		ns	82
Waiting time from CATX, Sleep exit with SLX2CAT = 1_B or Self Refresh Sleep exit when SRSLX2CAT = 1_B to ERR output = 0 level		tCSP_ACK_PRE		–	ns	
Waiting time from successful CSP command to ERR output = +1 level		tCSP_ACK_POST		–	ns	
Waiting time from last CA bus stress pattern until CSP command		tPRECSP_NOP2		–	nCK4	
Waiting time from a CA parity error to a CSP command that the DRAM requires to guarantee the recognition of the command		tCAPAR2CSP	CAPAR2ERR + 2	–	nCK4	44
CSP command to valid command delay	CAPARBLK disabled	tCSP_POST		–	ns	45
	CAPARBLK enabled			–	ns	
Duty cycle corrector (DCC) start to hold MRS command delay		tDCC		–	nCK4 or ns	46

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]	SYMBOL	VALUES		UNIT	NOTES
		MIN	MAX		
Interface Training Timings (continued)					
Stop CAOSC Interval Oscillator MRS to CAOSC_COUT_VALUE Info Read command delay	tCAOSCO	MAX(40, 8 × tCK4)	–	ns	
Stop CAOSC MRS to Start CAOSC MRS command delay	tCAOSCINT	MAX(40, 8 × tCK4)	–	ns	
Info Mode Timings					
Info Read to Info Read command delay different info register address	tIRD2IRD		–	nCK4	
Info Read to Info Read command delay same info register address	tIRD2IRDSA	BL/8	–	nCK4	47, 48
Row Access Timings					
Activate to Activate or Refresh command period	tRC	tRAS(min) + tRP(min)	–	ns	49
Activate to Precharge command period	tRAS		9 × tREFI(max)	ns	50
Activate to Read command delay	tRCDRD		–	ns	
Activate to Write command delay	tRCDWR		–	ns	
Activate to LDFF command delay	tRCDLTR		–	ns	51
Activate to RDTR command delay	tRCDRTR		–	ns	51
Activate to WRTR command delay	tRCDWTR		–	ns	51
Activate bank A to Activate bank B command delay	tRRD		–	ns	
Per-Bank Refresh to Activate, Per-Bank Refresh or LDFF command delay	tRREFD		–	ns	
LDFF to Activate, PREab, PREpb, REFab (TR= H), REFpb command delay	tLTRR	tRREFD(min)	–	ns	
LDFF to LDFF command cycle time	tLTLTR	4	–	nCK4	
LDFF31 to RDTR command delay	tLTRTR		–	nCK4	55
PREpb to Activate (different bank), REFpb (different bank) or Mode Register Set command delay	tRPD	tPPD	–	ns	
Read to Precharge same bank command delay	tRTPSB		–	ns	
Precharge to Precharge command delay	tPPD	2	–	nCK4	
Row precharge time	tRP		–	ns	
Write recovery time	tWR		–	ns	
Column Access Timings					
Read to Read or Write to Write different bank command delay	tCCD	BL/8	–	nCK4	47, 52, 53
Read to Read or Write to Write same bank command delay	tCCDSB	4	–	nCK4	54

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]	SYMBOL	VALUES		UNIT	NOTES
		MIN	MAX		
Column Access Timings (continued)					
Read to LDFF command delay	tRDTLT	RL + DQERL + BL/8 + 2	–	nCK4	47, 53
RDTR or Info Read to LDFF command delay	tRTRTLT	RL + BL/8 + 2	–	nCK4	47
Write to LDFF command delay	tWRTLT	WL + BL/8 + RU(tWTR(min) / tCK4)	–	nCK4	47, 53
Read to RDTR, RDWTEC or Info Read command delay	tRDRTR	RL + DQERL + BL/8 + 2	–	nCK4	47, 53, 56
RDTR, RDWTEC or Info Read to Read command delay	tRTRRD	RL + BL/8 + 2	–	nCK4	47, 53, 56
RDTR to RDWTEC or Info Read, or RDWTEC to RDTR or Info Read, or Info Read to RDTR or RDWTEC command delay	tRTRRTR	RL + BL/8 + 2	–	nCK4	47, 56
WRTR to RDTR, RDWTEC, Info Read or Read command delay	tWTRTR	WL + BL/8 + 2	–	nCK4	47, 53, 56
Write to WRTR command delay	tWRWTR	WL + BL/8 + 2	–	nCK4	47, 53
Write to RDTR, RDWTEC or Info Read command delay	tWRRTR	WL + BL/8 + RU(tWTR(min) / tCK4)	–	nCK4	47, 53, 56, 57
WRTR to Write command delay	tWTRWR	WL + BL/8 + 2	–	nCK4	47, 53, 58
Internal Write to Read different bank command delay	tWTR		–	ns	59
Internal Write to Read same bank command delay	tWTRSB		–	ns	60
Read to Write or WRTR command delay	tRTW	RL + DQERL + BL/8 + 3 - WL + RU(tBTT / tCK4)	–	nCK4	47, 53, 61
RDTR, RDWTEC or Info Read to Write or WRTR command delay	tRTRWR	RL + BL/8 + 3 - WL + RU(tBTT / tCK4)	–	nCK4	47, 53, 61
Bus turnaround time	tBTT	tWCK2DQO(max) - tWCK2DQI(min)		ns	61
Read, RDTR, RDWTEC or Info Read to REFab command delay	tRDREFab	RL + DQERL + BL/8 + 2	–	nCK4	47, 53, 56, 62
Refresh and Refresh Management Timings					
Average periodic refresh interval with REFab command	tREFI	–	1.9	µs	
Average periodic refresh interval with REFpb command	tREFIPB	–	tREFI(max) / 16	µs	
REFab command period	tRFCab		–	ns	63
REFpb command period	tRFCpb		–	ns	
RFMab command period	tRFMab		–	ns	64
RFMpb command period	tRFMpb		–	ns	64
DRFM command cycle time per row	tRRF	60	–	ns	
DRFM command duration	tDRFM	2 × BRC × tRRF(min)	–	ns	65

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]		SYMBOL	VALUES		UNIT	NOTES
			MIN	MAX		
Refresh and Refresh Management Timings (continued)						
Minimum time in Sleep Mode for per-bank RAA count to be reset to 0		tRAASRF		–	ns	
Minimum rate of REFab commands for impedance calibration updates		tABREF	1	–	ms	66
REFab (TR=H) to Mode Register Set command delay		tREFMRS	MAX (tKO, tMRD	–	ns	
REFab (TR=H) to LDFF command delay		tREFLDFF	MAX (tKO, tRREFD)	–	ns	
REFab to next valid command delay		tKO		–	ns	67, 84
Power-Down Timings						
Power-Down entry to exit time		tPD		9 × tREFI(max)	ns	
Power-Down exit time		tXP		–	nCK4	
Activate to Power-Down Entry command delay		tACTPDE	2	–	nCK4	
Precharge to Power-Down Entry command delay		tPREPDE	2	–	nCK4	
REFpb, RFMab or RFMpb to Power-Down Entry command delay		tREFPDE	2	–	nCK4	
REFab with TR = L to Power-Down Entry command delay and Auto ECS enabled (MR22 OP11 = 1_B)		tREF_ECS_PDE		–	nCK4	
DRFM to Power-Down Entry command delay		tDRFMPDE	tDRFM(min)	–	ns	65
MRS to Power-Down Entry command delay		tMRSPDE	tMOD(min)	–	nCK4	
Read to Power-Down Entry command delay		tRDPDE	RL + DQERL + BL/8 + 2	–	nCK4	47
RDTR, RDWTEC or Info Read to Power-Down Entry command delay		tRTRPDE	RL + BL/8 + 2		nCK4	47
Write or WRTR to Power-Down Entry command delay		tWRPDE	WL + BL/8 + 2 + WRCRC2ERR	–	nCK4	47
Self Refresh and Sleep Mode Timings						
RNOP2/CNOP2 commands required upon self refresh or sleep mode entry	CAPAR off or CAPAR on with CAPAR2ERR = 1-15	tCPDED	MAX (10, CAPAR2ERR + 2)	–	nCK4	68
	CAPAR on with CAPAR2ERR = 0		MAX (10, RU(tWCK2ERRINT(max) / tCK4) + 2)	–	nCK4	
Self Refresh entry to exit time		tSR		–	ns	
Self Refresh exit to valid command delay		tXS	tRFCab(min)	–	ns	
Self Refresh entry to LDFF command delay		tSRE2LDFF		–	ns	
Self Refresh entry to RDTR command delay		tSRE2RDTR		–	ns	
Self Refresh entry to WRTR command delay		tSRE2WRTR		–	ns	

