Solid State Technology Association

COMMITTEE LETTER BALLOT

JEDEC Confidential Document

COMMITTEE: JC-42.2
ITEM NUMBER: 1883.98
DATE: 2024-06-04

SUBJECT: Work-in-progress version of HBM4 Full Specification Revision 0.71

BACKGROUND: This ballot draft of HBM4 full specification reflects 2024 1Q Voting comments and numerous feedback from task group members since 1st release. The TG has not yet put this on consensus as some content was not part of the original full spec ballot authorization. The TG intends to issue a full spec ballot next quarter and requires broad authorization from the committee to be allowed to integrate content directly into the new spec when agreement is reached rather than only be allowed to integrate previous passed content as normally done as the specification matures toward publication.

KEYWORDS: HBM4, Full Specification, JESD238

SPONSOR:
Barry Wagner
NVIDIA
BWagner@nvidia.com

ORIGINATOR:
Kijoon Chang
SK hynix
kijoon1.chang@sk.com

PROPOSED

Revision History

Revision	Date	Page	Description
0.7	03/07/2024		- Initial Release
0.71	04/23/2024		- Electrical Characteristics, MTV updates - Appending 2024 1Q ballots, Addressing voting comments from 2023 3Q and 2023 4Q
			-
			-
			-
			-
			-
			-

PROPOSED

JEDEC STANDARD

High Bandwidth Memory DRAM (HBM4)

JESD238

April 2024

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION

3103 NORTH 10TH STREET, SUITE 240-S, ARLINGTON, VA 22201, USA | PH +1.703.907.7540 | FX +1.703.907.7583 | WWW.JEDEC.ORG

PROPOSED

NOTICE

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

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

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

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

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

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

Published by
© JEDEC Solid State Technology Association 2022
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2108

PRICE: Contact JEDEC

Printed in the U.S.A.

PROPOSED

PLEASE!

DON'T VIOLATE

THE LAW!

For information, contact:

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

or refer to www.jedec.org under Standards-Documents/Copyright Information.

This page left intentionally blank

HIGH BANDWIDTH MEMORY DRAM (HBM4)

Contents

Tables

Figures

This page left intentionally blank

HIGH BANDWIDTH MEMORY (HBM4) DRAM

(From JEDEC Board Ballot JCB-21-59, formulated under the cognizance of the JC-42.2 Subcommittee on DRAM Memories, item number xxxx.xx).

1 Scope

The HBM4 DRAM is tightly coupled to the host compute die with a distributed interface. The interface is divided into independent channels. Each channel is completely independent of one another. Channels are not necessarily synchronous to each other. The HBM4 DRAM uses a wide-interface architecture to achieve high-speed, low power operation. Each channel interface maintains a 64 bit data bus operating at double data rate (DDR).

2 Features

- 256 bit prefetch per memory read and write access
- BL = 8
- 64 DQ width + ECC/SEV pins support / channel
- Pseudo Channel (PC) mode operation; 32 DQ width for PC mode
- Differential clock inputs (CK_t/CK_c) for command/address
- Double data rate (DDR) command/address. Row Activate commands require one-and-a-half-cycle, all other row commands require a half-cycle except for PDE, SRE with one cycle. Column command require only one cycle
- Semi-independent row and column command interfaces allowing Activates/Precharges to be issued in parallel with Read/Writes
- Data referenced to unidirectional differential data strobes RDQS_t/RDQS_c and WDQS_t/WDQS_c. One strobe pair each per DWORD
- Up to 32 channels / device
- Channel density of 3 Gb to 16 Gb
- 16, 32, 48 or 64 banks per channel; varies by device density / channel
- Bank grouping supported
- 1 KB page size per pseudo channel (PC)
- DBIac support configurable via MRS
- Self refresh modes
- Vendor Specific I/O voltage, Tx driver voltage TBD
- DRAM core voltage 1.05 V, independent of I/O voltage
- Unterminated data/address/command/clock interfaces
- Unmatched data interfaces

3 Organization

HBM4 DRAM is optimized for high-bandwidth operation utilizing several independent interfaces called channels (CH). Each channel is further segmented into semi-independent pseudo channels (PC).

Each HBM4 stack will support up to 32 channels. Figure 1 shows an example stack containing 8 DRAM dies, each die supporting 8 independent channels with 2 pseudo channels per channel. Each die contributes additional capacity and additional channels to the stack (up to a maximum of 32 channels per stack or 64 pseudo channels). HBM4 requires 4 DRAM dies to support 32 channels. Additional DRAM dies beyond 4 add additional capacity, SIDs and additional banks per pseudo channel. HBM4 can support stacks of either 4, 8, 12 or 16 DRAM dies. See 3.2 Channel Addressing for more details.

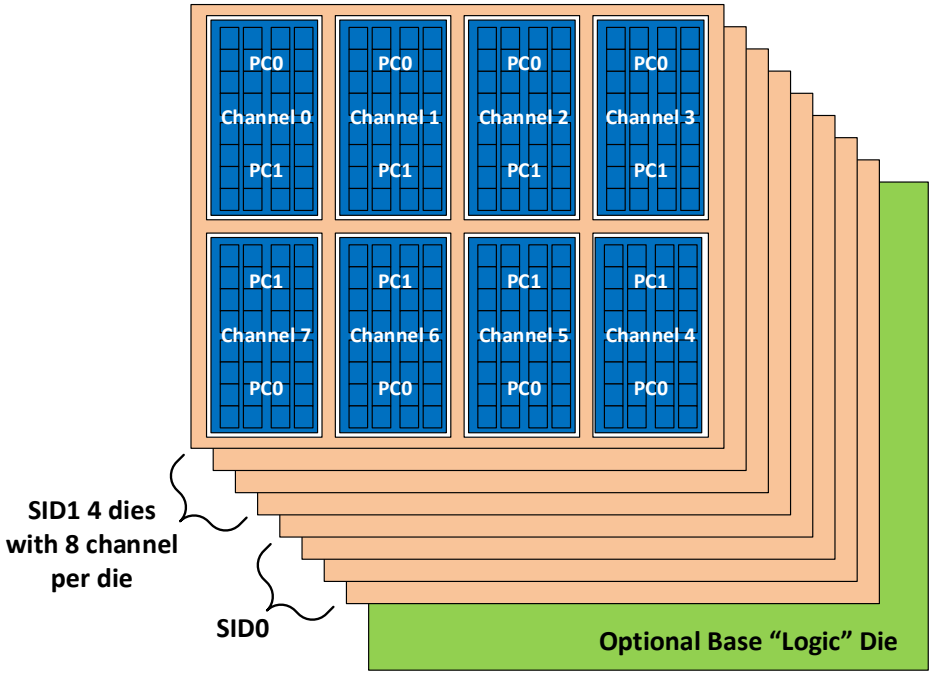


Figure 1 — Example Logical Overiew of a HBM4 Device

The DRAM vendor may choose to require an optional interface die that sits at the bottom of the stack and provides signal redistribution and other functions. The vendor may choose to implement many of the logic functions typically found on DRAM die on this logic die. This standard does not explicitly require nor prohibit such a solution.

The division of channels and pseudo channels among the DRAM dies within a stack is left to the vendor. Figure 1, with the memory for eight channels implemented on each die, is not a required organization. Organizations are permitted where the memory for a single channel is distributed among multiple dies; however, all accesses within a single channel must have the same latency for all accesses.

3.1 Channel Definition

Each channel consists of an independent command and data interface. RESET_n, CATTRIP, IEEE1500 test port and power supply signals are common to all channels. Channels are independently clocked, and need not to be synchronous, however the clock is shared between both pseudo channels in a channel.

Since each channel is independent, much of this standard will describe a single channel. Where signal names are involved, families of signals belonging to a given channel will have the suffix 0, 1, …, 31 for channels 0 through 31. If no suffix is present, the signal(s) being described are generic instances of the various per-channel signals.

3.1.1 Signal Count

Table 1 — Single Channel Signal Count

Function	Number of Microbumps	Notes
Data	64	DQ[63:0]
Column command/ Address	8	C[7:0]
Row command/ Address	10	R[9:0]
DBI	8	1 DBI per 8 DQs
ECC	4	2 ECC per 32 DQs
SEV	4	2 SEV per 32 DQs
DPAR	2	1 PAR per 32 DQs
APAR	1	1 PAR per AWORD
DERR	2	1 DERR per 32 DQs
Strobe	8	1 RDQS_t/RDQS_c, WDQS_t/WDQS_c per 32 DQs
Clock	2	CK_t/CK_c
AERR	1	AERR per AWORD
Redundant Data	4	RD[3:0]
Redundant Address	1	Redundant row / column
RFU	1	1 RFU per AWORD
Total	120	

Table 2 — Global Signal Count

Function	Number of Microbumps	Description	Notes
RESET_n	2	Global Stack Reset	1
WRCK	2	IEEE1500 Clock	1
WRST_n	2	IEEE1500 only Reset	1
WSI	2	IEEE1500 Serial Input	1
SelectWIR	2	IEEE1500 Select WIR	1
CaptureWR	2	IEEE1500 Capture WR	1
ShiftWR	2	IEEE1500 Shift WR	1
UpdateWR	2	IEEE1500 Update WR	1
WSO	32	1 IEEE1500 Serial Output Per Channel [0:31]	
RM	2	Redundant WSO	
CATTRIP	2	Catastrophic Temperature Sensor	1
MRFU	4	Midstack RFU	
Total	56		

NOTE 1 Duplicate microbumps for link redundancy

3.1.2 Pseudo Channel Definition

A Pseudo channel (PC) divides a channel into two individual sub-channels of 32 DQ each, providing 256 bit prefetch per memory read and write access for each pseudo channel. Each pseudo-channel provides access to an independent set of DRAM banks of a defined page size. Requests from one pseudo-channel may not access data attached to a different pseudo-channel.

Both pseudo channels operate semi-independent. They share the channel's row and column command bus as well as CK inputs, but decode and execute commands individually as illustrated in Figure 2. Address PC is used to direct commands to either to pseudo channel 0 (PC = 0) or pseudo channel 1 (PC = 1). Power-down and self refresh are common to both pseudo channels. All I/O signals of DWORD0 are associated with pseudo channel 0, and all I/O signals of DWORD1 with pseudo channel 1.

Array access timings as listed in the table below are applicable for each individual pseudo channel. For example, an ACTIVATE to PC0 can be followed by an ACTIVATE to PC1 as shown in Figure 2. However, a subsequent ACTIVATE to PC0 can only be done after tRRD (PC0). For commands that are common to both pseudo channels (PDE, PDX, SRE, SRX and MRS), it is required that the respective timing conditions are met by both pseudo channels when issuing that command. Both pseudo channels also share the channel's mode registers.

Table 3 — Array Access Timings Counted Individually Per Pseudo Channel

Array Timing Group	Notes
Row Access Timings	t_{RC}, t_{RAS}, t_{RCDRD}, t_{RCDWR}, t_{RRDL}, t_{RRDS}, t_{FAW}, t_{RTP}, t_{RP}, t_{WR}
Column Access Timings	t_{CCDL}, t_{CCDS}, t_{CCDR}, t_{WTRL}, t_{WTRS}, t_{RTW}
Refresh Timings	t_{RFC}, t_{RFCPB}, t_{RREFD}, t_{REFI}, t_{REFIPB}, t_{RTW}

3.1.2 Pseudo Channel (cont'd)

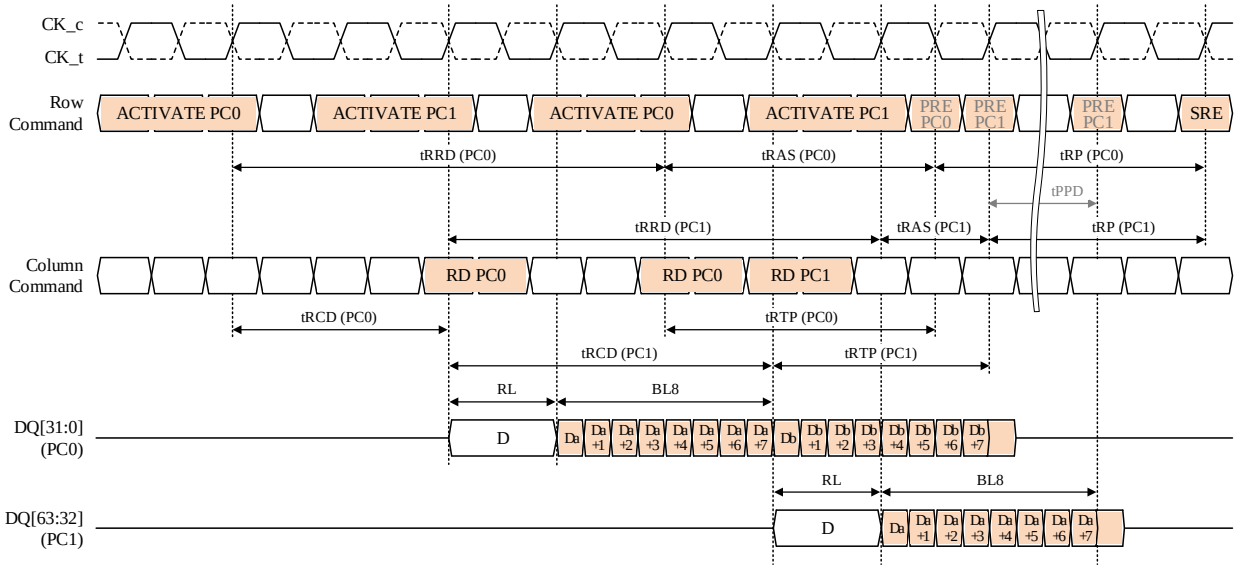


Figure 2 — Pseudo Channel Operation

3.1.3 Dual Command Interfaces

To enable higher performance, HBM4 DRAMs exploit the increase in available signals to provide semi-independent row and column command interfaces for each channel. These interfaces increase command bandwidth and performance by allowing read and write commands to be issued simultaneously with other commands like activates and precharges. See Commands.

3.2 Channel Addressing

Table 4 — HBM4 Channel Addressing

Configuration	24Gb 4H[8]	24Gb 8H	24Gb 12H	24Gb 16H	Note
Density per Channel	**3Gb**	**6Gb**	**9Gb**	**12Gb**	
Density per PC	**1.5Gb**	**3Gb**	**4.5Gb**	**6Gb**	
Prefetch Size per PC (bits)	256	256	256	256	1, 2
Row Address	RA[13:0] [5]	RA[13:0] [5]	RA[13:0] [5]	RA[13:0] [5]	
Column Address	CA[4:0]	CA[4:0]	CA[4:0]	CA[4:0]	
Bank Address	BA[3:0]	SID[0], BA[3:0]	SID[1:0] [6], BA[3:0]	SID[1:0], BA[3:0]	
Page Size per PC	1KB	1KB	1KB	1KB	1, 3
Density Code	0000	0010	0100	0110	8
Configuration	**32Gb 4H[8]**	**32Gb 8H**	**32Gb 12H**	**32Gb 16H**	
Density per Channel	**4Gb**	**8Gb**	**12Gb**	**16Gb**	**Note**
Density per PC	**2Gb**	**4Gb**	**6Gb**	**8Gb**	
Prefetch Size per PC (bits)	256	256	256	256	1, 2
Row Address	RA[13:0]	RA[13:0]	RA[13:0]	RA[13:0]	
Column Address	CA[4:0]	CA[4:0]	CA[4:0]	CA[4:0]	
Bank Address	BA[3:0]	SID[0], BA[3:0]	SID[1:0][6], BA[3:0]	SID[1:0] , BA[3:0]	
Page Size per PC	1KB	1KB	1KB	1KB	1, 3
Density Code	0001	0011	0101	0111	7

NOTE 1 Prefetch size and page size reflect the effective addressing along with row and column commands. Both do not include the optional ECC bits as described in section 3.1 Channel Definition.

NOTE 2 The burst order is fixed for Reads and Writes, and the HBM device does not assign column address bits to distinguish between the eight UI of a BL8 burst. A memory controller may internally assign such column address bits but those column address bits are not transmitted to the HBM device.

NOTE 3 Page Size = 2^COLBITS * (Prefetch Size / 8); where COLBITS is the number of column address bits. Page size and prefetch size per pseudo channel in Pseudo Channel. MSB of RA is used to select half of open 2 KB page.

NOTE 4 SID, SID0, SID1 act as bank address bits in command execution. Specific AC timing parameters or variations on selected timing parameters may be linked to SID.

NOTE 5 RA[13:12] = 11 is invalid.

NOTE 6 SID[1:0] = 11 is invalid.

NOTE 7 The density code refers to the encoding of per-channel density in DEVICE_ID Wrapper Data Register, bits [43:40].

NOTE 8 SID[0]=1 is invalid in 4Hi configuration

3.2.1 Bank Groups

The banks within a device are divided into 2 or 4 or 6 or 8 bank groups. The assignment of banks to bank groups is shown in Table 5.

Different timing parameters are specified depending on whether back-to-back accesses are within the same bank group or across bank groups at shown in Table 6.