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]		SYMBOL	VALUES		UNIT	NOTES
			MIN	MAX		
Self Refresh and Sleep Mode Timings (continued)						
Sleep mode entry to exit time		tSLEEP		9 × tREFI(max)	ns	
Hibernate Self Refresh sleep mode entry to exit time		tHSLEEP		–	µs	
Self Refresh sleep mode entry to exit time		tSRFSLP		–	ns	
CA0 Low pulse width to exit from sleep mode		tSLX		–	ns	
Additional pause upon sleep mode exit after a WCK frequency change		tSLXFC		–	ns	69, 70
Self Refresh Sleep exit to CSP command delay when ECS operation is enabled (MR22 OP11 =1_B)		tSLX_ECS		–	ns	
Sleep mode exit when SLX2CAT = 1_B or Self Refresh Sleep mode exit when SRSLX2CAT = 1_B to CA training entry delay		tSLX_CAT	See *TABLE 145*		Ns	70, 71
Sleep mode exit when SLX2CAT = 1_B or Self Refresh Sleep mode exit when SRSLX2CAT = 1_B to RCK on delay		tSLXCAT2RCK	–	tSLX_CAT(max) + tADR(max) - tCK4	ns	
Sleep mode exit to CSP command delay		tCSP_PRE	See *TABLE 145*		ns	70, 71
After SLX, RCK asynchronous restart when in RCK always on mode	No frequency change	tRCK_AON_SLX	–		ns	83
	Frequency change		–		ns	
Precharge to Sleep Mode Entry command delay		tPRESLE	RU(tRP(min) / tCK4) + 2	–	nCK4	
REFab to Sleep Mode Entry command delay		tREFabSLE	RU(tRFCab(min) / tCK4) + 2	–	nCK4	
REFpb to Sleep Mode Entry command delay		tREFpbSLE	RU(tRFCpb(min) / tCK4) + 2	–	nCK4	
MRS to Sleep Mode Entry command delay		tMRSSLE	tMOD(min)	–	nCK4	
RDTR, RDWTEC or Info Read to Sleep Mode Entry command delay		tRTRSLE	RL + BL/8 + 2		nCK4	47
Read with AP to Sleep Mode Entry command delay		tRDASLE	MAX(RL + DQERL + BL/8 + 4, RTPSB)	–	nCK4	47
Write with AP to Sleep Mode Entry command delay		tWRASLE	WL + BL/8 + 2 + MAX(WRCRC2ERR, WR)	–	nCK4	47
WRTR to Sleep Mode Entry command delay		tWTRSLE	WL + BL/8 + 2 + WRCRC2ERR	–	nCK4	47
CA0 Low pulse width to exit from Hibernate Self Refresh Sleep mode		tXHP	10	–	ns	
Exit Hibernate Self Refresh Sleep mode to Self Refresh Sleep mode exit delay		tXSH	500	–	µs	
Mode Register Set Timings						
Mode Register Set command period		tMRD		–	nCK4	

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]		SYMBOL	VALUES		UNIT	NOTES
			MIN	MAX		
Mode Register Set Timings						
Mode Register Set command update delay		tMOD		–	nCK4	72
Mode Register 15 Set command update delay		tMOD15	2 × tMOD(min)	–	nCK4	73
Mode Register Set command update delay for RCK		tMODRCK		–	nCK4	74
Read to Mode Register Set command delay		tRDMRS	RL + DQERL + BL/8 + 2	–	nCK4	47
RDTR, RDWTEC or Info Read to Mode Register Set command delay		tRTRMRS	RL + BL/8 + 2	–	nCK4	47
Write or WRTR to Mode Register Set command delay		tWRMRS	WL + BL/8 + WRCRC2ERR + 2	–	nCK4	47
Activate, REFpb or LDFF to Mode Register Set command delay		tACTMRS	tMRD(min)	–	nCK4	75
RCKSTRT to Mode Register Set command delay		tRCKSTRT2MRS	RCKEN + 2	–	nCK4	85
RCKSTOP to Mode Register Set command delay (except MR9 and 12)		tRCKSTOP2MRS	RCKSTOP_LAT + 2	–	nCK4	86
VREFCA reference voltage change settling time		tVREFCA		–	ns	76
VREFD reference voltage change settling time		tVREFD		–	ns	77
Error Check and Scrub (ECS) Timings						
Average periodic interval of ECS events	4Gb/channel	tECSint	5.15		ms	78
	6Gb/channel		3.43		ms	
	8Gb/channel		2.57		ms	
	12Gb/channel		1.72		ms	
	16Gb/channel		1.29		ms	
	24Gb/channel		0.86		ms	
	32Gb/channel		0.64		ms	
Time to complete one ECS operation		tECSC	tRFCab(min)	–	ns	
ECS logs valid after Self Refresh Sleep exit to Self Refresh TR=H (SRSLX2CAT MR0 OP6=1_B) when Auto ECS enabled (MR22 OP11 = 1_B)		tSRSX_ECSLOG_UPD		–	ns	
Post Package Repair Timings						
Post package repair (hPPR) programming time		tPGM	2	–	s	
Post package repair (hPPR) exit time		tPGM_EXIT	tRP(min)	–	ns	
Post package repair (hPPR) address settling time		tPGMPST	500	–	µs	
Miscellaneous Timings						
DVS voltage stabilization time		tVS		–	µs	79

Table 141 — AC Timings (cont'd)

PARAMETER [1,2,3]	SYMBOL	VALUES		UNIT	NOTES
		MIN	MAX		
Miscellaneous Timings (continued)					
DQ map mode on to stable output delay	tDQMAPON			ns	80
DQ map mode off delay	tDQMAPOFF	–		ns	80
ODT disable time before the start of the Read, RDTR, Info Read and RDWTEC DQ and DQE burst	tODT_off			ns	
ODT enable time after the end of the Read, RDTR, Info Read and RDWTEC DQ and DQE burst	tODT_on			ns	

NOTES for Table 141 - AC Timings

1. All parameters assume proper device initialization.
2. Tests for AC timing may be conducted at nominal supply voltage levels, but the related specifications and device operation are guaranteed for the full voltage and temperature range specified.
3. Values in fields that are left blank are vendor specific. The vendor datasheet shall be consulted for actual values.
4. The average WCK clock high and low width are averaged over 500 WCK cycles.

$$tWCK(avg) = \left(\sum_{j=1}^{N} tWCKj\right)/N$$

$$where\ N = 500$$

5. The absolute WCK clock high and low width applies to each individual WCK clock cycle.
6. CK4 is a DRAM internal clock that operates at a divided-by-4 WCK frequency. All latencies and other timings are referenced to CK4. As CK4 is an internal clock, there is no definition for CK4 timings. However, the accumulated jitter over 4-contiguous WCK cycles is the significant contributor to CK4 timing uncertainty. Consult the vendor for more details on the internal CK4.
7. Parameter fWCKNRZ applies when NRZ mode is selected in MR0 OP8.
8. Parameter fWCKSERCK applies when the RCK type is set to single-ended in MR9 OP2. The maximum value is vendor specific and may be less than fWCK(max).
9. Parameter fWCKHRCK applies when the RCK level is set to half swing in MR9 OP3. The maximum value is vendor specific and may be less than fWCK(max).
10. Parameter fWCKPGM applies when post package repair is enabled in MR31 OP11.
11. Parameter tWCK2CA defines the WCK-to-CA offset range for CA inputs. The minimum and maximum values are positive numbers and cover all delay variation over PVT. Command/address (CA) bus training is required to determine the actual tWCK2CA offset for reliable device operation.
12. Parameter tCA2CA defines the maximum skew among the CA[4:0] inputs under worst case conditions. Parameter tWCK2CA defines the mean value of the earliest and latest CA input, tCA2CA(min) the negative offset to tWCK2CA for the earliest CA input and tCA2CA(max) the positive offset to tWCK2CA for the latest CA input.
13. The CA input pulse width is a design target. The value will be characterized but not tested on each device.
14. Parameter tWCK2CA_VOLT defines the variation of tWCK2CA from changes of the VDDQ supply voltage. The parameter is determined for the worst-case process corner and maximum operating temperature; the value will be characterized but not tested on each device.
15. Parameter tWCK2CA_TEMP defines the variation of tWCK2CA from changes of the operating temperature. The parameter is determined for the worst-case process corner and nominal VDDQ supply voltage; the value will be characterized but not tested on each device.
16. RCK_t and RCK_c are both single-ended output drivers. If enabled, they can be configured to drive a differential clock pattern on both outputs, or a single-ended clock pattern on RCK_t only.
17. The support of the DCC is optional.
18. Parameter tWCK2RCK defines the WCK-to-RCK_t/_c offset range. The minimum and maximum values are positive numbers and cover all delay variation over PVT. RCK must be enabled to determine the actual tWCK2RCK offset for reliable device operation.
19. Parameter tWCK2RCK_VOLT defines the variation of tWCK2RCK from changes of the VDDQ supply voltage. The parameter is determined for the worst-case process corner and maximum operating temperature; the value will be characterized but not tested on each device.
20. Parameter tWCK2RCK_TEMP defines the variation of tWCK2RCK from changes of the operating temperature. The parameter is determined for the worst-case process corner and nominal VDDQ supply voltage; the value will be characterized but not tested on each device.
21. Parameter tRCK2LZ defines the time period for RCK to transition from High-Z to driving static H/L levels.
22. Parameter tRCK2HZ defines the time period for RCK to transition from driving static H/L levels to High-Z.
23. The RCKEN latency must be set to a value large enough to satisfy the tRCK_ST timing.

NOTES for Table 141 - AC Timings

24. Parameter tRCK_HS equals RL - RCKEN - RCK_LS when RCKMODE is set to Start with Read commands (MR9 OP[1:0]=01_B); the parameter equals RL + tRCKSTRT2RD - RCKEN - RCK_LS when RCKMODE is set to Start with RCKSTRT command (MR9 OP[1:0]=10_B).
25. Parameter tRCKST2SP applies when RCKMODE is set to Start with RCKSTRT command (MR9 OP[1:0]=10_B).
26. Parameter tRD2RCKSTOP applies when RCKMODE is set to Start with Read command (MR9 OP[1:0]=01_B). Read commands in this context comprise RD, RDA, IRD, RDTR and RDWTEC commands.
27. Parameter tRCKSTRT2RD applies when RCKMODE is set to Start with RCKSTRT command (MR9 OP[1:0]=10_B). Read commands in this context comprise RD, RDA, IRD, RDTR and RDWTEC commands.
28. Parameter tRCKSP2ST applies when RCKMODE is set to Start with Read command (MR9 OP[1:0]=01_B) or Start with RCKSTRT command (MR9 OP[1:0]=10_B). Read commands in this context comprise RD, RDA, IRD, RDTR and RDWTEC commands.
29. Parameter tRCKSTOP2MR9_12 must be observed when the RCK mode in MR9 OP[1:0] is set to either Start with Read command or Start with RCKSTRT command and RCK is running. In both cases RCK must be stopped via the RCKSTOP command before changing RCK related settings in MR9 or MR12.
30. Parameter tWCK2DQI defines the WCK-to-DQ/DQE offset range for write data. The minimum and maximum values are positive numbers and cover all delay variation over PVT. Write data training is required to determine the actual tWCK2DQI offset for reliable device operation.
31. Parameter tDQ2DQI defines the maximum skew among all DQ/DQE inputs under worst case conditions. Parameter tWCK2DQI defines the mean value of the earliest and latest DQ/DQE input, tDQ2DQI(min) the negative offset to tWCK2DQI for the earliest DQ/DQE input and tDQ2DQI(max) the positive offset to tWCK2DQI for the latest DQ/DQE input.
32. Parameter tWCK2DQI_VOLT defines the variation of tWCK2DQI from changes of the VDDQ supply voltage. The parameter is determined for the worst-case process corner and maximum operating temperature; the value will be characterized but not tested on each device.
33. Parameter tWCK2DQI_TEMP defines the variation of tWCK2DQI from changes of the operating temperature. The parameter is determined for the worst-case process corner and nominal VDDQ supply voltage; the value will be characterized but not tested on each device.
34. Parameter tWCK2DQO defines the WCK-to-DQ/DQE offset range for read data. The minimum and maximum values are positive numbers and cover all delay variation over PVT. Read data training is required to determine the actual tWCK2DQO offset for reliable device operation.
35. Parameter tDQ2DQO defines the maximum skew among all DQ/DQE outputs under worst case conditions. Parameter tWCK2DQO defines the mean value of the earliest and latest DQ/DQE output, tDQ2DQO(min) the negative offset to tWCK2DQO for the earliest DQ/DQE output and tDQ2DQO(max) the positive offset to tWCK2DQO for the latest DQ/DQE output.
36. Parameter tWCK2DQO_VOLT defines the variation of tWCK2DQO from changes of the VDDQ supply voltage. The parameter is determined for the worst-case process corner and maximum operating temperature; the value will be characterized but not tested on each device.
37. Parameter tWCK2DQO_TEMP defines the variation of tWCK2DQO from changes of the operating temperature. The parameter is determined for the worst-case process corner and nominal VDDQ supply voltage; the value will be characterized but not tested on each device.
38. Parameter tSEV2ERR defines the internal latency from a Read command for reporting severity errors on the ERR signal when SEV2ERR is enabled in MR5 OP9.
39. Parameter tWCK2ERRINT defines the internal WCK-to-CA parity error delay range with variable CAPARERR latency. The support of the variable latency is vendor specific.
40. Parameter tWCK2ERRO defines the WCK-to-ERR offset range. The minimum and maximum values are positive numbers and cover all delay variation over PVT. ERR pin training is required to determine the actual tWCK2ERRO offset for reliable device operation.
41. Parameter tWCK2ERRO_VOLT defines the variation of tWCK2ERRO from changes of the VDDQ supply voltage. The parameter is determined for the worst-case process corner and maximum operating temperature; the value will be characterized but not tested on each device.
42. Parameter tWCK2ERRO_TEMP defines the variation of tWCK2ERRO from changes of the operating temperature. The parameter is determined for the worst-case process corner and nominal VDDQ supply voltage; the value will be characterized but not tested on each device.
43. Parameter tCAT_DQ2DQO defines the maximum skew among all DQ outputs under worst case conditions in CA training mode. Parameter tADR defines the mean value of the earliest and latest DQ output, tCAT_DQ2DQO(min) the negative offset to tADR for the earliest DQ output and tCAT_DQ2DQO(max) the positive offset to tADR for the latest DQ output.
44. The DRAM may not recognize a CSP command for tCAPAR2CSP time after a CA parity error is found.
45. RNOP2 and CNOP2 commands must be issued during the tCSP_POST period.
46. Parameter tDCC defines the minimum number of CK4 clock cycles required between the MRS command that starts the DCC and a subsequent MRS command that sets the DCC to hold state.
47. BL is 16 in PAM3 mode and 32 in NRZ mode.
48. The tIRD2IRDSA delay for back-to-back IRD commands to the same Info Register Address (IRA) permits a steady data output for longer than a single burst. Back-to-back IRD commands to the same Info Register Address (IRA) are supported for IRAs 0 to 7 and 32 to 39 only.
49. The number of CK4 cycles required to meet tRC is the number of CK4 cycles required for tRAS and the number of CK4 cycles required for tRP.
50. For Reads and Write with Auto Precharge enabled the internal precharge will be hold off until tRAS(min) has been satisfied. The number of CK4 clock cycles to meet tRAS(min) is programmed in MR3 OP[6:0].
51. Parameters tRCDLTR, tRCDRTR, and tRCDWTR apply for a first ACT command issued in bank idle state; the parameters shall be ignored for subsequent ACT commands issued in bank active state.
52. Parameter tCCD applies when consecutive Read or Write commands access different banks. The parameter also applies to gapless consecutive RDTR or WRTR commands.
53. Read commands comprise RD and RDA commands. Write commands comprise WR and WRA commands.
54. Parameter tCCDSB applies when consecutive Read or Write commands access the same bank.
55. Parameter tLTRTR applies when data training is set to FIFO mode in MR23 OP0.
56. The RDWTEC command is used in Write training to read out write training burst error counter values. The command is only valid when LFSR data training is enabled in MR23 OP0.
57. Parameter tWRRTR applies when data training is set to LFSR mode in MR23 OP0. In FIFO training mode it is required to load the FIFO via LDFF or WRTR commands before issuing RDTR commands.