Table 5 — Bank Group Assignments

Banks	16 Banks BA[3:0]	32 Banks SID, BA[3:0]	48 Banks SID[1:0][1], BA[3:0]	64 Banks SID[1:0], BA[3:0]
0 and 7	Group A	Group A	Group A	Group A
8 to 15	Group B	Group B	Group B	Group B
16 to 23	N/A	Group C	Group C	Group C
24 to 31		Group D	Group D	Group D
32 to 39		N/A	Group E	Group E
40 to 47			Group F	Group F
48 to 55			N/A	Group G
56 to 63				Group H

NOTE 1 SID[1:0] = 11 is invalid

Table 6 — Command Sequence Affected by Bank Groups

Command Sequence	Corresponding AC Timing Parameter		Notes
	Accesses To Different Bank Groups	Accesses Within Same Bank Group	
ACTIVATE to ACTIVATE	t_{RRDS}	t_{RRDL}	
WRITE to WRITE	t_{CCDS}	t_{CCDL}	
READ to READ	t_{CCDS} or t_{CCDR}	t_{CCDL}	
Internal WRITE to READ	t_{WTRS}	t_{WTRL}	
READ to PRECHARGE	-	t_{RTP}	1

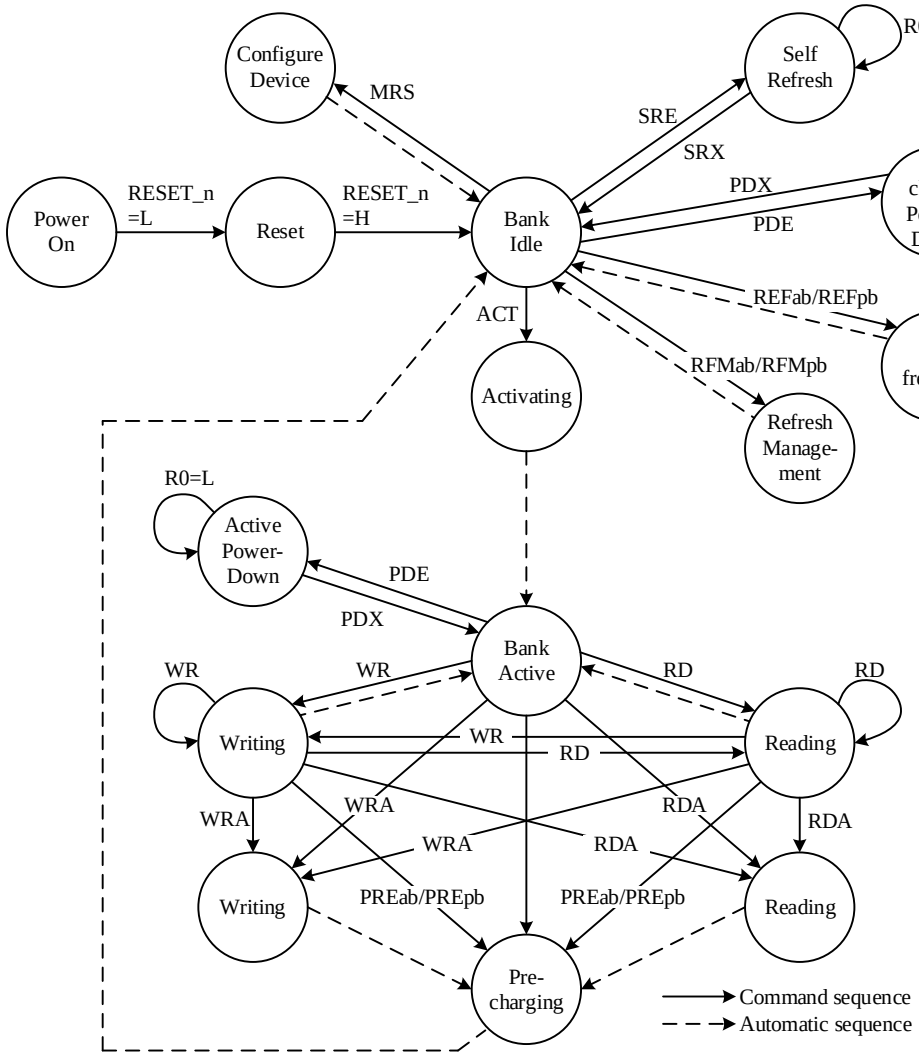
NOTE 1 Parameters t_{RTP} applies only when READ and PRECHARGE go to the same bank.
NOTE 2 Parameters t_{CCDR} replaces parameter t_{CCDS} when consecutive READs go to banks with different stack IDs (SID).

3.3 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 either not shown or not fully shown in the diagram:

- state transitions involving more than one bank;
- interactions from the use of IEEE1500 instructions to load mode registers or execute test functions;
- the immediate transition from any state to reset state by asserting RESET_n LOW or by loading the IEEE1500 instructions HBM_RESET;
- the ECS and ECC Engine Test Mode operation;
- DCA and DCM;
- Loopback Test Mode;
- WDQS-to-CK Alignment Training
- Rx Offset Calibration Training (TBD)

For a complete description of the device behavior, use the information provided in the state diagram along with the command truth tables and AC timing specifications.



Code	Command
ACT	Active
PREab	Precharge all
PREpb	Precharge
RD	Read
RDA	Read with Auto Precharge
WR	Write
WRA	Write with Auto Precharge
REFab	All bank refresh
REFpb	Per bank refresh
RFMab	All bank refresh management
RFMpb	Per bank refresh management
PDE	Power down entry
PDX	Power down exit
SRE	Self refresh entry
SRX	Self refresh exit
MRS	More register set

Figure 3 — Simplified State Diagram

4 Initialization

To power-up and initialize the HBM4 device into functional operation the sequence in section 4.1 must be followed. At any time after the power-up initialization, the HBM4 device may be reset using the sequence in this section. A limited set of IEEE 1500 port instructions may be used within the initialization sequences, as described in section 4.3.

The interactions between HBM4 functional reset and the IEEE 1500 port reset are as follows (also see section 13.2 IEEE Standard 1500):

- Functional reset requires that the IEEE 1500 port also be reset.
- The IEEE 1500 port can be reset at any time without impacting normal operation.
- The IEEE 1500 port may be brought out of reset and a limited set of instructions may be used after a minimum time after RESET_n has been deasserted. See section 4.3.
- If not needed, the IEEE 1500 port may be left in reset (WRST_n = LOW) during normal operation.

4.1 HBM4 Power-up and Initialization Sequence

HBM4 device must be powered up and initialized in a predefined manner. The following sequence and timing must be satisfied for HBM4 power up and initialization sequence. Also refer to Figure 4.

1. Apply power to the V_{DDC}, V_{DDQ}, V_{DDQL} and V_{PP} supplies following the requirements in the Power Ramp Conditions table. V_{PP} must ramp at the same time or earlier than V_{DDC} and V_{DDQ}. V_{DDC} and V_{DDQ} must ramp simultaneously under the same level. V_{DDC} and V_{DDQ} must ramp at the same time or earlier than V_{DDQL}. During power supply ramp time t_{INIT0}, RESET_n, WRST_n and all other input signals may be in an undefined state (driven LOW or HIGH, or Hi-Z). After T_0 in Figure 4 is reached, V_{DDC} and V_{DDQ} must be greater than V_{DDQL}-200 mV.
2. RESET_n and WRST_n must be driven LOW (below 0.2 × V_{DDQ}) before or at the same time when t_{INIT0} expires as shown in Figure 4 (time T_a). All other input signals may be in an undefined state (driven LOW or HIGH, or Hi-Z) at this point. RESET_n must be maintained LOW for a minimum of t_{INIT1} time with stable power. After t_{INIT6} time has elapsed, the HBM4 device drives RDQS_t and RDQS_c to LOW and HIGH static levels, respectively, and AERR, DERR and CATTRIP signals to LOW.
3. A time t_{INIT2} before RESET_n is pulled HIGH, CK_t and CK_c must be driven to static LOW and HIGH levels, respectively.
4. After RESET_n is driven HIGH, R[3:0] must be driven to PDE state (HIGH, LOW, HIGH, LOW) and C[2:0] must be driven to CNOP state (HIGH, HIGH, HIGH) for a t_{INIT7} time before CK clock is toggled. R[9:4] and C[7:3] are allowed to remain in an undefined state. The HBM4 device resets into the precharged power-down state. During t_{INIT3}, the HBM4 device will read and apply internal fuse configuration data and perform I/O driver impedance calibration. At the same time the WRST_n signal may be optionally driven HIGH to enable a subset of the IEEE 1500 instructions (see Initialization Sequence For Use Of IEEE 1500 Instruction Including Lane Repairs and IEEE Standard 1500 sections). In that case, all other IEEE1500 inputs (WRCK, SelectWIR, ShiftWR, CaptureWR, UpdateWR, WSI) must be driven per IEEE1500 Port Input and Output Timings figure at time t_{WINIT2} before WRST_n is pulled HIGH(see IEEE1500 Test Port AC Timing Parameters). CATTRIP data must stay LOW from the end of t_{INIT6} to the end of t_{INIT3} and valid data must start after t_{INIT3}.

4.1 HBM4 Power-up and Initialization Sequence (cont'd)

5. While R[3:0] and C[2:0] remain driven to PDE state as defined in step 4, the CK clock shall be started and stable clocks shall be maintained for minimum of t_{INIT4} time before driving R[3:0] HIGH. Since R[0] of R[3:0] is a synchronous signal, the corresponding setup time to clock (t_{IS}) must be met. Also, RNOP and CNOP commands must be registered (with t_{IS} / t_{IH} satisfied). After R[3:0] are registered HIGH, a minimum t_{INIT5} time must be satisfied before issuing a first MRS command. At or before the time that R[3:0] are driven HIGH, WDQS_t and WDQS_c must be driven to LOW and HIGH static levels, respectively. A stable CK clock shall be maintained except when a channel is in power-down or self refresh state. See Power-Down and Self Refresh section for conditions about stopping and re-starting the CK clock.

6. Issue all MRS commands to configure the HBM4 device appropriately for the application setting.

7. The HBM4 device is now ready for normal operation.

Table 7 — Initialization Timing Parameters

Symbol	Description	Min	Max	Unit
t_{INIT0}	Power supply ramp time	0.01	200	ms
t_{INIT1}	RESET_n signal LOW time at power-up (after stable power)	200		us
t_{INIT2}	CK_c and CK_t must be driven to HIGH and LOW before RESET_n deassertion	10		ns
t_{INIT3}	Precharged power-down state and WRST_n LOW time after RESET_n deassertion	4		ms
t_{INIT4}	CK clock stable time before R[3:0] HIGH	10		nCK
t_{INIT5}	Idle time before first MRS command	200		ns
t_{INIT6}	RDQS_t, RDQS_c driven valid and AERR, DERR and CATTRIP driven LOW after RESET_n assertion		100	ns
t_{PW_RESET}	RESET_n signal LOW time with stable power	1		us
t_{INIT7}	R[3:0] and C[2:0] must be driven to PDE and CNOP before CK clock toggling	2		nCK

Table 8 — Power Ramp Conditions

After	Application Condition
T_0	V_{PP} must be greater than V_{DDC}, V_{DDQ}
	V_{DDC} and V_{DDQ} must be greater than V_{DDQL} – 200 mV
NOTE 1	T_0 is the point when any power supply first reaches 300 mV
NOTE 2	Voltage ramp conditions in this table apply between T_0 and controlled power-off.
NOTE 3	T_a is the point at which all supply voltages are within their defined ranges.
NOTE 4	Power ramp duration t_{INIT0} (T_a – T_0) must not exceed 200 ms.

4.1 HBM4 Power-up and Initialization Sequence (cont'd)

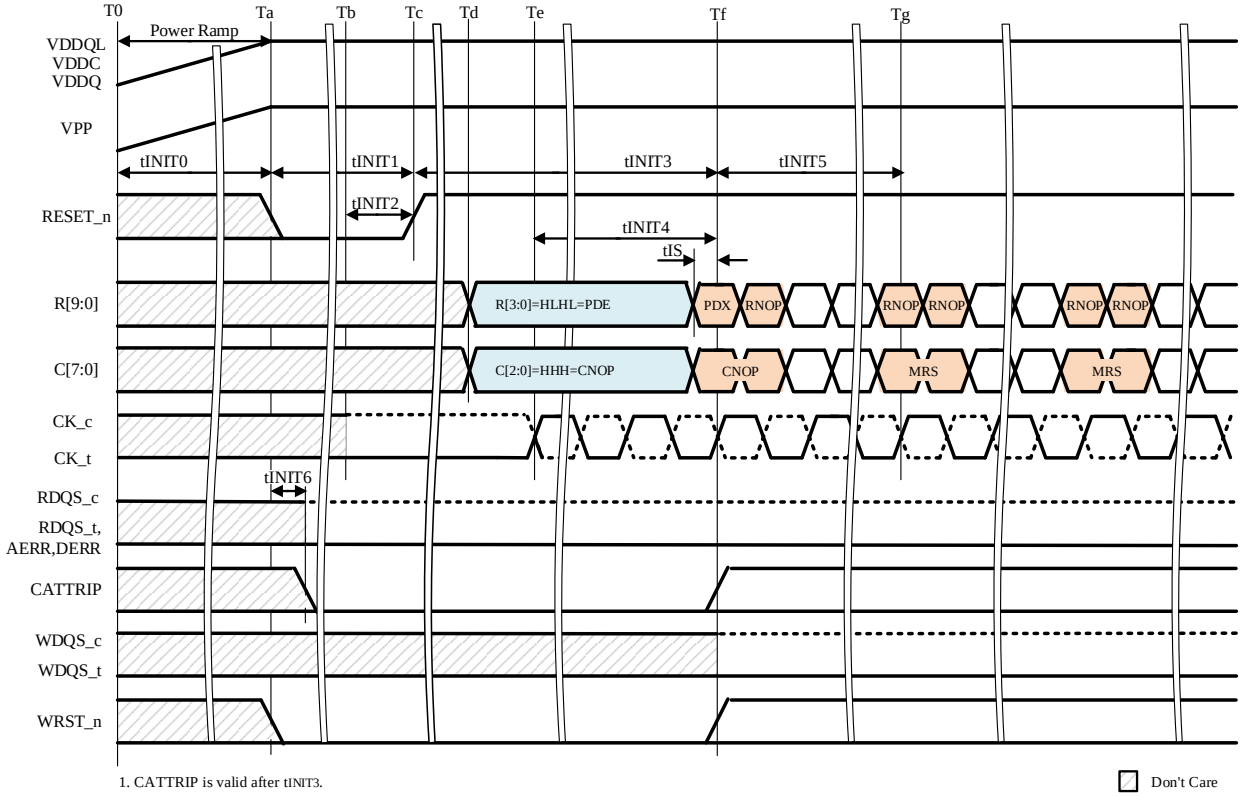


Figure 4 — Power-up and Initialization

4.2 Controlled Power-off Sequence

For a controlled power-off, the conditions in Table 9 must be met:

While powering off, all input levels must be between VSS and VDDQ or VDDQL during voltage ramp to avoid latch-up.

Table 9 — Power Supply Conditions

Between	Application Condition
T_X and T_Z	V_{PP} must be greater than V_{DDC}, V_{DDQ}
	V_{DDC} and V_{DDQ} must be greater than V_{DDQL} – 200 mV
NOTE 1 T_x is the point where any power supply drops below the minimum value specified. NOTE 2 Tz is the point where all power supplies are below 300 mV. After Tz, the HBM4 device is powered off.	

4.3 Initialization Sequence with Stable Power

Steps 1 and 2 must be satisfied to perform a functional reset when power is kept stable at the HBM4 DRAM. See Figure 5.

1. RESET_n must be driven LOW anytime when a functional reset is needed. All other input signals may be in an undefined state (driven LOW or HIGH, or Hi-Z) at this point except WRST_n and CATTRIP as shown in Figure 5. RESET_n must be maintained LOW for a minimum of t_{PW_RESET}. R[3:0] must be driven to PDE state (HIGH, LOW, HIGH, LOW) and C[2:0] must be driven to CNOP state (HIGH, HIGH, HIGH) for a t_{INIT7} time before CK clock is toggled. R[9:4] and C[7:3] are allowed to remain in an undefined state. Alternately, the IEEE 1500 port HBM4_RESET instruction may be used to perform a re-initialization, with RESET_n continuing to be driven HIGH. Refer to HBM_RESET section.

2. Follow steps 3 to 6 as described in section HBM4 Power-up and Initialization Sequence. Note that the CATTRIP output is sticky and not cleared by a functional reset.

A time t_{INIT2} before RESET_n is pulled HIGH, CK_t and CK_c must be driven to static LOW and HIGH levels, respectively. See step 3 of the HBM4 Power-up and Initialization Sequence.