NOTES for Table 141 - AC Timings

58. Parameter tWTRWR applies when data training is set to LFSR mode in MR23 OP0. In FIFO training mode a WRTR command must be followed by a RDTR command.
59. Parameter tWTR applies when consecutive commands access different banks.
60. Parameter tWTRSB applies when consecutive commands access the same bank. The parameters also applies when ECC engine test mode is enabled.
61. The difference between the tWCK2DQO and tWCK2DQI timings must be considered when calculating the minimum system bus turnaround time tBTT. Any signal flight time between host and DRAM is not included in the formula.
62. RCK and data output timings may not be guaranteed during tKO after a REFab command which can impact the last read transaction. In that case, a REFab command shall only be issued when all previous read operations have completed which is when the last data element including CRC has been transmitted on the data outputs.
63. A maximum of 8 consecutive REFab commands can be posted to a channel, meaning that the maximum absolute interval between any REFab command and the next REFab command is 9 × tREFI.
64. Parameters tRFMab and tRFMpb apply only to devices that require the use of Refresh Management (RFM), or when Adaptive Refresh Management (ARFM) is enabled in MR8 OP[1:0].
65. Refer to the *DIRECTED REFRESH MANAGEMENT (DRFM)* section for details.
66. Parameter tABREF is relevant only when array refresh is normally performed using the REFpb command. REFab commands must be issued at a minimum rate of tABREF to allow impedance updates from the auto-calibration engine to occur.
67. Parameter tKO must be observed regardless whether auto-calibration is enabled or disabled in MR5 OP11.
68. CAPARERR is programmed in MR15 OP[11:8]. With CAPAR disabled, CAPAR2ERR is assumed to be 0 nCK4 in the equation, resulting in 10 nCK4.
69. Parameter tSLXFC must be observed following tSLX if the WCK frequency in MR12 OP[8:4] has changed. The value is vendor specific, and tSLXFC can be ignored in case a value of 0ns is specified.
70. RNOP2 and CNOP2 commands must be issued during tSLXFC, tSLX_CAT and tCSP_PRE periods.
71. Values for parameters tSLX_CAT and/or tCSP_PRE may depend on the WCK frequency as set in MR12 OP[8:4]. Refer to the vendor datasheet for details.
72. Parameter tMOD applies to MR10 to MR14 only when the FD_FLAG bit in MR0 OP11 is set to 1_B.
73. Parameter tMOD15 applies instead of tMOD only when the FD_FLAG bit in MR0 OP11 is set to 1_B. RNOP2 and CNOP2 are required during the tMOD15 period.
74. Parameter tMODRCK applies to RCK_t and RCK_c outputs instead of tMOD when RCKMODE in MR9 OP[1:0] is either changed from 00_B (disable) to 11_B (always on) or back to 00_B (disable).
75. Refer to the *MODE REGISTERS* section for mode registers that are allowed to be modified in bank active state.
76. Parameter tVREFCA applies when the VREFCA level has been changed in MR14 OP[6:0] or when the Half VREFCA bit has been changed in MR14 OP11. tVREFCA is a constant value for the device and is referenced from the MRS command to when the 90% level of the delta between old and new VREFC voltage has been reached.
77. Parameter tVREFD applies when the VREFDL and VREFDH levels have been changed in MR16 OP[6:0] and MR17 OP[6:0], respectively, or when the signaling level (PAM3 / NRZ) has been changed in MR0 OP8. tVREFD is a constant value for the device and is referenced from the MRS command to when the 90% level of the delta between old and new VREFD voltage has been reached.
78. Providing ECS events at tECSint rate ensure that the error check and scrub of the whole memory array is completed with 24 hours. Note that tECSint is density dependent and that for a given device the density per channel is different for 4-channel and 2-channel modes.
79. Parameter tVS must be observed when either the VDD or VDDQ supply voltage or both supply voltages have changed.
80. Parameter tDQMAPON defines the delay between the MRS command that enables DQ map mode and a valid read-out on the DQ signals.
81. Parameter tDQMAPOFF defines the delay between the MRS command that disables DQ map mode and when the DQ signals have returned to their default state.
82. tRCK_AON_CATX is a vendor specific max but must not be greater than tCATX + tCSP_POST.
83. tRCK_AON_SLX is a vendor specific max but must not be greater than tSLX + tCSP_PRE + tCSP_POST when exiting Sleep or exiting Self Refresh Sleep with no frequency change, and tSLX + tSLXFC + tCSP_PRE + tCSP_POST when exiting Self Refresh Sleep with a frequency change.
84. The timing for REFab (TR=H) to Mode Register Set command delay is tREFMRS and REFab (TR=H) to LDFF command delay is tREFLDFF.
85. RCKSTRT to MR9 or MR12 is illegal. Before issue MRS to MR9 or MR12 command, RCK must be properly stopped.
86. RCKSTOP to MR9 or MR12 programming delay is tRCKSTOP2MR9_12
87. The Min and Max values of the tWCK parameter are average values that define the operating frequency range of the device. For the Min/Max range of the tWCK absolute values please follow the indications in the AC Operating Conditions section.

10 AC Timings (cont'd)

Table 142 — Latency Timings (Part 1) for GDDR7 DRAMs Supporting Implicit CAPARBLK_LAT

Data Rate	fWCK MR12 OP[8:4]		RL MR1 OP[5:0]				WL MR2 OP[3:0]				RCKEN MR9 OP[9:5]				RCKSTOP_LAT MR9 OP[11:10]			
			CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B		CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B		CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B		CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B	
[Gbps]	[GHz]	Code	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
	NRZ	00000																
2	0.5	00001																
4	1.0	00010																
6	1.5	00011																
8	2.0	00100																
10	2.5	00101																
12	3.0	00110																
14	3.5	00111																
16	4.0	01000																
18	4.5	01001																
20	5.0	01010																
22	5.5	01011																
24	6.0	01100																
26	6.5	01101																
28	7.0	01110																
30	7.5	01111																
32	8.0	10000																
34	8.5	10001																
36	9.0	10010																
38	9.5	10011																
40	10.0	10100																
42	10.5	10101																
44	11.0	10110																
46	11.5	10111																
48	12.0	11000																

NOTE 1 Refer to the *MODE REGISTERS* section for the definition and use of the related register fields.

10 AC Timings (cont'd)

Table 143 — Latency Timings (Part 1) for GDDR7 DRAMs Supporting Explicit CAPARBLK_LAT

Data Rate	fWCK MR12 OP[8:4]		CAPARBLK_LAT MR15 OP[5:3]		Read Latency MR1 OP[5:0] + MR15 OP[5:3]				Write Latency MR2 OP[3:0] + MR15 OP[5:3]				RCK Enable Latency MR9 OP[9:5] + MR15 OP[5:3]				RCK Stop Latency MR9 OP[11:10] + MR15 OP[5:3]			
			CAPAR BLK Off	CAPAR BLK ON	CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B		CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B		CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B		CAPARBLK disabled MR15 OP1 = 0_B		CAPARBLK enabled MR15 OP1 = 1_B	
[Gbps]	[GHz]	Code			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
	NRZ	00000	0																	
2	0.5	00001	0																	
4	1.0	00010	0																	
6	1.5	00011	0																	
8	2.0	00100	0																	
10	2.5	00101	0																	
12	3.0	00110	0																	
14	3.5	00111	0																	
16	4.0	01000	0																	
18	4.5	01001	0																	
20	5.0	01010	0																	
22	5.5	01011	0																	
24	6.0	01100	0																	
26	6.5	01101	0																	
28	7.0	01110	0																	
30	7.5	01111	0																	
32	8.0	10000	0																	
34	8.5	10001	0																	
36	9.0	10010	0																	
38	9.5	10011	0																	
40	10.0	10100	0																	
42	10.5	10101	0																	
44	11.0	10110	0																	
46	11.5	10111	0																	
48	12.0	11000	0																	

NOTE 1 Refer to the *MODE REGISTERS* section for the definition and use of the related register fields.

10 AC Timings (cont'd)

Table 144 — Latency Timings (Part 2)

Data Rate	fWCK MR12 OP[8:4]		DQERL MR1 OP[11:8]		WRCRC2ERR MR2 OP[11:7]		RAS MR3 OP[6:0]		WR MR4 OP[6:0]		RTP MR4 OP[11:8]		CAPAR2ERR MR15 OP[11:8]		CAPAR2ERR MR15 OP[11:8] = 0	
[Gbps]	[GHz]	Code	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
	NRZ	00000														
2	0.5	00001														
4	1.0	00010														
6	1.5	00011														
8	2.0	00100														
10	2.5	00101														
12	3.0	00110														
14	3.5	00111														
16	4.0	01000														
18	4.5	01001														
20	5.0	01010														
22	5.5	01011														
24	6.0	01100														
26	6.5	01101														
28	7.0	01110														
30	7.5	01111														
32	8.0	10000														
34	8.5	10001														
36	9.0	10010														
38	9.5	10011														
40	10.0	10100														
42	10.5	10101														
44	11.0	10110														
46	11.5	10111														
48	12.0	11000														

NOTE 1 Refer to the *MODE REGISTERS* section for the definition and use of the related register fields.
NOTE 2 Timings in this table are the same for CAPARBLK on and off cases.
NOTE 3 For CAPAR2ERR, GDDR7 SGRAMs may support the variable latency setting (MR15 OP[11:8] = 0000_B), one or more fixed latency settings (MR15 OP[11:8] ≠ 0000_B), or both.