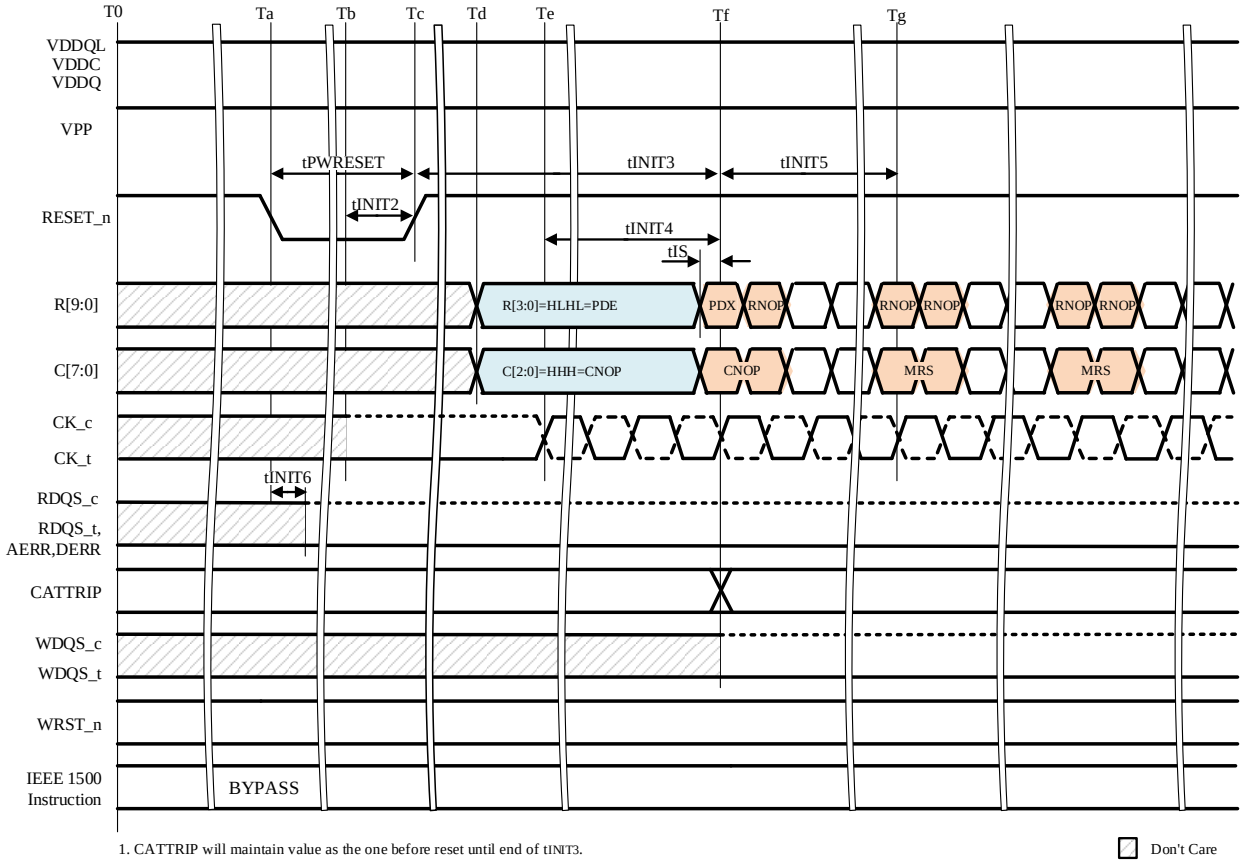


Figure 5 — HBM4 RESET and Initialization Sequence with Stable Power

4.4 Initialization Sequence For Use Of IEEE 1500 Instruction Including Lane Repairs and Channel Disable

All IEEE 1500 port instructions are allowed after t_{INIT3} without completing the full initialization sequence.NOTE 1 After EXTEST operations, another RESET_n toggle may be required.
NOTE 2 R[9:0] and C[7:0] mean logical pin name because those pin's physical location will be changed after soft or hard lane repair.

Figure **6** illustrates usage of the EXTEST and SOFT_LANE_REPAIR instructions and Figure 7 the usage of the CHANNEL_DISABLE instruction within the initialization sequence. These sequence may be applied as part of the power-up or stable-power initialization sequence to check for and correct failed connections on the row and column command buses, which must be correctly driven to RNOP and CNOP as part of this initialization sequence. It may also be used to disable a channel before normal operation mode is entered. DWORD lane repairs are also allowed.

1. At time T_a, RESET_n and WRST_n must be driven LOW.

2. After a minimum time t_{INIT1} (if during an initial power-up sequence) or after t_{PW_RESET} (if during a stable power initialization sequence) RESET_n shall be driven HIGH. t_{INIT2} must also be met.

3. After t_{INIT3}, WRST_n is driven HIGH. IEEE 1500 port instructions may now be used. (Note that the WRST_n low pulse width t_{WRSTL} is met since t_{WRSTL} is less than the t_{INIT1} or t_{PW_RESET}). Refer to IEEE1500 Test Port AC Timing Parameters for timing requirements for operating the IEEE 1500 port, including t_{SWRST}. At this point, a defective channel may be disabled; also, defective lane detection and soft lane repair may be executed. EXTEST operations may be applied to identify lanes needing repair. If soft lane repair is needed, SOFT_LANE_REPAIR and HARD_LANE_REPAIR operations can be applied after another RESET_n toggle, which is required after EXTEST instruction operation. An IEEE 1500 port BYPASS instruction should be applied to return all HBM4 signals to their normal functional mode after SOFT_LANE_REPAIR operations. Alternately, WRST_n may be driven LOW.

4. The initialization sequence may then continue per steps 4 to 6 of HBM4 Power-up and Initialization Sequence, as needed.

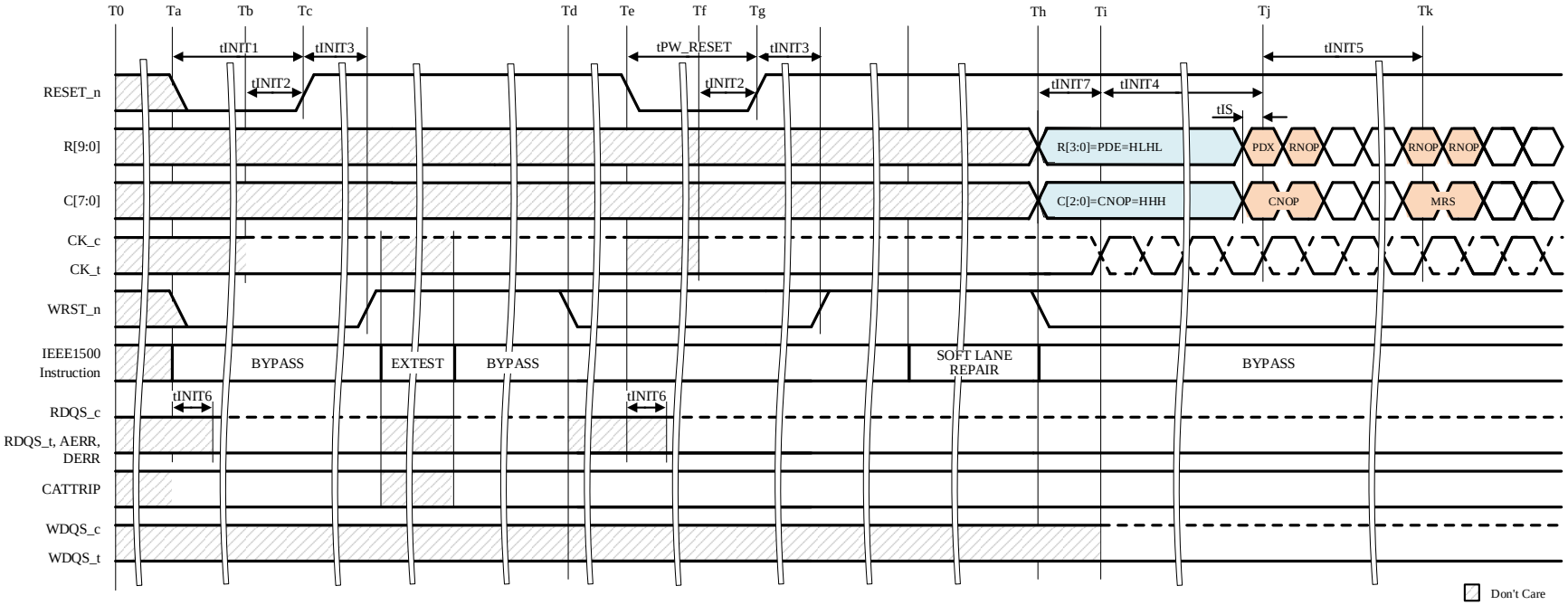
During the t_{INIT3} period before WRST_n is driven HIGH, the HBM4 device executes various internal configuration operations, including applying hard lane repairs based on previously fused data. Executing soft lane repair instructions after t_{INIT3} overwrites any previously programmed hard lane repair data. It is suggested that the hard lane repair data is read from the HBM4 device and merged in any new lane repairs before applying the new soft lane repair operations. Any applicable IEEE 1500 port instructions timings must be met before continuing to time point T_h, such as t_{SLREP} if a SOFT_LANE_REPAIR instruction has been applied.

The EXTEST instructions are not required before applying the soft lane repair(s). Previously determined needed lane repairs may be applied as part of each initialization event.

A time t_{INIT2} before RESET_n is pulled HIGH, CK_t and CK_c must be driven to static LOW and HIGH levels, respectively. See step 3 of the HBM4 Power-up and Initialization Sequence.

R[3:0] must be driven PDE state and C[2:0] must be driven CNOP state for a t_{INIT7} time.

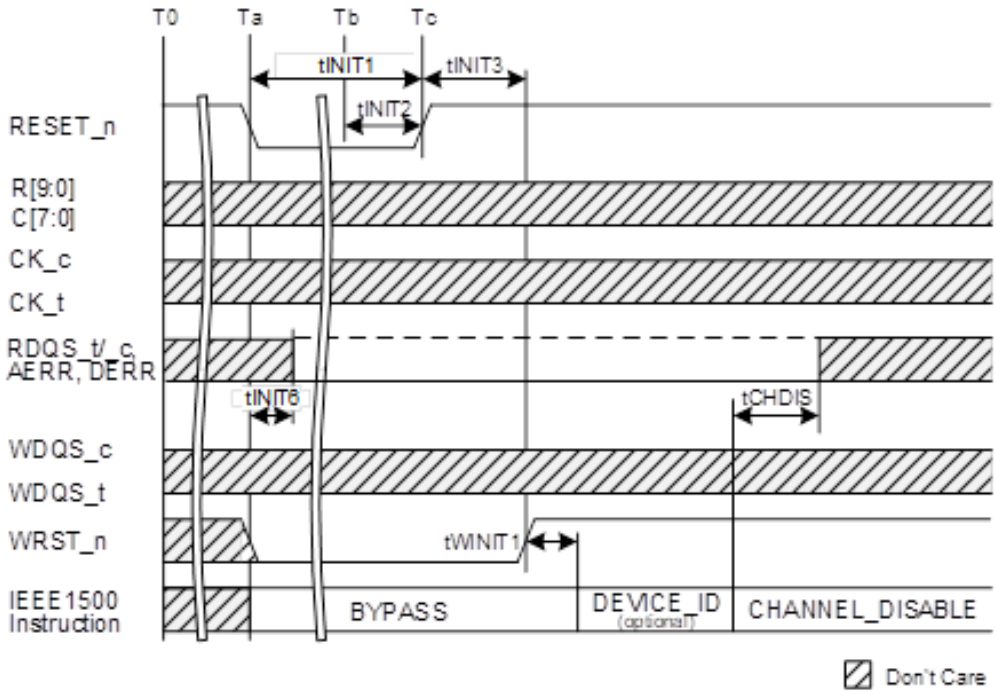
4.4 Initialization Sequence For Use Of IEEE 1500 Instruction Including Lane Repairs and Channel Disable (cont'd)



NOTE 1 After EXTEST operations, another RESET_n toggle may be required.
NOTE 2 R[9:0] and C[7:0] mean logical pin name because those pin's physical location will be changed after soft or hard lane repair.

Figure 6 — Initialization Sequence with Lane Repair or Channel Disable

4.4 Initialization Sequence For Use Of IEEE 1500 Instruction Including Lane Repairs and Channel Disable (cont'd)



NOTE 1 A disabled channel will turn off all AWORD and DWORD input and output buffers including CK_t/CK_c and WDQS_t/WDQS_c inputs, thus allowing all external signals to float. The CK clock is allowed to be High-Z throughput this initialization sequence.

Figure 7 — Initialization Sequence with Channel Disable

5 Mode Registers

The Mode Registers define the specific mode of operation for the HBM4 DRAM. Nineteen 8-bit wide Mode Registers (MR0 to MR19) are defined as in Table 11 through Table 31. MR12 and MR17 are special mode register and reserved for vendor specific features. Mode Registers are common to both pseudo channels (PC0 and PC1). Reprogramming the Mode Registers does not alter the contents of the memory array.

Mode Registers are programmed via the MODE REGISTER SET (MRS) command and retain the stored information until they are reprogrammed, chip reset, or until the device loses power. Mode Register can also be programmed via the IEEE1500 instruction MODE_REGISTER_DUMP_SET; this instruction can also be used to retrieve the Mode Register content.

Mode Registers must be loaded when all banks are idle and the time t_{RDMRS} from a preceding READ command has elapsed. The controller must wait the specified time t_{MOD} before initiating any subsequent operations. Violating either of these requirements will result in unspecified operation.

No default states are defined for Mode Registers except when otherwise noted. Users therefore must fully initialize all Mode Registers to the desired values upon power-up or after a subsequent chip reset.

When an entire Mode Register is marked as RFU (“Reserved for future use”), then it is considered as not supported by the HBM4 DRAM, and its content is Don’t Care. Reserved states should not be used, as unknown operation or incompatibility with future versions may result. RFU bits in these registers must be programmed to 0.

5 Mode Registers (cont'd)

Table 10 — HBM4 Mode Register Overview

Mode Register		OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
	MA[4:0][1]								
MR0 (Table 11)	00000	Test Mode (TM)	CA Parity (CAPAR)	Write Parity (WPAR)	Read Parity (RPAR)	DRFM	TCSR	Write DBI (WDBI)	Read DBI (RDBI)
MR1 (Table 12)	00001	Parity Latency (PL)				Write Latency (WL)			
MR2 (Table 13)	00010	Read Latency (RL)							
MR3 (Table 14)	00011	Write Recovery for Auto Pre-charge (WR)							
MR4 (Table 15)	00100	Activate to Precharge (RAS)							
MR5 (Table 16)	00101	RFU				Read to Auto Precharge (RTP)			
MR6 (Table 17)	00110	DCM Flip	DCM (Duty Cycle Monitor)	Pullup Driver Strength		Pulldown Driver Strength			
MR7 (Table 18)	00111	CATTRIP	RFU	DWORD MISR Control		RFU	DWORD Read Mux Control	DWORD Loopback	
MR8 (Table 20)	01000	DRFM Bounded Refresh Configuration (BRC)		RFM Levels (RFML)		WDQS-to-CK Training (WDQS2CK)	ECS error log auto reset (ECSLOG)	Rx Calibration Offset	DA Port Lockout
MR9 (Table 21)	01001	ECS error Type and Address Reset (ECSRES)	ECS Multi-bit Error Correction (ECSCEM)	Auto ECS during Self Refresh (ECSSRF)	Auto ECS via REFab (ECSREF)	Error Vector Pattern (ECCVEC)	Error Vector Input Mode (ECCTM)	Severity Report-ing (SEVR)	Meta Data (MD)
MR10 (Table 22)	01010	DCA code for RDQS1 (PC1)				DCA code for RDQS0 (PC0)			
MR11 (Table 23)	01011	DCA code for WDQS1 (PC1)				DCA code for WDQS0 (PC0)			
MR12 (Table 24)	01100	Reserved for Vendor Specific Features							
MR13 (Table 25)	01101	Reference Voltage for AWORD inputs (VREFCA)							
MR14 (Table 26)	01110	RFU	Reference Voltage for DWORD inputs (VREFD)						RFU
MR15 (Table 27)	01111	RFU				DFE Code (PC1)		DFE Code (PC0)	
MR16 (Table 28)	10000	Reserved for Decision Feedback Equalizer (DFE)							
MR17 (Table 29)	10001	Reserved for Vendor Specific Features							
MR18 (Table 30)	10010	RFU							
MR19 (Table 31)	10011	RFU							

NOTE 1 MA4 is a valid mode register address bit that must be set to 0 for the mode registers defined in this table.

5 Mode Registers (cont'd)

Table 11 — Mode Register 0 (MR0)

Field	Bits	Description	Notes
Test Mode (TM)	OP[7]	0 – Normal Operation (Default) 1 – Test Mode (Vendor specific): only to be used by the DRAM manufacturer. No functional operation is specified with test mode enabled.	
Command Address Parity (CAPAR)	OP[6]	0 – Disabled (Default) 1 – Enabled	1
Write Parity (WPAR)	OP[5]	0 – Disabled 1 – Enabled	2
Read Parity (RPAR)	OP[4]	0 – Disabled 1 – Enabled	2
DRFM	OP[3]	0 – Disabled (Default) 1 – Enabled	
Temperature Compensated Self Refresh (TCSR)	OP[2]	0 – Disabled 1 – Enabled (Default)	
Write DBI (WDBI)	OP[1]	0 – Disabled 1 – Enabled	3
Read DBI (RDBI)	OP[0]	0 – Disabled 1 – Enabled	3

NOTE 1 Refer to the Command/Address Parity section for details regarding CA Parity.
NOTE 2 Refer to the Data Parity section for details regarding Write Parity and Read Parity.
NOTE 3 Refer to the Data Bus Inversion (DBIac) section for details regarding WDBI and RDBI.

Table 12 — Mode Register 1 (MR1)

Field	Bits	Description	Notes
Parity Latency (PL)	OP[7:5]	000 – 0 nCK 001 – 1 nCK 010 – 2 nCK 011 – 3 nCK 100 – 4 nCK All others – Reserved	1, 2
Write Latency (WL)	OP[4:0]	00100 – 4 nCK 00101 – 5 nCK 00110 – 6 nCK … 10010 – 18 nCK 10011 – 19 nCK All Others - Reserved	1, 3

NOTE 1 All PL and WL values are optional, however the supported min-to-max ranges must be contiguous.
NOTE 2 Refer to the Data Parity section for details regarding Parity Latency (PL) definition and use with write and read operations.
NOTE 3 Refer to the WRITE command section for details regarding the Write Latency (WL) definitions and use.

5 Mode Registers (cont'd)

Table 13 — Mode Register 2 (MR2)

Field	Bits	Description	Notes
Read Latency (RL)	OP[7:0]	00010001 – 17 nCK 00010010 – 18 nCK 00010011 – 19 nCK … 01011001 – 89 nCK 01011010 – 90 nCK All others – Reserved	1, 2

NOTE 1 All RL values are optional, however the supported min-to-max ranges must be contiguous.
NOTE 2 Refer to the READ command section for details regarding the Read Latency (RL) definitions and use.

Table 14 — Mode Register 3 (MR3)

Field	Bits	Description	Notes
Write Recovery to Auto Precharge (WR)	OP[7:0]	00000100 – 4 nCK 00000101 – 5 nCK 00000110 – 6 nCK … 00111110 – 62 nCK 00111111 – 63 nCK All others – Reserved	1, 2

NOTE 1 All WR values are optional, however the supported min-to-max range must be contiguous.
NOTE 2 WR must be programmed with a value greater than or equal to RU{t_{WR}/t_{CK}}, where RU stands for round up, t_{WR} is the analog value from the vendor datasheet and t_{CK} is the operating clock cycle time. If an HBM4 DRAM does not support the mode register definition of t_{WR} in clock cycles, the WR mode register settings will be ignored.

Table 15 — Mode Register 4 (MR4)

Field	Bits	Description	Notes
Activate to Precharge (RAS)	OP[7:0]	00000100 – 4 nCK 00000101 – 5 nCK 00000110 – 6 nCK … 00111110 – 62 nCK 00111111 – 63 nCK All others – Reserved	1, 2

NOTE 1 All RAS values are optional, however the supported min-to-max range must be contiguous.
NOTE 2 RAS must be programmed with a value greater than or equal to RU{t_{RAS}/t_{CK}}, where RU stands for round up, t_{RAS} is the analog value from the vendor datasheet and t_{CK} is the operating clock cycle time. If an HBM4 DRAM does not support the mode register definition of t_{RAS} in clock cycles, the RAS mode register settings will be ignored.

5 Mode Registers (cont'd)

Table 16 — Mode Register 5 (MR5)

Field	Bits	Description	Notes
RFU	OP[7:4]	0000	
Read to Auto Pre-charge (RTP)	OP[3:0]	0010 – 2 nCK 0011 – 3 nCK 0100 – 4 nCK … 1110 – 14 nCK 1111 – 15 nCK All others – Reserved	1, 2

NOTE 1 All RTP values are optional, however the supported min-to-max range must be contiguous.

NOTE 2 RTP must be programmed with a value greater than or equal to RU{t_{RTP}/t_{CK}}, where RU stands for round up, t_{RTP} is the analog value from the vendor datasheet and t_{CK} is the operating clock cycle time. If an HBM4 DRAM does not support the mode register definition of t_{RTP} in clock cycles, the RTP mode register settings will be ignored.

Table 17 — Mode Register 6 (MR6)

Field	Bits	Description	Notes
Duty Cycle Monitor (DCM) Flip	OP[7]	0 – Disabled (Default) 1 – Enabled	
Duty Cycle Monitor (DCM)	OP[6]	0 – Disabled (Default) 1 – Enabled	
Pullup Driver Strength	OP[5:3]	000 – 25Ohm 001 – 20Ohm 010 – 16.7Ohm (Default) 011 – 14.3Ohm All others – Reserved	1
Pulldown Driver Strength	OP[2:0]	000 – 25Ohm 001 – 20Ohm 010 – 16.7Ohm (Default) 011 – 14.3Ohm All others – Reserved	1

NOTE 1 Refer to the Transmit Driver Resistance (Table 91) table for the details.

5 Mode Registers (cont'd)

Table 18 — Mode Register 7 (MR7)

Field	Bits	Description	Notes
CATTRIP	OP[7]	0 – CATTRIP pin drives a LOW or HIGH depending on CATTRIP sensor output (Default) 1 – CATTRIP pin drives a static HIGH	1
RFU	OP[6]	0	
DWORD MISR Control	OP[5:3]	The bits are only evaluated if DWORD Loop-back is enabled in OP0 000 – Preset: the DWORD MISR is preset as described in the HBM4 Loopback Test Modes section, and all DWORD LFSR_COMPARE_STICKY bits are reset to 0. 001 – LFSR mode (READ direction) 010 – Register mode (WRITE and READ directions): DWORD writes are captured directly into the MISR without compression. The MISR will contain the most recent write data. 011 – MISR mode (WRITE direction) 100 – LFSR Compare mode (WRITE direction) All others - Reserved	2, 3
RFU	OP[2]	0	
DWORD Read Mux Control	OP[1]	The bit is only evaluated with READ commands and if DWORD Loopback is enabled in OP0 0 – Return data from DWORD MISR (see OP[5:3]) 1 – Return LFSR_COMPARE_STICKY bits (OP[5:3] is ignored)	2, 3
DWORD Loopback	OP[0]	0 – Disabled (Default) 1 – Enabled: all Writes and Reads will be to/from the MISR. Notes: a) does not require any row activation b) column addresses associated with WRITE and READ commands are ignored	2

NOTE 1 The CATTRIP pin can be asserted to "HIGH" from any of the channels [31:0] MR7 OP7 bit (logic OR).
NOTE 2 See HBM4 Loopback Test Modes for DWORD MISR mode features and use.
NOTE 3 Refer to Table 19 for details on DWORD MISR operation with WRITE and READ commands.

5 Mode Registers (cont'd)

Table 19 — DWORD MISR Read and Write Operations in Loopback Test Mode (MR7 OP0=1)

MR7 OP[1]	MR7 OP[5:3]	DWORD MISR Operation[1]		Comments
		WRITE	READ	
0	000 (Preset)	Write data are ignored	Read the Preset value (clock-like pattern)	Neither Writes nor Reads alter the MISR content
	001 (LFSR)	Write data are ignored	Generate read data from LFSR	Writes do not alter the MISR content
	010 (Register)	MISR stores the second half (UI 4 to 7) or all data (UI 0 to 7) of the most recent Write from register mode, with only (UI 4 to 7) readable via WDR (see note 2)	Read the MISR content (UI 0 to 3 and repeated for UI 4 to 7, or all data (UI 0 to 7) of the most recent Write (see note 2))	Reads do not alter the MISR content
	011 (MISR)	Write data are accumulated in the MISR		
	100 (LFSR Compare)	Write data are compared against data generated by the LFSR		
1	XXX	Write data are ignored	Read sticky error bits	Neither Writes nor Reads alter the MISR content

NOTE 1 See Loopback Test Modes for DWORD MISR and LFSR features and use.
NOTE 2 Depending on implementation, the MISR either stores the second half (UI 4 to 7) or all data (UI 0 to 7) to the most recent Write, and subsequent Reads return either the second half (UI 4 to 7) or all data (UI 0 to 7) to that most recent Write. If a Read shall send identical data regardless of the actual implementation, users should send the same write data on UI 0 to 3 and UI 4 to 7 of the most recent write.

Table 20 — Mode Register 8 (MR8)

Field	Bits	Description	Notes
DRFM Bounded Refresh Configuration (BRC)	OP[7:6]	00 – Always ±1, Ratio ±2 (tDRFM = 4 x tRRF) 01 – Always ±1, ±2, Ratio ±3 (tDRFM = 6 x tRRF) 10 – Always ±1, ±2, ±3, Ratio ±4 (tDRFM = 8 x tRRF) 11 – RFU	1
RFM Levels (RFML)	OP[5:4]	00 – Default Level (RFM may be required or not) 01 – Level A (RFM is required) 10 – Level B (RFM is required) 11 – Level C (RFM is required)	2
WDQS-to-CK Training (WDQS2CK)	OP[3]	0 – Disabled (Default) 1 – Enabled	3
ECS error log auto reset (ECSLOG)	OP[2]	0 – Disabled (Default) 1 – Enabled	
Rx Offset Calibration Start/Stop	OP[1]	0 – Stop (Default) 1 – Start	1
DA Port Lockout	OP[0]	0 – Access to DA port is enabled (Default) 1 – Access to DA port is locked	4

NOTE 1 01 and 10 of BRC Configurations are optional features whereas 00 is required. Rx Offset Calibration is also optional features and must set to 0 if it is unavailable. Supportability of these optional features should be consulted by vendor datasheets and Table 126 (DEVICE_ID_WDR)

NOTE 2 The support of Adaptive Refresh Management (ARFM) is optional for the DRAM vendor. HBM4 DRAMs not supporting (ARFM) will define these bits as RFU. RAAIMT, RAAMMT and RAADEC values for default RFM level and RFM levels A to C are set by DRAM vendor and can be read via the IEEE1500 DEVICE_ID WDR.

NOTE 3 Refer to the WDQS-to-CK Alignment Training section for details.

NOTE 4 DA Port Lockout bit is defined for channels 0 and 4 only. The bit is RFU for all other channels. Once enabled, the bit can only be cleared by powering off the device. The IEEE1500 MODE_REGISTER_DUMP_SET instruction cannot be used to set or clear the bit, but allows reading the bit.

5 Mode Registers (cont'd)

Table 21 — Mode Register 9 (MR9)

Field	Bits	Description	Notes
ECS Error Type and Address Reset (ECSRES)	OP[7]	0 – Maintain the ECS error type and address log (Default) 1 – Reset the ECS error type and address log (self-clearing)	1
ECS multi-bit error correction (ECSCEM)	OP[6]	0 – Correction of multi-bit errors during ECS cycles is disabled 1 – Correction of multi-bit errors during ECS cycles is enabled	
Auto ECS during Self Refresh (ECSSRF)	OP[5]	0 – Auto ECS during self refresh mode is disabled (Default) 1 – Auto ECS during self refresh mode is Enabled	2
Auto ECS via REFab (ECSREF)	OP[4]	0 – Auto ECS via REFab command is disabled (Default) 1 – Auto ECS via REFab command is enabled	2, 3
Error Vector Pattern (ECCVEC)	OP[3]	The bit is only evaluated when ECC Vector Input Mode is enabled in OP2 0 – Codeword 0 (CW0): Data '1' means error bit and data '0' means non-error bit 1 – Codeword 1 (CW1): Data '0' means error bit and data '1' means non-error bit	
Error Vector Input Mode (ECCTM)	OP[2]	0 – ECC Engine Test Mode is disabled (default) 1 – ECC Engine Test Mode is enabled	
Severity Reporting (SEVR)	OP[1]	0 – Error severity reporting is disabled and the SEV signals are High-Z 1 – Error severity reporting is enabled. The SEV signals drive error severity information during Reads and otherwise are High-Z.	4
Meta Data (MD)	OP[0]	0 – ECC signals are disabled. Read and write operations do not include meta data 1 – ECC signals are enabled. Read and write operations include meta data transmitted via ECC pins	

NOTE 1 The bit is self-clearing meaning that it automatically returns back to 0 after the reset function has been issued.
NOTE 2 For ECS operation either ECSSRF or ECSREF (or both) must be enabled.
NOTE 3 When ECS during REFab is enabled, the host must issue REFab commands at an average rate of t_{ECSint}.
NOTE 4 Input data on SEV signals during write operations will be ignored regardless of the SEVR setting.

Table 22 — Mode Register 10 (MR10)

Field	Bits	Description	Notes
DCA Code for RDQS1 (PC1)	OP[7:4]	TBD	
DCA Code for RDQS0 (PC0)	OP[3:0]	TBD	

5 Mode Registers (cont'd)

Table 23 — Mode Register 11 (MR11)

Field	Bits	Description	Notes
DCA code for WDQS1 (PC1)	OP[7:4]	0000 – 0 steps (Default; no correction) 0001 – -1 step 0010 – -2 steps ... 0110 – -6 steps 0111 – -7 steps 1000 – Reserved 1001 – +1 step 1010 – +2 steps ... 1110 – +6 steps 1111 – +7 steps	1, 2
DCA code for WDQS0 (PC0)	OP[3:0]	0000 – 0 steps (Default; no correction) 0001 – -1 step 0010 – -2 steps ... 0110 – -6 steps 0111 – -7 steps 1000 – Reserved 1001 – +1 step 1010 – +2 steps ... 1110 – +6 steps 1111 – +7 steps	1, 2

NOTE 1 Values of 0001 to 0111 decrease the internal WDQS duty cycle, and values of 1001 to 1111 increase the internal WDQS duty cycle.
NOTE 2 The step size (in ps) is vendor specific and may be non-linear.

Table 24 — Mode Register 12 (MR12)

Field	Bits	Description	Notes
Reserved for Vendor Specific Features	OP[7:0]	Vendor Specific	1

NOTE 1 MR12 is reserved for vendor specific features. Refer to the vendor's datasheet for details.

Table 25 — Mode Register 13 (MR13)

Field	Bits	Description	Notes
Reference voltage for AWORD inputs (VREFCA)	OP[6:1]	000000 – 0.18 x V_{DDQL} 000001 – 0.19 x V_{DDQL} … 011111 – 0.49 x V_{DDQL} 100000 – 0.50 x V_{DDQL} (Default) 100001 – 0.51 x V_{DDQL} … 111110 – 0.80 x V_{DDQL}	

		111111 – 0.81 x V_{DDQL}	
NOTE 1 Refer to the DC and AC Operating Conditions section for the AWORD input receiver voltage level specification.			

Table 26 — Mode Register 14 (MR14)

Field	Bits	Description	Notes
RFU	OP[7]	0	
Reference voltage for DWORD inputs (VREFD)	OP[6:1]	000000 – 0.18 x V_{DDQL} 000001 – 0.19 x V_{DDQL} … 011111 – 0.49 x V_{DDQL} 100000 – 0.50 x V_{DDQL} (Default) 100001 – 0.51 x V_{DDQL} … 111110 – 0.80 x V_{DDQL} 111111 – 0.81 x V_{DDQL}	
RFU	OP[0]	0	
NOTE 1 Refer to the DC and AC Operating Conditions section for the AWORD input receiver voltage level specification.			

Table 27 — Mode Register 15 (MR15)

Field	Bits	Description	Notes
RFU	OP[7:4]	0000	
DFE Code (PC1)	OP[3:2]	00 – Disable (Default) 01 – Step + 1 10 – Step + 2 11 – Step + 3	1
DFE Code (PC0)	OP[1:0]	00 – Disable (Default) 01 – Step + 1 10 – Step + 2 11 – Step + 3	1
NOTE 1 Size of each Decision Feedback Equalizer (DFE) steps are vendor specific. Refer to vendor datasheets			

Table 28 — Mode Register 16 (MR16)

Field	Bits	Description	Notes
DFE	OP[7:0]	TBD	

Table 29 — Mode Register 17 (MR17)

Field	Bits	Description	Notes
Vendor Specific	OP[7:0]	Vendor Specific	1
NOTE 1 MR17 is reserved for vendor specific features. Refer to the vendor's datasheet for details.			

Table 30 — Mode Register 18 (MR18)

Field	Bits	Description	Notes
RFU	OP[7:0]	00000000	

Table 31 — Mode Register 19 (MR19)

Field	Bits	Description	Notes
RFU	OP[7:0]	00000000	

6 Operation

6.1 HBM4 Clocking Overview

The HBM device captures commands and addresses on the row and column buses using a differential clock CK_t/CK_c. Both buses operate at double data rate (DDR).

The HBM device has uni-directional differential Write strobes (WDQS_t/WDQS_c) and Read strobes (RDQS_t/RDQS_c) per 32 DQ(DWORD). The data bus operates at double data rate (DDR).

HBM4 utilizes two types of clock with different frequencies. The strobe frequency is twice the frequency of the command clock, requiring an HBM4 to have reset-type clock-divider in the WDQS clock tree (Figure 10). By dividing the WDQS, the operation speed of DRAM internal circuits in WDQS domain is reduced to half. The direction of the internal WDQS/2 transition may vary depending on vendor's choice. Command clock and WDQS are generated from the same PLL and RDQS clock is generated from WDQS. WDQS internal divider is initialized to be a pre-defined internal divider state after Self Refresh exit or Power-up or Power down exit sequence. The sum of preamble and postamble for both READ and WRITE operation is required to be an even number so that the internal divider's state, phase of internal WDQS/2, is maintained. Therefore, HBM4 WDQS does not require a specific sync operation before READ and WRITE operations. WDQS starts toggling before starting WRITE or READ operations for reducing ISI. During inactivity, WDQS/ RDQS are required to be static (WDQS/RDQS_t is Low, WDQS/RDQS_c is High). When WRITE training for unmatched DQ/DQS path, DQ should be shifted to align phase to the point where CK and WDQS are in sync.