10 AC Timings (cont'd)

Table 145 — Frequency Dependent AC Timings

Data Rate	fWCK MR12 OP[8:4]		tSLX_CAT		tCSP_PRE		
[Gbps]	[GHz]	Code	Min	Max	Min	Max	Unit
	NRZ	00000	–			–	ns
2	0.5	00001					ns
4	1.0	00010					ns
6	1.5	00011					ns
8	2.0	00100					ns
10	2.5	00101					ns
12	3.0	00110					ns
14	3.5	00111					ns
16	4.0	01000					ns
18	4.5	01001					ns
20	5.0	01010					ns
22	5.5	01011					ns
24	6.0	01100					ns
26	6.5	01101					ns
28	7.0	01110					ns
30	7.5	01111					ns
32	8.0	10000					ns
34	8.5	10001					ns
36	9.0	10010					ns
38	9.5	10011					ns
40	10.0	10100					ns
42	10.5	10101					ns
44	11.0	10110					ns
46	11.5	10111					ns
48	12.0	11000					ns

NOTE 1 Refer to the *AC TIMINGS* table for related parameter definitions.
NOTE 2 Vendors may specify a single value that applies at all WCK frequencies.

11 Package Specification

11.1 Signals

Table 146 — Ball-Out Description

SYMBOL	TYPE	SIGNALING	DESCRIPTION
WCK_t_[A:D], WCK_c_[A:D]	Input	NRZ	System Clock: WCK_t and WCK_c is a differential clock input used for command and address (CA) capture, write data capture and for driving read data. All latencies will be referenced to an internally divided-by-4 WCK (CK4) clock.
RCK_t_[A:D], RCK_c_[A:D]	Output	NRZ	Read Strobe: RCK_t and RCK_c is a differential strobe output driven along with read data. RCK can also be configured to be single-ended or off.
CA[4:0]_[A:D]	Input	NRZ	Command and Address (CA) Inputs: the CA inputs receive command, address, CA parity, CA bus inversion, mode register opcodes or other information that control the operation of the device. Refer to the Command Truth Tables for details.
DQp[9:0]_[A:D]	I/O	PAM3 / NRZ	Data Input/Output: 10-bit data bus. In PAM3 mode, write and read data are transferred on DQ [9:0]; in NRZ mode, DQ [9:8] are disabled and data are transferred on DQ [7:0] only. DQ[9:0] are enabled as outputs in DQ Mapping and CA Bus Training modes regardless of selected PAM3 or NRZ signaling.
DQE_[A:D]	I/O	PAM3 / NRZ	In PAM3 mode, remaining write and read data are transferred on DQE. In both PAM3 and NRZ modes, DQE also transmits • the calculated CRC checksum when CRC is enabled. • Poison flag on writes and reads (if enabled) • Severity flag on reads (if enabled). DQE is also used as input during CA Bus Training.
ERR_[A:D]	Output	PAM3	Error: the ERR output signals CA parity errors and write data CRC errors back to the host. In a special mode (see note 4), ERR can also be configured to transmit the flag related to error severity reporting on reads. The output drives PAM3 levels regardless of the signaling scheme programmed for data.
V_{DDQ}	Supply	-	I/O Power Supply. Isolated on the die for improved noise immunity.
V_{DD}	Supply	-	Power Supply
V_{SS}	Supply	-	Ground
V_{PP}	Supply	-	Pump Voltage
ZQ_AB, ZQ_CD	Reference	-	External Reference for autocalibration. ZQ_AB is associated with channels A and B, and ZQ_CD is associated with channels C and D
RESET_n	Input	NRZ	RESET_n Low asynchronously initiates a full chip reset. With RESET_n Low, all ODTs are disabled. A full chip reset may be performed at any time by pulling RESET_n Low.

NOTE 1 Indices "_A", "_B", "_C" or "_D" represent the channel indicator. Signal names including the channel indicator are used whenever more than one channel of the device is referenced, as e.g., with the ball-out. The channel indicator is omitted whenever features and functions common to all channels are described.

NOTE 2 Refer to *TABLE 147* for the mapping of the physical DQp[9:0] pins to logical DQ[9:0] signals.

NOTE 3 The RESET_n signal is common to all channels.

NOTE 4 This special mode is controlled by the SEV2ERR bit in MR5 OP9. See *MODE REGISTERS* section for details.

11.2 Ball-Out

GDDR7 SGRAMs support a vendor defined mapping of logical DQ[9:0] signals to physical package pins. The ball-out as shown in *FIGURE 152* uses the term DQp[9:0]_[A:D] with letter "p" referring to the physical locations of the 10 DQ pins per channel. The correspondence between these pin locations and the logical DQ[9:0] signals will be found in a vendor specific mapping table (*TABLE 147*). It is pointed out that the location of the 10 DQ and their mapping to logical signals is the same for the 4 channels, which preserves the symmetry across the channels.

1	2	3	4	5	6	7		8	9	10	11	12	13	14
VSS	VSS	DQp1_A	VSS	DQE_A	VSS	DQp0_A	A	DQp0_B	VSS	DQE_B	VSS	DQp1_B	VSS	VSS
DQp2_A	VSS	VSS	VDDQ	VSS	VDD	VSS	B	VSS	VDD	VSS	VDDQ	VSS	VSS	DQp2_B
VSS	VSS	DQp3_A	VSS	DQp4_A	VSS	DQp5_A	C	DQp5_B	VSS	DQp4_B	VSS	DQp3_B	VSS	VSS
RCK_c_A	VDDQ	VSS	VDDQ	VSS	VDDQ	VSS	D	VSS	VDDQ	VSS	VDDQ	VSS	VDDQ	RCK_c_B
RCK_t_A	VSS	DQp7_A	VSS	DQp6_A	VSS	WCK_c_A	E	WCK_c_B	VSS	DQp6_B	VSS	DQp7_B	VSS	RCK_t_B
VSS	VDDQ	VSS	VDDQ	VSS	VSS	WCK_t_A	F	WCK_t_B	VSS	VSS	VDDQ	VSS	VDDQ	VSS
DQp8_A	VSS	DQp9_A	VSS	ERR_A	VDDQ	VSS	G	VSS	VDDQ	ERR_B	VSS	DQp9_B	VSS	DQp8_B
VSS	VSS	VSS	VDD	VSS	VDD	CA0_A	H	CA0_B	VDD	VSS	VDD	VSS	VSS	VSS
VPP	CA4_A	VDD	CA3_A	CA2_A	VDD	CA1_A	J	CA1_B	VDD	CA2_B	CA3_B	VDD	CA4_B	VPP
ZQ_AB	RESET_n	VSS	VSS	VDD	VSS	VSS	K	VSS	VSS	VDD	VSS	VSS	VSS	ZQ_CD
VPP	CA4_D	VDD	CA3_D	CA2_D	VDD	CA1_D	L	CA1_C	VDD	CA2_C	CA3_C	VDD	CA4_C	VPP
VSS	VSS	VSS	VDD	VSS	VDD	CA0_D	M	CA0_C	VDD	VSS	VDD	VSS	VSS	VSS
DQp8_D	VSS	DQp9_D	VSS	ERR_D	VDDQ	VSS	N	VSS	VDDQ	ERR_C	VSS	DQp9_C	VSS	DQp8_C
VSS	VDDQ	VSS	VDDQ	VSS	VSS	WCK_t_D	P	WCK_t_C	VSS	VSS	VDDQ	VSS	VDDQ	VSS
RCK_t_D	VSS	DQp7_D	VSS	DQp6_D	VSS	WCK_c_D	R	WCK_c_C	VSS	DQp6_C	VSS	DQp7_C	VSS	RCK_t_C
RCK_c_D	VDDQ	VSS	VDDQ	VSS	VDDQ	VSS	T	VSS	VDDQ	VSS	VDDQ	VSS	VDDQ	RCK_c_C
VSS	VSS	DQp3_D	VSS	DQp4_D	VSS	DQp5_D	U	DQp5_C	VSS	DQp4_C	VSS	DQp3_C	VSS	VSS
DQp2_D	VSS	VSS	VDDQ	VSS	VDD	VSS	V	VSS	VDD	VSS	VDDQ	VSS	VSS	DQp2_C
VSS	VSS	DQp1_D	VSS	DQE_D	VSS	DQp0_D	W	DQp0_C	VSS	DQE_C	VSS	DQp1_C	VSS	VSS

Figure 152 — 266 Ball BGA Ball-Out

11.2 Ball-Out (cont'd)

Table 147 — Example Logical Signal to Physical Pin Mapping for DQ[9:0]

PIN	BALL CHANNEL A	BALL CHANNEL B	BALL CHANNEL C	BALL CHANNEL D	LOGICAL SIGNAL
DQp0	A7	A8	W8	W7	DQ9
DQp1	A3	A12	W12	W3	DQ8
DQp2	B1	B14	V14	V1	DQ7
DQp3	C3	C12	U12	U3	DQ6
DQp4	C5	C10	U10	U5	DQ5
DQp5	C7	C8	U8	U7	DQ0
DQp6	E5	E10	R10	R5	DQ1
DQp7	E3	E12	R12	R3	DQ2
DQp8	G1	G14	N14	N1	DQ3
DQp9	G3	G12	N12	N3	DQ4

NOTE 1 The ball coordinates reflect the fixed locations of the DQp[9:0] pins in channels A to D. The mapping to logical signals as in the "Logical Signal" column is vendor specific; the mapping shown in the table is provided as an example and applies to all 4 channels. Refer to the vendor datasheet for the specific signal mapping.

11.3 DQ Map Mode

The DQ map mode allows the host to electronically retrieve the vendor defined mapping of logical DQ[9:0] signals to physical DQ package balls. This mode is enabled by setting MR29 OP4 to 1_B. While in this mode, the physical DQ that corresponds to the logical DQ as selected by MR29 OP[3:0] drives a static Low (or -1 level in PAM3 mode), while all other DQs drive a static High (or +1 level in PAM3 mode). The complete logical-to-physical DQ mapping will be obtained by selecting each logical DQ one after the other and observing which physical DQ drives a Low. Refer to *TABLE 148* for the pin selection.

A first DQ pin drives the static Low latest a time tDQMAPON(max) after the MRS commands that enables DQ map mode; the value will be held until another DQ is selected by a subsequent MRS command that changes the pin selection in MR29 OP[4:0], or until an MRS command that terminates DQ map mode by setting MR29 OP4 back to 0_B. The DQ pins return to their default state latest by tDQMAPOFF(max). While in DQ map mode, DQE state is vendor specific. However, it should not be driven to Low in NRZ mode and to -1 or 0 level in PAM3 mode.

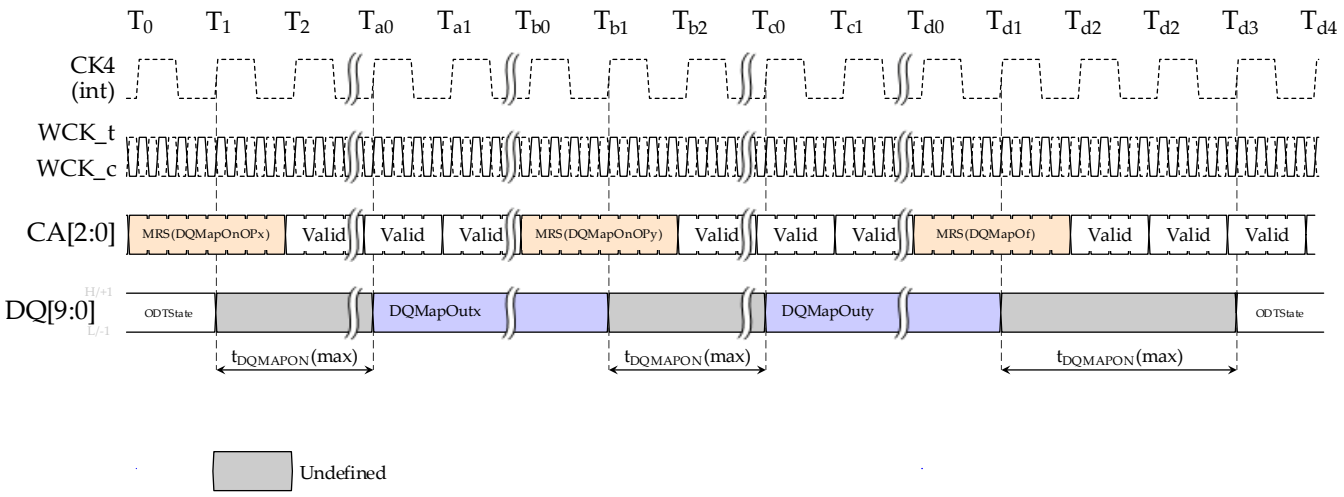
Only NOP commands and MRS commands to MR29 are allowed while in DQ map mode, and RCK will retain its state (on or off) as set prior to enabling DQ map mode. It is pointed out that no phase relationship is specified between RCK and DQs in this mode.

Table 148 — Static Logical Pin State in DQ Map Mode [1]

MR9 OP[4:0]	STATIC LOGICAL PIN STATE IN DQ MAP MODE									
	DQ9	DQ8	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
00000_B	DQ map mode is disabled									
10000_B	H	H	H	H	H	H	H	H	H	L
10001_B	H	H	H	H	H	H	H	H	L	H
10010_B	H	H	H	H	H	H	H	L	H	H
10011_B	H	H	H	H	H	H	L	H	H	H
10100_B	H	H	H	H	H	L	H	H	H	H
10101_B	H	H	H	H	L	H	H	H	H	H
10110_B	H	H	H	L	H	H	H	H	H	H
10111_B	H	H	L	H	H	H	H	H	H	H
11000_B	H	L	H	H	H	H	H	H	H	H
11001_B	L	H	H	H	H	H	H	H	H	H

NOTE 1 The equivalent output levels in PAM3 mode are +1 and -1.

11.3 DQ Map Mode (cont'd)



NOTE 1 Valid commands are shown single CK4 cycle for illustration purposes but could be single or multicycle. In either case the reference for timings is to the last CK4 cycle of the command. See the *CLOCKING* section for more details.

Figure 153 — DQ Map Mode Timing

11.4 Package Outline

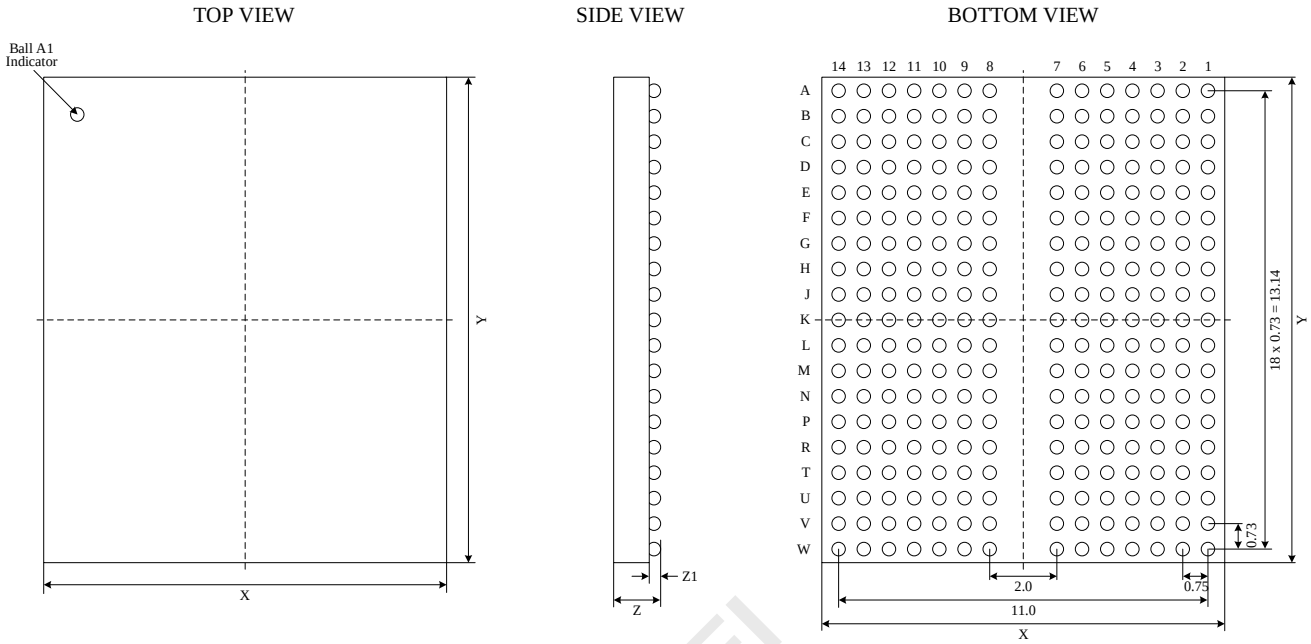


Figure 154 — Package Dimensions

Table 149 — Package Parameters

Parameter	Symbol	Minimum	Nominal	Maximum	Variation	Unit
Width	X		12.0		±0.1	mm
Length	Y		14.0		±0.1	mm
Height	Z		1.0		±0.1	mm
Ball diameter	b	0.35		0.50	±0.05	mm
Standoff	Z1					mm
Ball pitch	eX		0.75			mm
	eY		0.73			mm

NOTE 1 GDDR7 package width, length, ball diameter and standoff specification is compliant to MO-353, variation PBGA-B266[266]_I0p73-R12p0x14p0Z#-C0p5Z#.