The following nomenclature is being used throughout this standard:

- a rising CK (or WDQS, RDQS) edge is defined as the crossing of the positive edge of CK_t (or WDQS_t, RDQS_t) and the negative edge of CK_c (or WDQS_c, RDQS_c);
- a falling CK (or WDQS, RDQS) edge is defined as the crossing of the negative edge of CK_t (or WDQS_t, RDQS_t) and the positive edge of CK_c (or WDQS_c, RDQS_c).

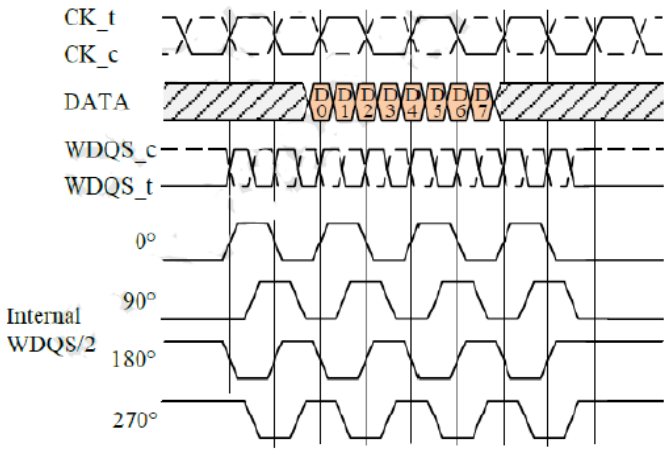
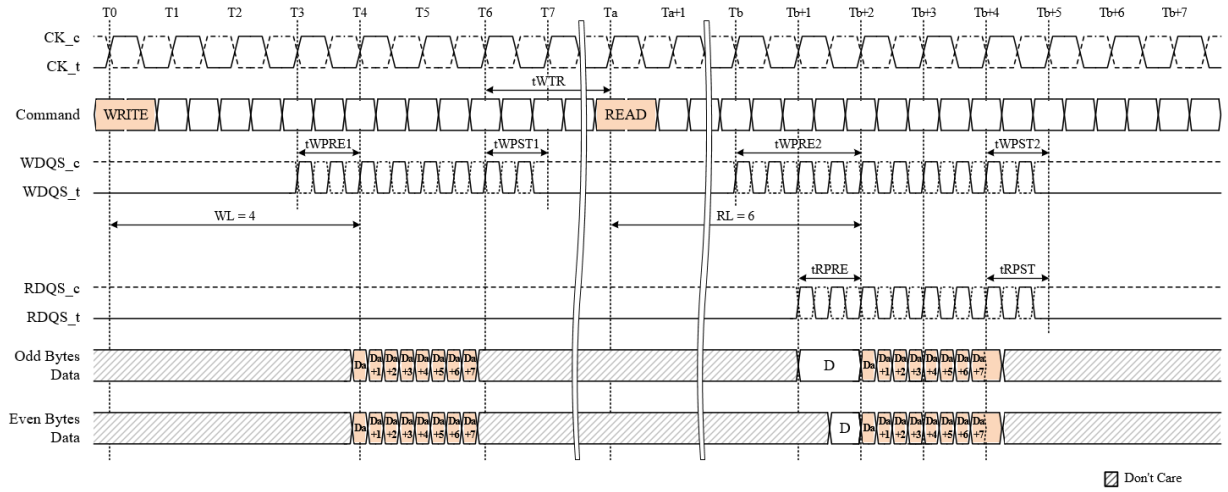


Figure 8 — Aligned WDQS Internal Divider Example

6.1 HBM4 Clocking Overview (cont'd)



NOTE 1 tWPRE1 : Write preamble of WDQS, tWPST1 : Write postamble of WDQS
NOTE 2 tWPRE2 : Read preamble of WDQS, tWPST2 : Read postamble of WDQS
NOTE 3 tRPRE : Read preamble of RDQS, tRPST : Read postamble of RDQS

Figure 9 — Clocking and Interface Relationship Write to Read Timing

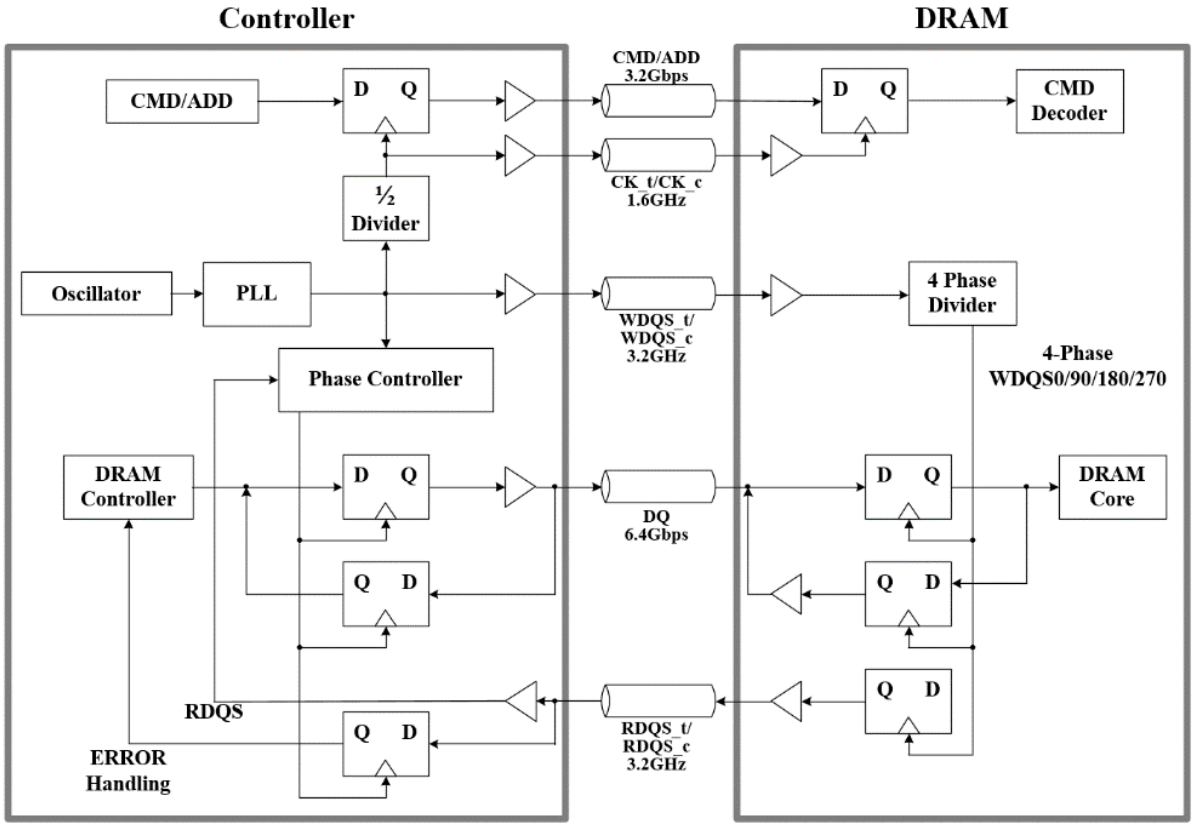


Figure 10 — High Level Block Diagram Example of Clocking Scheme

6.1.1 WDQS-to-CK Alignment Training

WDQS-to-CK alignment training allows the host to observe the phase offset between the WDQS strobes in both PCs and the CK clock to aid in keeping the phase relationship within the limits given by the t_{DQSS} specification. The WDQS2CK bit in MR8 OP3 is associated with this training mode.

WDQS-to-CK alignment training is required to be performed at least once after device initialization if adherence to the t_{DQSS} timing cannot be guaranteed without performing this alignment training. WDQS-to-CK alignment training is not required if the t_{DQSS} timing is met. An effort to further narrow the WDQS-to-CK phase offset by using this training mode will not improve the stable device operation.

Steps 1 through 7 are required for WDQS-to-CK alignment training:

1. Enter WDQS-to-CK alignment training mode by setting the WDQS2CK bit to 1 and wait t_{MOD} Commands allowed while in this mode are REFab, REFpb, RFMab, RFMpb, RNOP, CNOP and MRS to exit WDQS-to-CK alignment training. Internal current spikes generated by the use of REFab, REFpb, RFMab and RFMpb commands in this mode may negatively impact the training result. Controllers that cannot account for this impact should avoid use of REFab, REFpb, RFMab and RFMpb commands in this mode.
2. Enable both WDQS0 and WDQS1 strobes; keep both strobes constantly running in order to generate a valid read-out at both phase detectors with each CK clock cycle;
3. Slowly sweep the WDQS0 and WDQS1 phases with respect to the CK clock, and monitor both DERR0 and DERR1 signals for the phase detector's result as shown in Table 32 and Figure 11; each phase detector latches the 0° phase of the internally divided WDQS strobe (0° phase) with each rising CK clock edge and provides the result on the DERR0 signal for WDQS0 and the DERR1 signal for WDQS1 after $t_{WDQS2PD}$;
4. After a minimum of 8 WDQS pulses have been received, the strobes may be halted at any time while WDQS-to-CK alignment training mode is enabled; the phase detector does not provide a valid read-out in this case and it’s result on the DERR signals should be ignored;
5. The ideal alignment is indicated by the phase detector output transitioning from "early" to "late" when the delay of the WDQS phase is continuously increased;
6. When the phase relationship between WDQS and CK meets the t_{DQSS} specification, stop both WDQS strobes; ensure that the number of WDQS pulses issued while in this training mode is an even number such that the internal WDQS state is back at its reset state once the training has finished. With that, no specific synchronization between CK and WDQS is required for correct write and read operation;
7. Exit WDQS-to-CK alignment training mode by setting the WDQS2CK bit to 0 and wait t_{MOD}.

Table 32 — Phase Detector and DERR Signal Behavior

Internal WDQS/2 (0° Phase) Sampled By CK	WDQS Phase	DERR0 DERR1	Recommended Action
HIGH	Early	HIGH	Increase delay on WDQS
LOW	Late	LOW	Decrease delay on WDQS

6.1.1 WDQS-to-CK Alignment Training (cont'd)

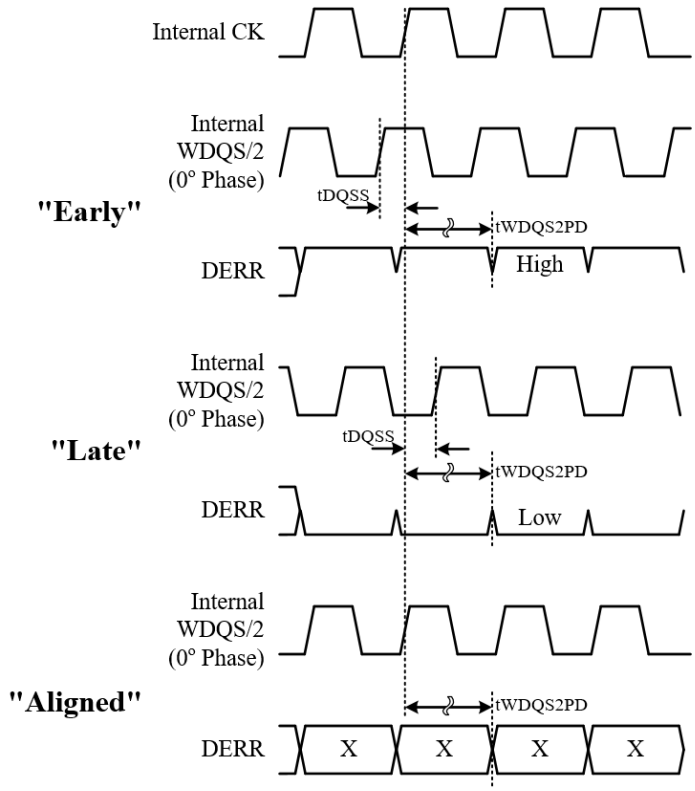


Figure 11 — DERR Signal Behavior in WDQS-to-CK Alignment Training

6.2 HBM4 Data Bus Inversion (DBIac)

6.2.1 Data Bus Inversion (DBIac)

HBM4 DRAMs supports a byte granular Data Bus Inversion (DBIac). The corresponding DBI signal is a DDR I/O and driven or sampled along with the DQs for read and write operations.

The word DBI refers to the internal state of the device unless explicitly noted as DBI signal. The DBIac function can be enabled or disabled independently for writes per MR0 OP1 (WDBI) and for reads per MR0 OP0 (RDBI).

The DBI input is a Don't care and the DBI input receivers are disabled when WDBI is disabled. The DBI output buffers are turned off when RDBI is disabled.

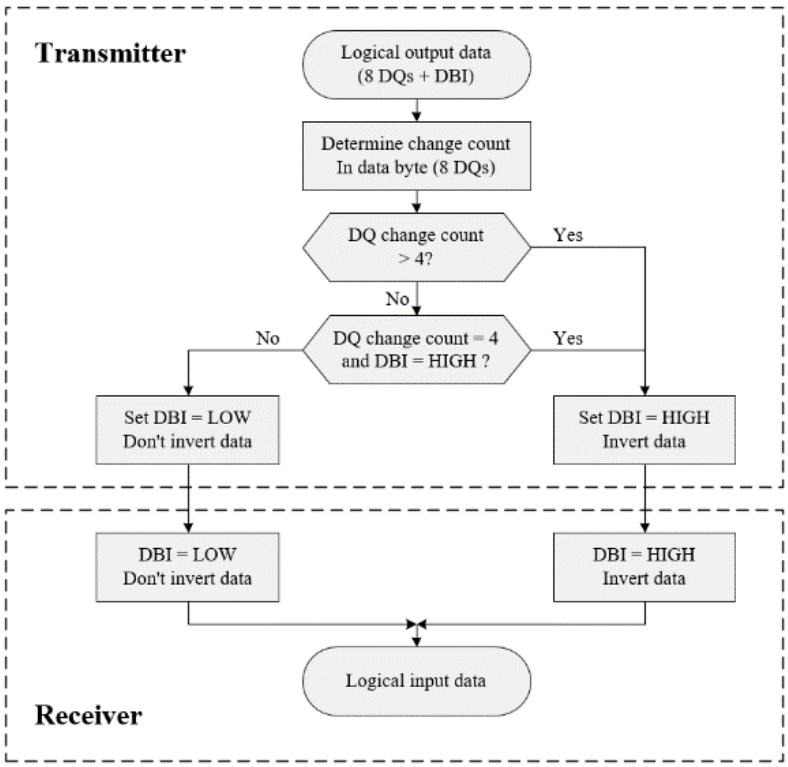


Figure 12 — DBIac Algorithm

Write operation: the HBM4 DRAM inverts write data received on the DQ inputs in case DBI is sampled HIGH, or leaves the write data non-inverted in case DBI is sampled LOW. Note that the ECC inputs are not affected by the DBIac function.

Read operation: the HBM4 DRAM counts the number of DQ signals that are transitioning from the previous state. Note that the ECC and SEV outputs are not affected by DBIac. See Internal DBIac States with Read for bus pre-conditioning. The HBM4 DRAM inverts read data and sets DBI HIGH when the number of transitioning data bits within a byte is greater than 4, or when the number of transitioning data bits within a byte equals 4 and DBI was HIGH; otherwise the HBM4 DRAM does not invert the read data and sets DBI LOW.

6.2.1 Data Bus Inversion (DBIac) (cont'd)

Table 33 — DBI(ac) Truth Table

DQ Charge Count	Previous DBI State	New DBI State	New DQ State
0 to 3	X	LOW	Not inverted
4	LOW		
	HIGH	HIGH	Inverted
5 to 8	X		

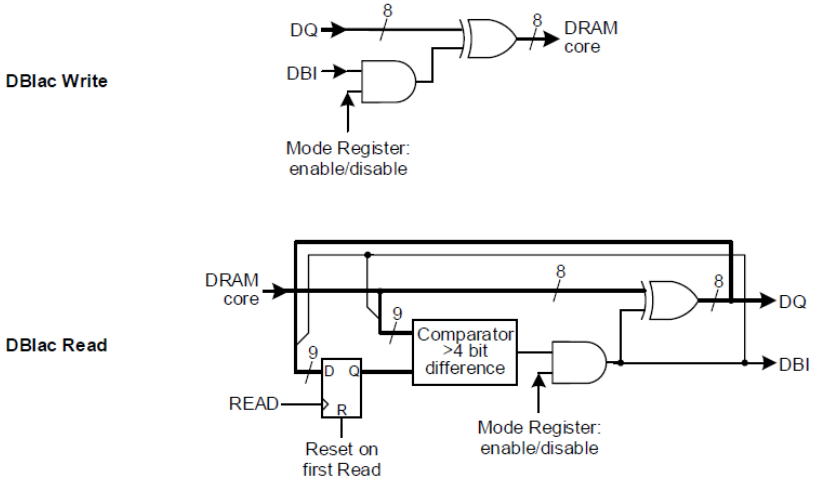


Figure 13 — Example DBIac Logic for Write and Read

6.2.1.1 Internal DBIac State with Read

The HBM4 DRAM resets the internal DBIac state to LOW whenever any of the following events occur:

- RESET_n signal de-assertion;
- a MODE REGISTER SET (MRS) command is received;
- a write-to-read bus turnaround;
- Self Refresh exit

For all other events or commands, the internal DBIac state is not reset to LOW and the HBM4 DRAM will use its previous state for DBIac calculation.

First Read Command:

When a first READ command is registered after a DBI reset, the HBM4 DRAM preconditions the bus to LOW prior to read data regardless whether RDBI is enabled or disabled in the mode register, as shown in Figure 14 in case of a write-to-read bus turnaround. The internal state D7 corresponding to the last UI of the read burst is internally stored as a seed value for a subsequent read burst.

The DPAR signal is not included in the DBI calculation and not preconditioned to LOW; its initial state is undefined (LOW or HIGH).

6.2.1.1 Internal DBIac State with Read (cont'd)

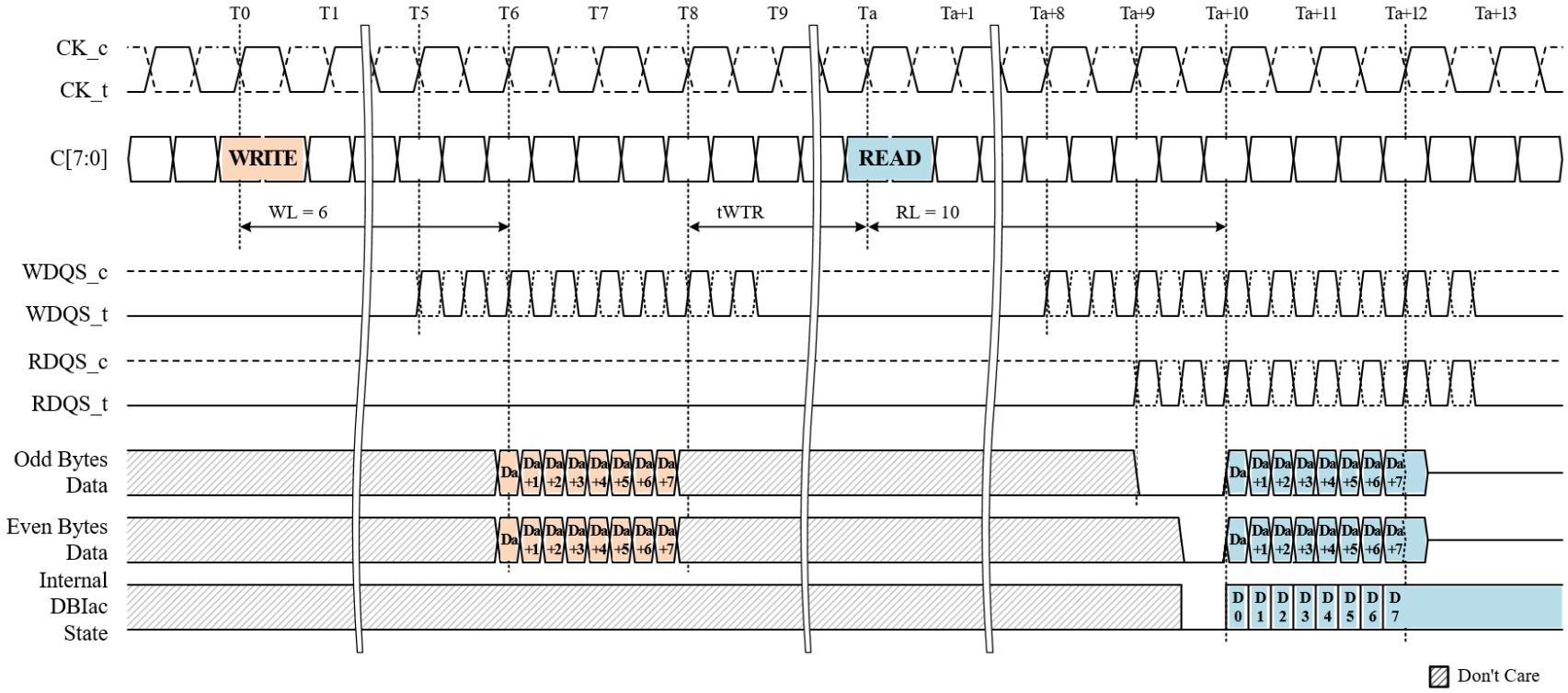


Figure 14 — Internal DBIac State Reset for Write to Read

6.2.1.2 Internal DBIac State with Consecutive Read Commands (Seamless and non-seamless)

Once the Read burst is complete, the HBM4 DRAM tri-states all DQ, DBI and ECC output drivers. However, the HBM4 DRAM internally stores the last data-out of the DQ, DBI, ECC and SEV outputs to pre-condition the bus prior to a subsequent read; it also uses the last data-out of the DQ and DBI outputs for DBIac calculation for any subsequent read operation barring a condition to DBI reset. For non-gapless read operations, the HBM4 DRAM pre-conditions all data outputs to the last data-out of the previous burst nominally two WDQS cycles (odd bytes) and one WDQS cycle (even bytes) prior to the first valid data bit as shown in Figure 15.

6.2.1.2 Internal DBIac State with Consecutive Read Commands (Seamless and non-seamless) (cont'd)

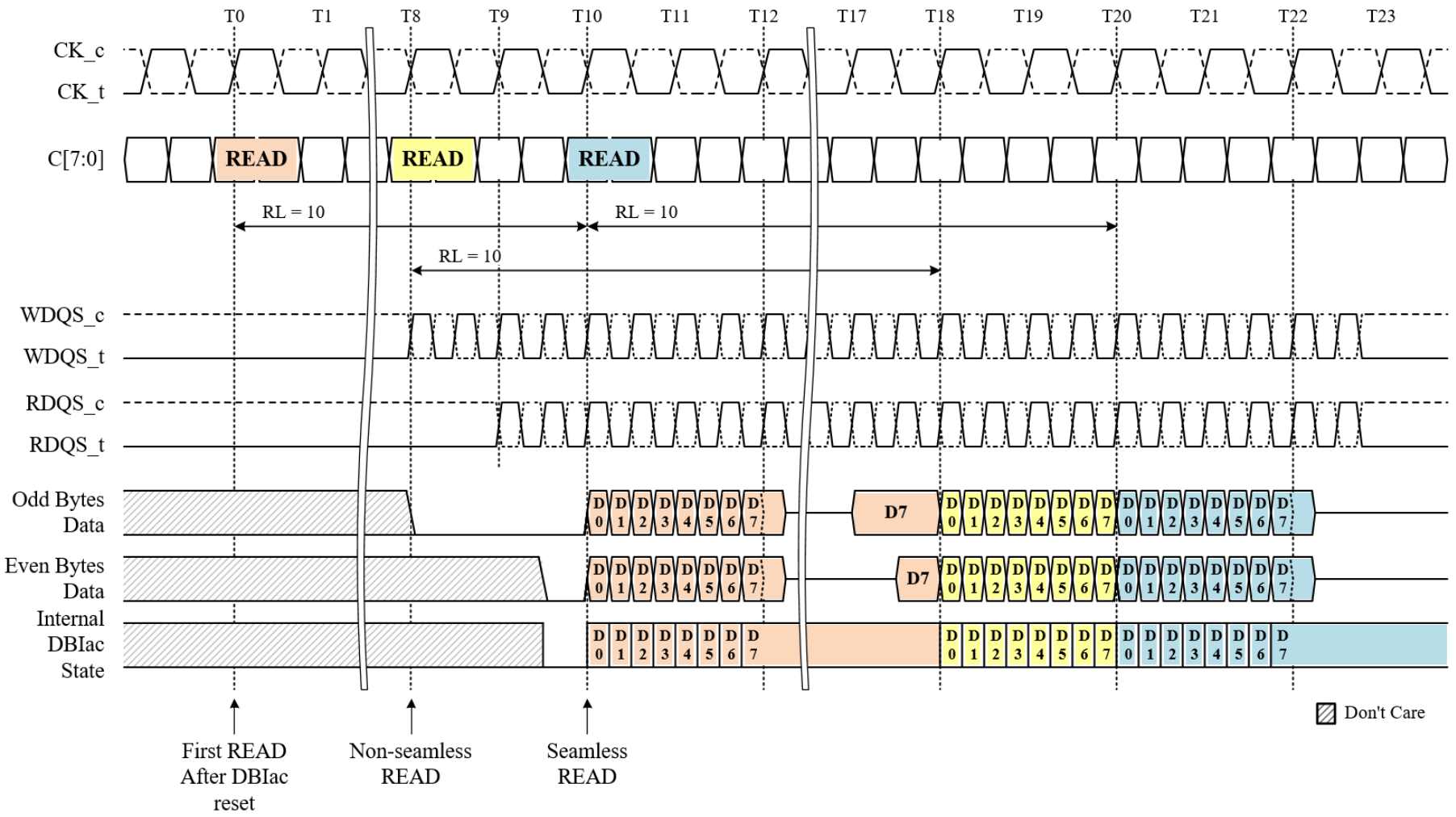


Figure 15 — Bus Preconditioning and DBI States for Read

6.3 Commands

The HBM4 DRAM features DDR commands entered on both rising and falling CK clock edges. Row Activate commands require one-and-a-half-cycle and other row commands require only a half-cycle except for PDE and SRE with one cycle. Column commands require only one cycle.

The command interface includes a reserved DDR input signal ARFU which is omitted from subsequent truth tables but required to be driven to a valid signal level along with the other AWORD inputs.

6.3.1 Command Truth Tables

Table 34 — Row Commands Truth Table

Command [4]	Symbol	Clock Cycle	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	Notes
Row No Operation	RNOP	R or F	H	H	H	H	V	V	V	V	V	V	1, 2, 3
Activate	ACT	R	L	H	H	PC	SID0 / V	SID1 / V	BA0	BA1	BA2	BA3	1, 2, 3, 5, 6
		F	H	H	RA8	RA9	RA10	RA11	RA12	RA13	RA14 / V	DRFM	
		R	H	H	RA0	RA1	RA2	RA3	RA4	RA5	RA6	RA7	
Precharge per-bank	PREpb	R or F	H	L	L	PC	SID0 / V	SID1 / V	BA0	BA1	BA2	BA3	1, 2, 3, 5, 6
Precharge all-bank	PREab	R or F	H	L	H	PC	V	V	V	V	V	V	1, 2, 3, 5
Refresh Per-Bank	REFpb	R	L	L	L	PC	SID0 / V	SID1 / V	BA0	BA1	BA2	BA3	1, 2, 3, 5, 6
Refresh All-Bank	REFab	R	H	H	L	PC	V	V	V	V	L	V	1, 2, 3, 5
Refresh Management Per-Bank	RFMpb	R	L	L	H	PC	SID0 / V	SID1 / V	BA0	BA1	BA2	BA3	1, 2, 3, 5, 6, 7
Refresh Management All-Bank	RFMab	R	H	H	L	PC	V	V	V	V	H	V	1, 2, 3, 5, 7
Power-Down Entry	PDE	R	L	H	L	H	V	V	V	V	V	V	1, 2, 3
		F	L	H	L	H	V	V	V	V	V	V	
Self Refresh Entry	SRE	R	L	H	L	L	V	V	V	V	V	V	1, 2, 3
		F	L	H	L	L	V	V	V	V	V	V	
Power-Down and Self Refresh Exit	PDX/ SRX	R	H	H	H	H	V	V	V	V	V	V	1, 2, 8

NOTE 1 BA = Bank Address; RA = Row Address; PC = Pseudo Channel 0 or 1; SID = Stack ID; V = Valid Signal (either H or L, but not floating);

NOTE 2 R[9:0] must be driven to a valid signal level even if a stack ID address (SID) or row address (RA) is not defined for a specific density. APAR must be driven to a valid signal level even if CA parity is disabled in MR0 OP6.

NOTE 3 Parity is evaluated on all pins if CA parity is enabled in MR0 OP6.

NOTE 4 All other command encodings not shown in the table are reserved for future use.

NOTE 5 PC = 0 selects pseudo channel 0 (PC0), and PC = 1 selects pseudo channel 1 (PC1). The pseudo channel not selected by PC performs a RNOP.

NOTE 6 The SID bits act as bank address bits in conjunction with ACT, PREpb, REFpb and RFMpb commands, and related timing diagrams shall be interpreted accordingly. All other row commands do not use SID. Refer to the channel addressing table for HBM4 configurations using SID.

NOTE 7 An HBM4 DRAM not requiring refresh management (RFM) will execute an RNOP command instead of RFMab or RFMpb.

NOTE 8 Parity is not checked at Power-Down Exit or Self Refresh Exit. The HBM4 device requires RNOP and CNOP commands on Row and Column bus respectively with valid parity if CA parity is enabled during the power-down exit period (t_{XP}) and self refresh exit period (t_{XS}).

NOTE 9 ACT is a 1.5 cycle command. Only a RNOP command, a PREpb command to a different bank or a PREab to a different PC are allowed following an ACT command on the falling edge of the second cycle.

6.3.1 Command Truth Tables (cont'd)

Table 35 — Column Commands Truth Table

Command [4]	Symbol	Clock Cycle	C0	C1	C2	C3	C4	C5	C6	C7	Notes
Column No Operation	CNOP	R	H	H	H	V	V	V	V	V	1, 2, 3
		F	V	V	V	V	V	V	V	V	
Read	RD	R	H	L	H	L	PC	SID0 / V	SID1 / V	BA0	1, 2, 3, 5, 6, 7
		F	BA1	BA2	BA3	CA0	CA1	CA2	CA3	CA4	
Read w/ AP	RDA	R	H	L	H	H	PC	SID0 / V	SID1 / V	BA0	1, 2, 3, 5, 6, 7
		F	BA1	BA2	BA3	CA0	CA1	CA2	CA3	CA4	
Write	WR	R	H	L	L	L	PC	SID0 / V	SID1 / V	BA0	1, 2, 3, 5, 6
		F	BA1	BA2	BA3	CA0	CA1	CA2	CA3	CA4	
Write w/ AP	WRA	R	H	L	L	H	PC	SID0 / V	SID1 / V	BA0	1, 2, 3, 5, 6
		F	BA1	BA2	BA3	CA0	CA1	CA2	CA3	CA4	
Mode Register Set	MRS	R	L	L	L	MA4	OP5	OP6	OP7	MA0	1, 3, 8, 9
		F	MA1	MA2	MA3	OP0	OP1	OP2	OP3	OP4	

NOTE 1 BA = Bank Address; CA = Column Address; PC = Pseudo Channel 0 or 1; SID = Stack ID; MA = Mode Register Address; V = Valid Signal (either H or L, but not floating).

NOTE 2 C[7:0] must be driven to a valid signal level even if a stack ID address (SID) is not defined for a specific density, or if parity is disabled in the mode register. APAR must be driven to a valid signal level even if CA parity is disabled in MR0 OP6. C[7:0] are Don't Care when the device is in power-down or self refresh.

NOTE 3 Parity is evaluated on all pins if CA parity is enabled in MR0 OP6.

NOTE 4 All other command encodings not shown in the table are reserved for future use.

NOTE 5 PC = 0 selects pseudo channel 0 (PC0), and PC = 1 selects pseudo channel 1 (PC1). The pseudo channel not selected by PC performs a CNOP.

NOTE 6 The SID bits act as bank address bits in conjunction with READ and WRITE commands, and related timing diagrams shall be interpreted accordingly. All other column commands do not use SID. Refer to the channel addressing table for HBM4 configurations using SID.

NOTE 7 HBM4 configurations using the SID specify a timing parameter t_{CCDR} for consecutive READs to different SID. Vendor datasheets should be consulted for details.

NOTE 8 All mode registers are write-only by default using the MRS command.

NOTE 9 Refer to the HBM4 Mode Register Overview table for MA4 of MRS.

6.3.1 Command Truth Tables (cont'd)

Table 36 — Options for issuing PREab and PREpb commands

Command on Rising Clock Edge	Allowed PREab/PREpb Command(s) on Falling Clock Edge (Same Cycle)		
	Same PC, Same Bank	Same PC, Different Bank	Different PC, Any Bank
RNOP	PREab, PREpb		PREab, PREpb
ACT	--	PREpb	PREab, PREpb
PREab	--	--	PREab, PREpb
PREpb	--	PREpb	PREab, PREpb
REFab	--	--	PREab, PREpb
REFpb	--	PREpb	PREab, PREpb
RFMab	--	--	PREab, PREpb
RFMpb	--	PREpb	PREab, PREpb
PDE, SRE	--		
PDX, SRX	--		

6.3.2 Row Commands

6.3.2.1 Row No Operation (RNOP) Command

The ROW NO OPERATION (RNOP) command is a half-cycle command received on the row command inputs R[9:0] and latched either with the rising or with the falling CK clock edge (or both edges) as shown in Figure 16. RNOP is used to instruct the HBM4 device 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.

Row commands other than RNOP are defined either as half-cycle or as one-and-a-half-cycle commands that begin and end on a rising CK clock edge. These commands must be padded with RNOP on the falling CK clock edge of the same cycle. As an alternative, some row commands may be paired with PREpb or PREab commands on the falling CK clock edge instead of RNOP, with the specific conditions for these commands being explicitly described for each row command.