NOTE 2 The nominal ball diameter and standoff (Z1) are vendor specific.

11.5 2-Channel Mode Enable

A GDDR7 SGRAM based memory system is typically divided into several channels. A GDDR7 SGRAM device comprises 4 independent 8-bit wide channels. At initialization, it can be configured to operate in a 4-channel mode or be configured for a 2-channel mode. For 2-channel mode the devices are typically assembled on opposite sides of the PCB in what is referred to as a clamshell.

Whether in 4-channel mode or 2-channel mode the device will operate with a point-to-point connection on the high speed command / address (CA) and data signals. In 2-channel mode channels B and D will be inactive; all signals of channels B and D in that mode shall be in High-Z state and left floating in the system.

The channel configuration is set with RESET_n going High on channels A and C. For 4-channel mode signals CA0_A, CA0_B, CA0_C and CA0_CD must be driven High; for 2-channel mode signals CA0_A and CA0_C must be driven Low. Once the configuration has been set, it cannot be changed during normal operation. Usually, the configuration is fixed in the system.

Details of the 2-channel mode detection are depicted in *FIGURE 155*. A comparison of 4-channel mode and 2-channel mode systems is shown in *FIGURE 156*. *FIGURE 158* shows examples of the board topologies that are supported in GDDR7.

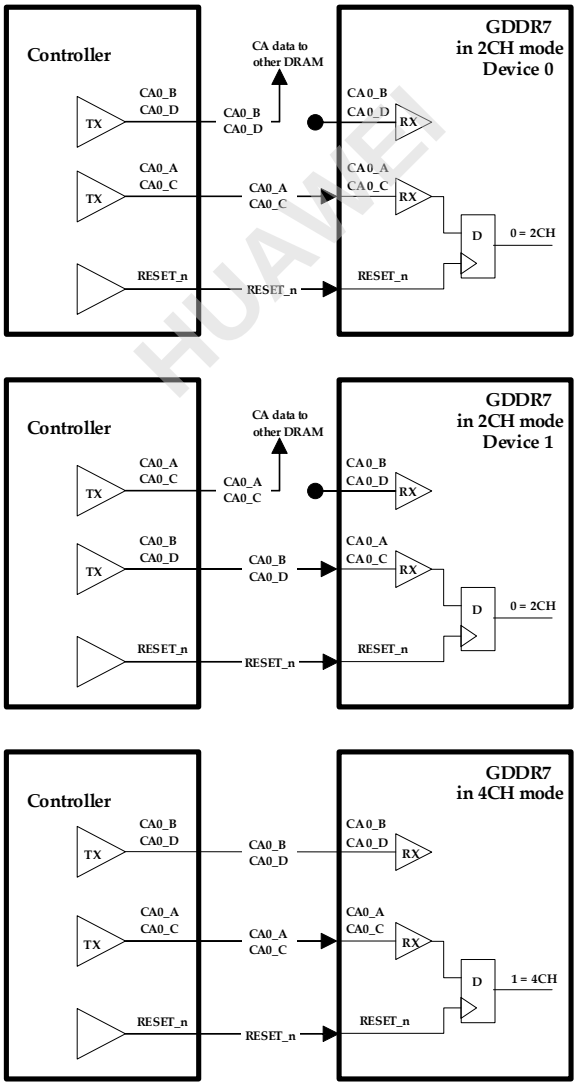


Figure 155 — Enabling 2-Channel Mode

11.5 2-Channel Mode Enable (cont'd)

Table 150 — CA State at Initialization for 4-Channel Mode vs. 2-Channel Mode

MODE	CA0_A	CA0_B	CA0_D	CA0_C
2-channel	LOW	Floating (on board)	Floating (on board)	LOW
4-channel	HIGH			

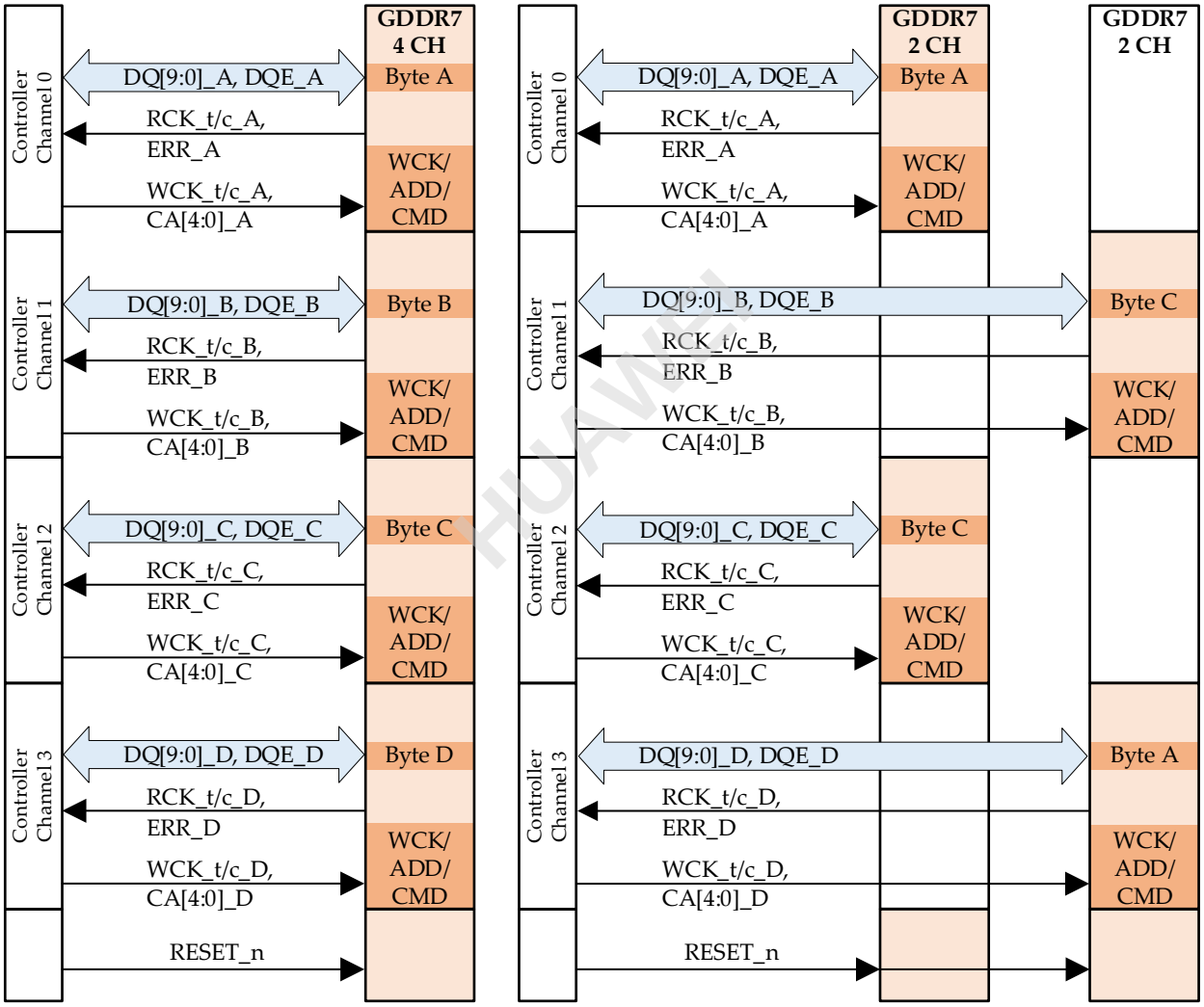


Figure 156 — Example System View for 4-Channel Mode vs. 2-Channel Mode

11.5 2-Channel Mode Enable (cont'd)

FIGURE 157 is an example that clarifies the use of 2-channel mode and how the channels are enabled/disabled to give the controller the view of the same bytes that a controller sees with a single 4-channel device. For a 2-channel using 2 devices in a clamshell design example, two channels come from Channel A and C from the top device and two channels come from the bottom Channel A and C and will look equivalent at the controller to a 4-channel mode.

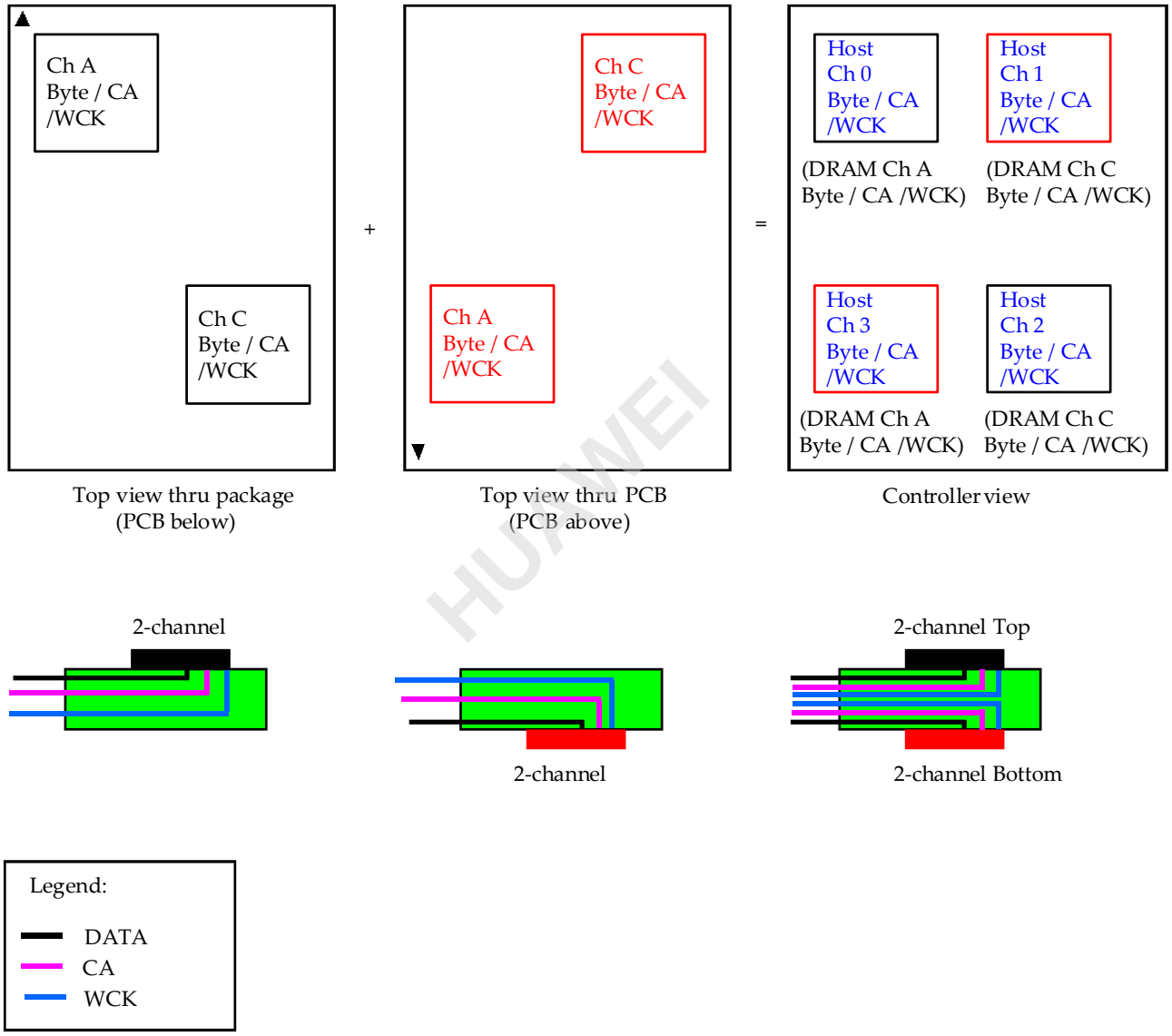


Figure 157 — Byte Orientation in Clamshell Topology

11.5 2-Channel Mode Enable (cont'd)

The simple block diagram in *FIGURE 158* demonstrates some of the flexibility of PCB routing.

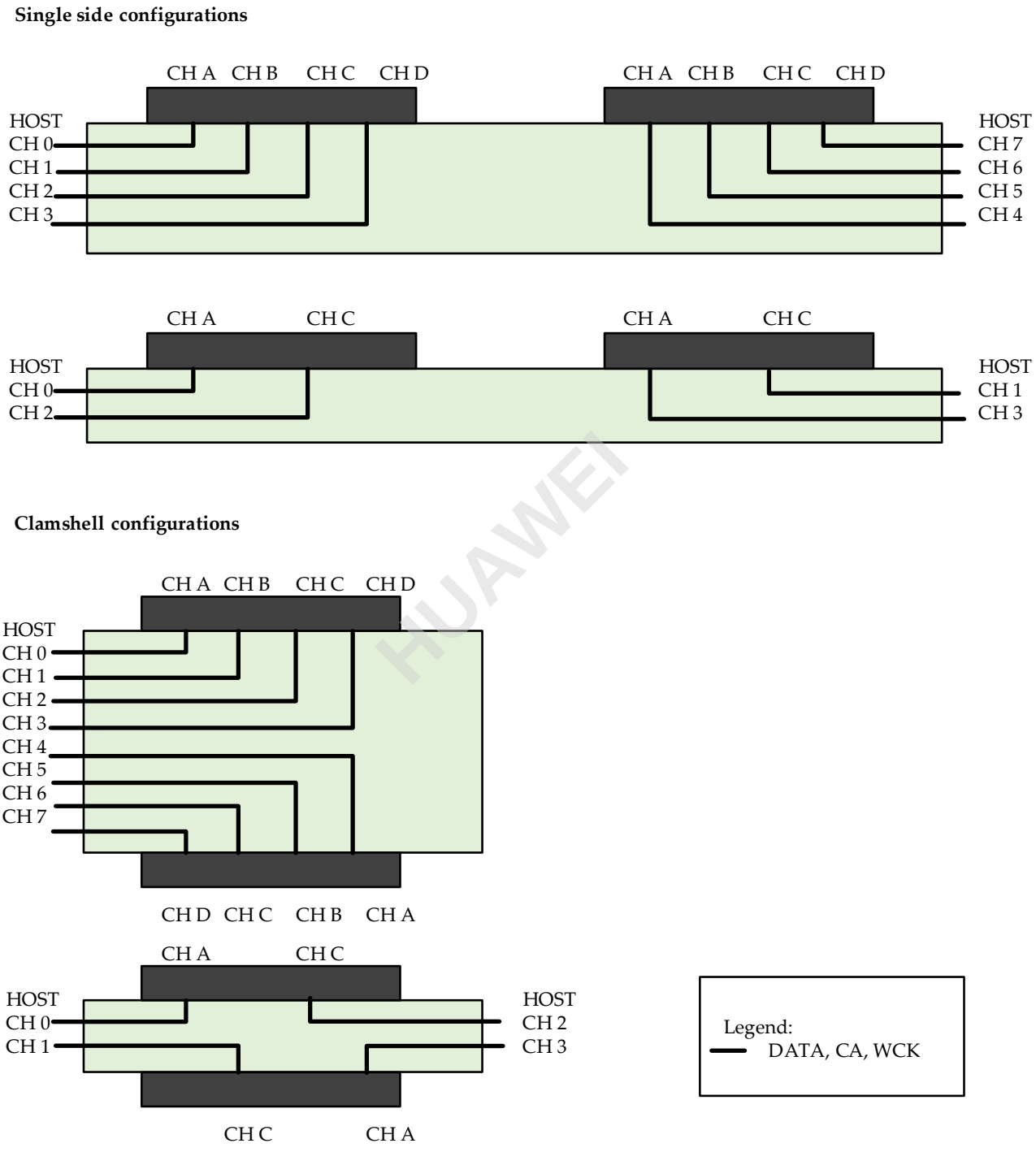


Figure 158 — Example GDDR7 PCB Layout Topologies

Annex A — (Informative) Differences between Document Revisions

A.1 Differences between JESD239.01 and JESD239

- Added a cautionary message on the cover page and page before Table of Contents regarding an ongoing update by the formulating subcommittee that may require host design changes.
- Updated the wording of the “Notice” and “Do Not Violate the Law” pages to the latest JEDEC standard
- Added this Annex

This page intentionally left blank

Standard Improvement Form **JEDEC Standard No. JESD239.01**

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 10th Street North
Suite 240S
Arlington, VA 22201

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: _____