Parity is evaluated with the RNOP command when the parity calculation is enabled in the Mode Register.

RNOP is assumed for the R[9:0] inputs on subsequent timing diagrams unless other row commands are explicitly shown.

6.3.2.1 Row No Operation (RNOP) Command (cont'd)

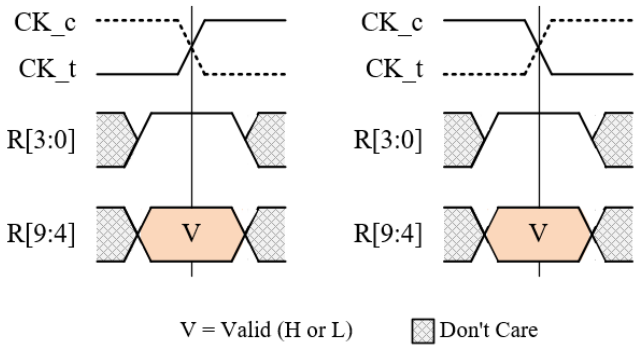


Figure 16 — RNOP command

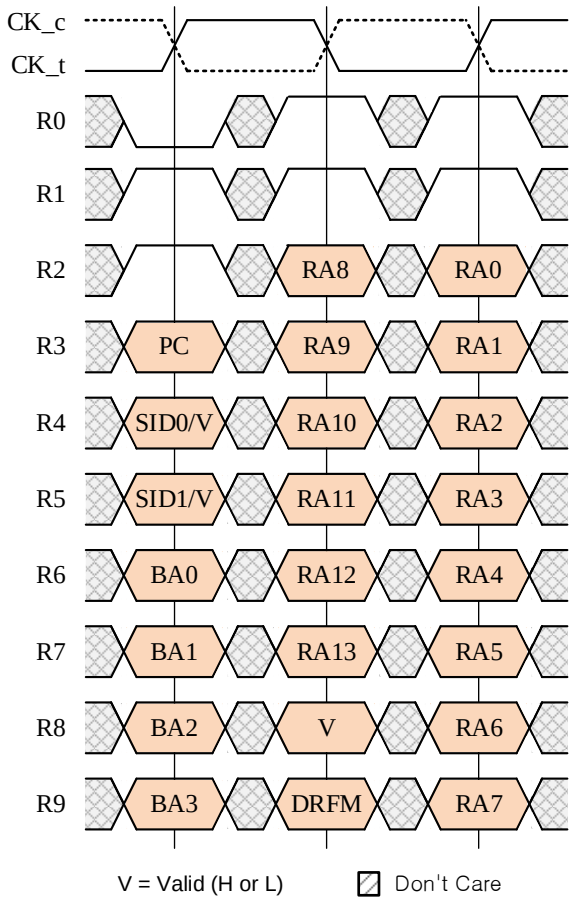
6.3.2.2 ACTIVATE (ACT) Command

Before a READ or WRITE command can be issued to a bank, a row in that bank must be opened. This is accomplished via the ACTIVATE command, which selects both the bank and the row to be activated. Once a row is open, a READ or WRITE command could be issued to that row, subject to the t_{RCD} specification.

The ACTIVATE command is a one-and-a-half-cycle command received on the row command inputs R[9:0] and latched with the rising and falling CK clock edges as shown in Figure 17. The command must be followed either by RNOP, PREpb or PREab on the falling CK clock edge of the second clock cycle. Note that a PREab in that case must be for the other pseudo channel. A PREpb command could be to any bank in the other pseudo channel as well as to a different bank in the same pseudo channel. In all cases the timing requirements for issuing these commands must be met.

The actual bank and row activation is initiated with the second rising CK clock edge of the ACTIVATE command; therefore all relevant timing parameters refer to this second rising CK clock edge as shown in Figure 18 and Figure 19.

6.3.2.2 ACTIVATE (ACT) Command (cont'd)



NOTE 1 BA = Bank Address; PC = Pseudo Channel 0 or 1; RA = Row Address; SID = Stack ID; V = Valid (H or L)

Figure 17 — ACTIVATE Command

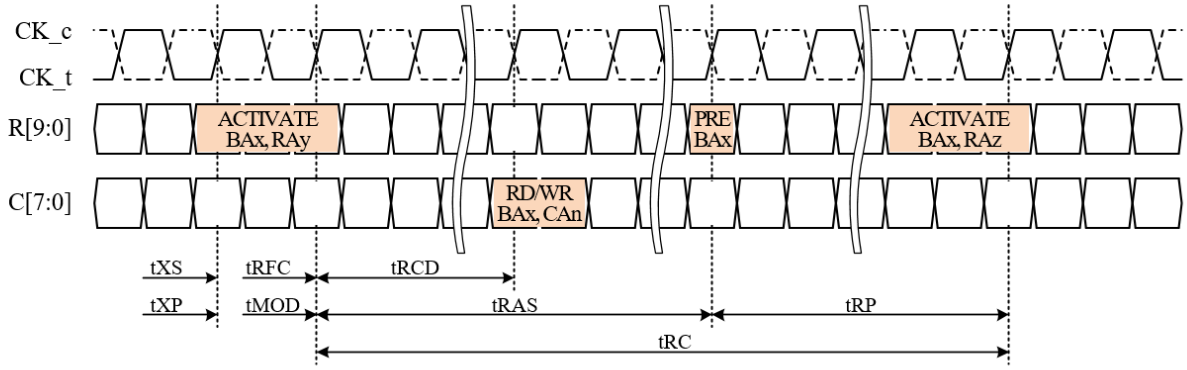
Parity is evaluated with the ACTIVATE command when the parity calculation is enabled in MR0 OP6.

A subsequent ACTIVATE command to another row in the same bank can only be issued after the previous row has been closed (precharged). A subsequent ACTIVATE can be issued to the same row address without closing the row to capture the address for DRFM. See the DRFM section for more details. The minimum time interval between successive ACTIVATE commands to the same bank is defined by t_{RC}, as shown in Figure 18. A minimum time t_{RAS} must have elapsed between opening and closing a row. The figure also shows two cases of t_{RAS} timings and command slots of the PRECHARGE command.

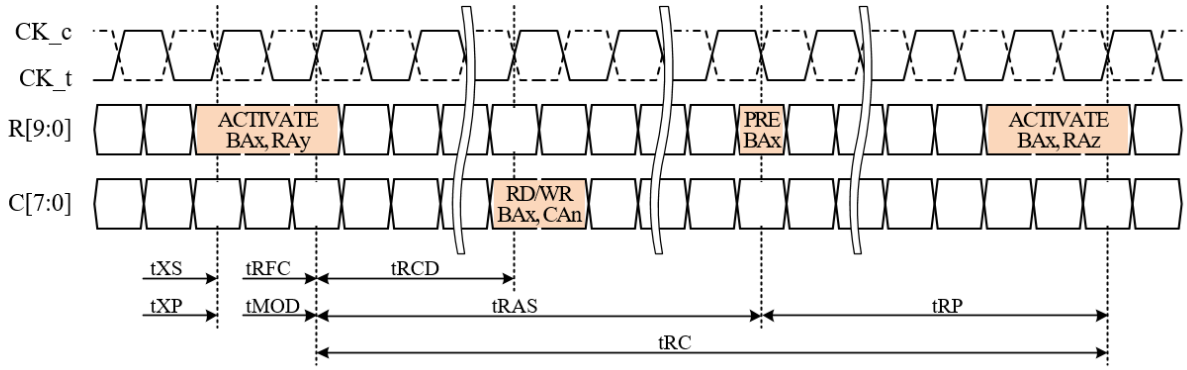
A subsequent ACTIVATE command to another bank can be issued while the first bank is being accessed, which results in a reduction of total row access overhead. The minimum time interval between successive ACTIVATE commands to different banks is defined by t_{RRD}. The row remains active until a PREpb command (or READ or WRITE command with Auto Precharge) is issued to the bank.

6.3.2.2 ACTIVATE (ACT) Command (cont'd)

Case1 : tRAS timing met at rising CK clock edge



Case2 : tRAS timing met at falling CK clock edge

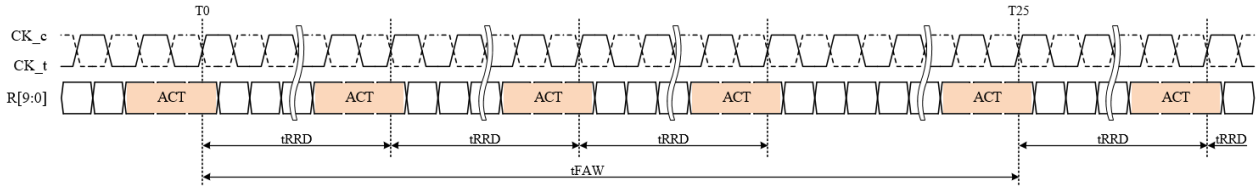


NOTE 1 BAx = bank address x; RAy,z = row addresses y,z; CAn = column address n.
NOTE 2 The PRECHARGE command shown could also be a PRECHARGE ALL command.
NOTE 3 t_{RCD} = t_{RCDRD} or t_{RCDWR}, depending on command; t_{RFC} = t_{RFCab} or t_{RFCpb}, depending on command.
NOTE 4 The reference for t_{XP} and t_{XS} timings is the first clock cycle of an ACTIVATE command, and the reference for t_{RFC} and t_{MOD} timing is the second clock cycle of an ACTIVATE command.

Figure 18 — Bank and Row Activation Command Cycle

6.3.2.2.1 Bank Restrictions

There is a need to limit the number of bank activations in a rolling window to ensure that the instantaneous current supplying capability of the device is not exceeded. To reflect the short term current supply capability, the parameter t_{FAW} (four activate window) is defined: no more than 4 banks may be activated in a rolling t_{FAW} window. Converting to clocks is done by dividing t_{FAW} (ns) by t_{CK} (ns) and rounding up to next integer value. As an example of the rolling window, if (t_{FAW}/t_{CK}) rounds up to 25 clocks, and an ACTIVATE command is issued at clock T0, no more than three further ACTIVATE commands may be issued at clocks T1 through T24 as illustrated in Figure 19.



NOTE 1 t_{RRD} = t_{RRDS} or t_{RRDL}, depending on accessed banks.
NOTE 2 Refer to the "REFRESH and PER-BANK REFRESH Command Scheduling Requirements" table for timing restrictions between all combinations of ACTIVATE and PER-BANK REFRESH commands.

Figure 19 — Multiple Bank Activations

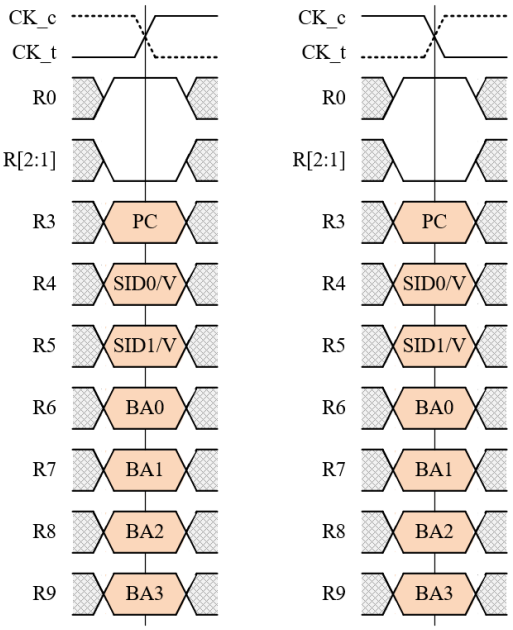
6.3.2.3 PRECHARGE (PREpb) and PRECHARGE ALL (PREab) Commands

The PRECHARGE (PREpb) and PRECHARGE ALL (PREab) commands are half-cycle commands received on the row command inputs R[9:0] and latched either with the rising or with the falling CK clock edge as shown in Figure 20 and Figure 21. The commands are used to deactivate the open row in a particular bank PREpb or the open rows in all banks PREab. The bank(s) will be in idle state and available for a subsequent row access a specified time t_{RP} after the PREpb command is issued.

The fact that both are half-cycle commands and defined on both the rising and the falling CK clock edges allows to issue one PREpb or PREab command on the rising CK clock edge and a second PREpb or PREab command on the falling CK clock edge of the same cycle and thus deactivate the open row in two different banks or even all banks in both pseudo channels within a single clock cycle, provided the t_{PPD} timing has been met. It is pointed out that the t_{RP} timing is always referenced from the CK clock edge on which the PREpb or PREab command is issued.

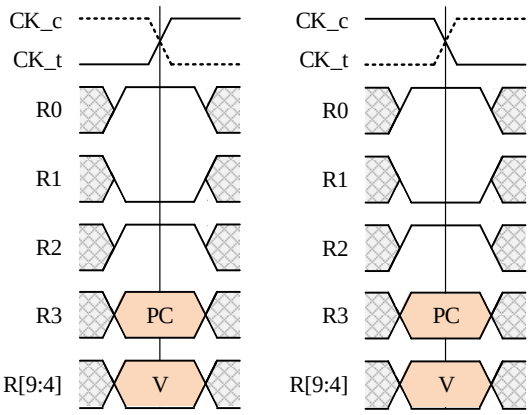
Parity is evaluated with the PREpb and PREab commands when the parity calculation is enabled in MR0 OP6.

6.3.2.3 PRECHARGE (PREpb) and PRECHARGE ALL (PREab) Commands (cont'd)



NOTE 1 BA = Bank Address; PC = Pseudo Channel 0 or 1; SID = Stack ID; V = Valid (H or L)

Figure 20 — PRECHARGE (PREpb) Command



NOTE 1 BA = Bank Address; PC = Pseudo Channel 0 or 1; SID = Stack ID; V = Valid (H or L)

Figure 21 — PRECHARGE ALL (PREab) Command

Input R2 determines whether one or all banks are to be precharged. In case where only one bank is to be precharged, bank addresses {SID[1:0], BA[3:0]} select the bank. Otherwise the bank addresses are treated as "Don't Care".

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 PREpb 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 PREpb command issued to the bank.

6.3.2.3.1 AUTO PRECHARGE

Auto Precharge is a feature which performs the same individual-bank precharge function described in Figure 20 and Figure 21, but without requiring an explicit PREpb command. Auto Precharge is nonpersistent meaning that it is enabled or disabled along for each individual READ or WRITE command.

For read bursts an auto precharge of the bank and row that is addressed with the READ command begins RTP clock cycles after the READ command was issued or after RAS has been met, with RTP as programmed in clock cycles in MR5 OP[3:0] and RAS as programmed in clock cycles in the RAS field of Mode Register MR4 OP[7:0].

For write bursts an auto precharge of the bank and row that is addressed with the WRITE command begins (WL + 2 + WR) clock cycles after the WRITE command was issued or after RAS has been met, with WR as programmed in clock cycles in MR3 OP[7:0] and RAS as programmed in clock cycles in MR4 OP[7:0].

Auto Precharge ensures that the precharge is initiated at the earliest valid stage within a burst. The user must not issue another command to the same bank until the precharge (t_{RP}) is completed. This is determined as if an explicit PREpb command was issued at the earliest possible time, as described for READ or WRITE commands. A precharge resulting from a READ or WRITE with Auto Precharge may occur in parallel with an explicit PREpb (or PREab) command. It is pointed out that an auto precharge is internally always issued with a rising CK clock edge, while explicit PREpb (or PREab) commands are supported on both clock edges.

6.3.2.3.1 AUTO PRECHARGE (cont'd)

Table 37 — Precharge and Auto Precharge Timings

From Command	To Command	Minimum Delay Between "From Command" to "To Command"	Unit	Note
READ	PRECHARGE (same bank)	t_{RTP}	nCK	
	PRECHARGE (different bank)	0	nCK	4
	PRECHARGE ALL	t_{RTP}	nCK	
	WRITE or WRITE w/ AP (any bank)	t_{RTW}	ns	
	READ or READ w/ AP (any bank)	t_{CCD}	nCK	5
READ w/ AP	PRECHARGE ALL	t_{RTP}	nCK	
	PRECHARGE (different bank)	0	nCK	4
	ACTIVATE or PER BANK REFRESH (same bank)	RTP + RU(t_{RP}/t_{CK})	nCK	2, 8
	WRITE or WRITE w/ AP (same bank)	Illegal		
	WRITE or WRITE w/ AP (different bank)	t_{RTW}	ns	
	READ or READ w/ AP (same bank)	Illegal		
	READ or READ w/ AP (different bank)	t_{CCD}	nCK	5
WRITE	PRECHARGE (same bank)	WL + 2 + RU(t_{WR}/t_{CK})	nCK	2, 6
	PRECHARGE (different bank)	0	nCK	4
	PRECHARGE ALL	WL + 2 + RU(t_{WR}/t_{CK})	nCK	2, 6
	WRITE or WRITE w/ AP (any bank)	t_{CCD}	nCK	5
	READ w/ AP (same bank)	WL + 2 + MAX[RU(t_{WR}/t_{CK}) – t_{RTP}, t_{WTR}]	nCK	2, 6, 7
	READ (same bank)	WL + 2 + t_{WTR}	nCK	6, 7
	READ or READ w/ AP (different bank)	WL + 2 + t_{WTR}	nCK	6, 7
WRITE w/ AP	PRECHARGE ALL	WL + 2 + RU(t_{WR}/t_{CK})	nCK	2, 6
	PRECHARGE (different bank)	0	nCK	4
	ACTIVATE or PER BANK REFRESH (same bank)	WL + 2 + WR + RU(t_{RP}/t_{CK})	nCK	2, 6, 8
	WRITE or WRITE w/ AP (same bank)	Illegal		
	WRITE or WRITE w/ AP (different bank)	t_{CCD}	nCK	5
	READ or READ w/ AP (same bank)	Illegal		
	READ or READ w/ AP (different bank)	WL + 2 + t_{WTR}	nCK	6, 7
PRECHARGE	PREpb (any bank)	t_{PPD}	nCK	3
	PREab	t_{PPD}	nCK	3
PRECHARGE ALL	PREpb or PREab	t_{PPD}	nCK	3

NOTE 1 A command issued during the minimum delay time is illegal.
NOTE 2 RU = round up to next integer.
NOTE 3 A PREpb 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.
NOTE 4 READ or WRITE and PREpb commands may be issued simultaneously.
NOTE 5 t_{CCD} could either be t_{CCDS} or t_{CCDL}; for READs, t_{CCD} could also be t_{CCDR}.
NOTE 6 WL = write latency.
NOTE 7 t_{WTR} could either be t_{WTRS} or t_{WTRL}.
NOTE 8 Even if t_{RP} is satisfied from PREab command, t_{RP} generated from previous WRA or RDA (Write or Read with Auto precharge) should also be satisfied.

6.3.2.4 Rounding Rules for Row Access Timings

The HBM4 DRAM allows the PREpb and PREab commands to be issued on both rising and falling CK clock edges, as e.g. illustrated in the Bank and Row Activation Command Cycle figure. To let a system take advantage of this flexibility in command scheduling, it is required to adapt the rounding rules for related row access timings

Traditionally, basic row access timings are converted into clock cycles using the formula NXX = RU(tXX/tCK), with XX representing either RAS, RTP, WR or RP parameters. This formula rounds the analog timings up to the next integer such that the subsequent command can be issued on the next rising clock edge that meets the analog value.

For HBM4 DRAM, this formula is replaced by NXX = 0.5 x RU(2 x tXX/tCK), which rounds analog timings to the next rising or following clock edge that meets the analog value. The result may be the same as with the traditional formula, or 0.5nCK less. The formula may be applied to row timings tRAS, tRTP, tWR and tRP, only. If rounding the tRP timing results in a falling edge as the command slot for a subsequent row access command, it is required to add 0.5nCK to the result because all such row commands following a row precharge can be issued on a rising clock edge only.

Examples:

- tRAS = 33 ns, tCK = 0.7 ns; NRAS = 0.5 x RU(2 x tRAS/tCK) = 0.5 x RU(2 x 33/0.7) = 0.5 x RU(94.29) = 47.5. Conclusion: When the ACTIVATE command was issued at T0, the earliest possible slot for a PRECHARGE command is at T47.5 (falling clock edge).

- tRP = 15 ns, tCK = 0.7 ns; NRP = 0.5 x RU(2 x tRP/tCK) = 0.5 x RU(2 x 15/0.7) = 0.5 x RU(42.85) = 21.5. Conclusion: When the PREpb command was issued at T0 (rising edge), the earliest possible slot for a subsequent ACTIVATE command is at T22, because the falling edge at T21.5 is not supported for an ACTIVATE command and 0.5nCK must be added to the result. However, when the PREpb command was issued at T0.5 (falling edge), the earliest possible slot for a subsequent ACTIVATE command is again at T2.

6.3.2.5 Refresh

The REFRESH command (REF) is used during normal operation of the HBM4 DRAMs. 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. The HBM4 DRAM requires REFRESH commands to be issued at an average periodic interval of tREFI.

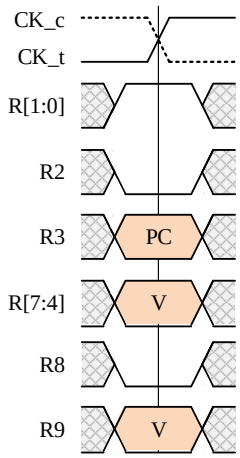
There are several types of refresh operations supported by HBM4 DRAMs.
- REFRESH all-bank (REFab)
- REFRESH per-bank (REFpb)
- REFRESH MANAGEMENT all-bank (RFMab)
- REFRESH MANAGEMENT per-bank (RFMpb)
- DIRECTED REFRESH MANAGEMENT (DRFMab)

This section describes the details of the refresh operations and requirements for each of the refresh operation types as well as the transitions between the refresh operation types.

6.3.2.5.1 REFRESH Command (REFab)

The REFRESH all-bank command (REFab) is a half-cycle command received on the row command inputs R[9:0] and latched with the rising CK clock edge as shown in Figure 22. The command must be followed either by RNOP, PREpb or PREab on the falling CK clock edge of the same cycle. Note that PREpb and PREab commands in this case must be for the other pseudo channel and the timing requirements for issuing these commands must be met. The REFab command also requires a CNOP command on the column command inputs C[7:0] unless the column command is for the other pseudo channel.

Parity is evaluated with the REFRESH command when the parity calculation is enabled in the Mode Register.



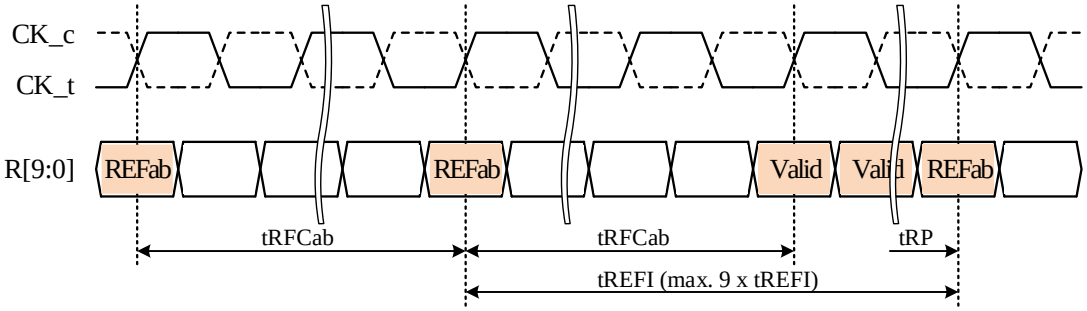
NOTE 1 PC = Pseudo Channel 0 or 1; V = Valid (H or L)

Figure 22 — REFRESH all-bank Command (REFab)

The REFab command is nonpersistent, so it must be issued each time a refresh is required. A minimum time t_{RFCab} is required between two REFab commands or a REFab command and any subsequent access command after the refresh operation. All banks must be precharged with t_{RP}-satisfied prior to the REFab command. The banks are in idle state after completion of the REFab command.

The refresh addressing is generated by an internal refresh controller. This makes the address bits “Don’t Care” during a REFab command.

6.3.2.5.1 REFRESH Command (REFab) (cont'd)



NOTE 1 Only RNOP and CNOP commands are allowed after a REFRESH command until t_{RFCab} has expired.
NOTE 2 The maximum time interval between two REFRESH commands is 9 x t_{REFI}.

Figure 23 — REFab Cycle

The HBM4 DRAM requires REFab cycles at an average periodic interval of t_{REFI}(MAX). To allow for improved efficiency in scheduling and switching between tasks, some flexibility in the absolute refresh interval is provided. A maximum of eight REFab commands can be postponed during operation of the device, meaning that at no point in time more than a total of eight REFab commands are allowed to be postponed. In case that eight REFab commands are postponed in a row, the resulting maximum interval between the surrounding REFab commands is limited to 9 × t_{REFI} (see Figure 24). At any given time, a maximum of 9 REFab commands can be issued within t_{REFI}.

This flexibility to postpone refresh commands also extends to REFpb commands (see REFpb). The maximum interval between refreshes to a particular bank is limited to 9 x t_{REFI}. At any given time, a maximum of 9 REFpb commands to a particular bank can be issued within t_{REFI}.

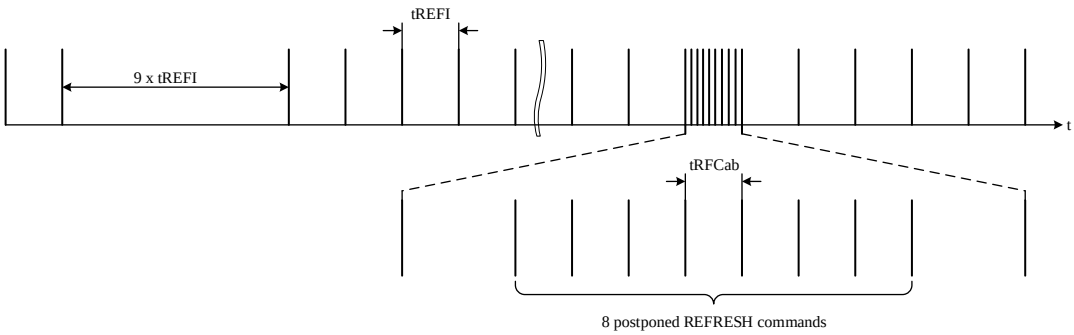


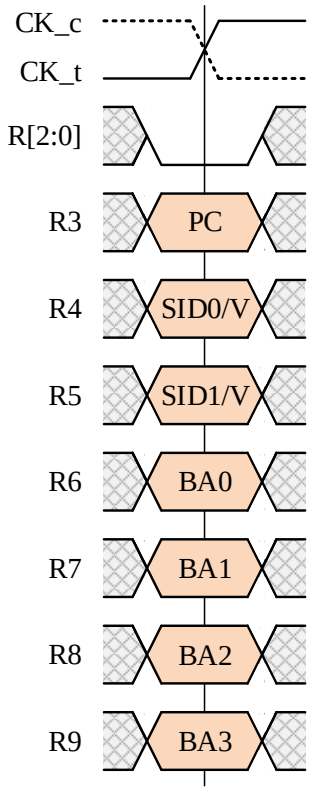
Figure 24 — Postponing Refresh Commands (Example)

Self refresh mode may be entered with a maximum of eight REFab commands being postponed. After exiting self refresh mode with one or more REFab commands postponed, additional REFab commands may be postponed to the extent that the total number of postponed REFab commands (before and after the self refresh) will never exceed eight. During self refresh mode, the number of postponed REFab commands does not change.

6.3.2.5.2 REFRESH per-bank Command (REFpb)

The REFRESH per-bank command (REFpb) provides an alternative solution for the refresh of the HBM4 device. The command initiates a refresh cycle on a single bank while accesses to other banks including writes and reads are not affected. REFpb is a half-cycle command received on the row command inputs R[9:0] and latched with the rising CK clock edge as shown in Figure 25. The command must be followed either by RNOP, PRECHARGE (PREpb) or PRECHARGE ALL (PREab) on the falling CK clock edge of the same cycle. Note that a PREab must be for the other pseudo channel. A PREpb command could be to any bank in the other pseudo channel as well as to a different bank in the same pseudo channel. In all cases the timing requirements for issuing these commands must be met.

Parity is evaluated with the REFpb command when the parity calculation is enabled in the Mode Register.

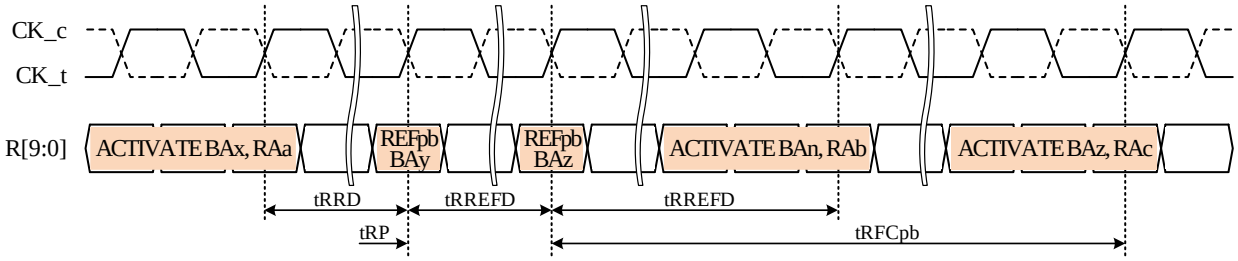


NOTE 1 BA = Bank Address; PC = Pseudo Channel 0 or 1; SID = Stack ID; V = Valid (H or L)

Figure 25 — REFRESH per-bank Command (REFpb)

The REFpb command is nonpersistent, so it must be issued each time a refresh is required. A minimum time t_{RRD} is required between an ACTIVATE command and a REFpb command to a different bank. A minimum time t_{RREFD} is required between any two REFpb commands (see below for an exception requiring t_{RFCpb}), and between a REFpb command and an ACTIVATE command to a different bank as shown in Figure 26. A minimum time t_{RFCpb} is required between a REFpb command and an access command to the same bank that follows. The bank to be refreshed must be precharged with t_{RP} satisfied prior to the REFpb command. The bank is in idle state after completion of the REFpb command.

6.3.2.5.2 REFRESH per-bank Command (REFpb) (cont'd)



NOTE 1 BAn,x,y,z = bank address n,x,y,z; RAa,b,c = row address a,b,c.
NOTE 2 t_{RRD} timing must be met between ACTIVATE commands and REFpb commands to different banks.
NOTE 3 t_{RREFD} timing must be met between consecutive REFpb commands to different banks, and between a REFpb command followed by an ACTIVATE command to the different bank.
NOTE 4 t_{RFCpb} timing must be met between a REFpb command followed by an ACTIVATE command to the same bank.

Figure 26 — REFpb Command Cycle

The row address for each bank is provided by internal refresh counters. This makes the row address bits "Don't Care" during REFpb commands.

Rules for issuing REFpb commands to the banks apply to each SID individually. A REFpb command to one of the 16 banks per SID can be issued in any order. After all banks within an SID have been refreshed using the REFpb command and after waiting for at least tRFCpb, the controller can issue another set of REFpb commands in the same or different order. However, it is illegal to send another REFpb command to a bank unless all banks within an SID have been refreshed using the REFpb command. The controller must track the bank being refreshed by the REFpb command.

A REFpb command and/or REFRESH MANAGEMENT command must not be issued during t_{RFCpb} succeeding the last REFpb (REFpb #N) of a single cycle as shown in Figure 27.

The bank count is synchronized between the controller and the HBM4 DRAM by resetting the bank count to zero. Synchronization can occur 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 banks.

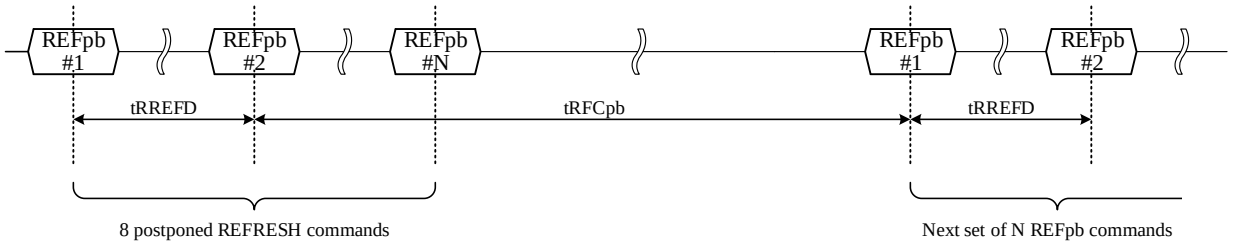


Figure 27 — Sets of REFpb Commands

6.3.2.5.2 REFRESH per-bank Command (REFpb) (cont'd)

The average rate of REFpb commands tREFIpb depends on the bank count N and can be calculated by the following formula:

tREFIpb = tREFI / N

The example in Table 38 shows two full sets of REFpb commands with the bank counter reset to 0 and the refresh counter incremented after 16 REFpb commands each. The third set of 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.

Table 38 — Refresh Counter Increments (Example)

Count	Sub-Count	Command	Bank Addr	Refresh Bank	Bank Counter	Refresh Counter
0	0	RESET_n, REFab, or SELF REFRESH ENTRY command			To 0	
1	1	REFpb	000	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	
…				…		

6.3.2.5.2 REFRESH per-bank Command (REFpb) (cont'd)

Table 39 — REFab and REFpb Command Scheduling Requirements

From Command	To Command	Minimum Delay Between "From Command" to "To Command"	Note
REFab	REFab	t_{RFCab}	
	REFpb (any bank)	t_{RFCab}	
	REFab	t_{RFCab}	
	RFMpb (any bank)	t_{RFCab}	
	ACTIVATE	t_{RFCab}	
REFpb	REFab	t_{RFCpb}	
	REFpb (different bank)	t_{RREFD}	
	REFpb (any bank)	t_{RFCpb}	3
	RFMFab	t_{RFCpb}	
	RFMpb (different bank)	t_{RREFD}	
	RFMpb (any bank)	t_{RFCpb}	4
	ACTIVATE (same bank)	t_{RFCpb}	
	ACTIVATE (different bank)	t_{RREFD}	1
RFMab	REFab	t_{RFCab}	
	REFpb (any bank)	t_{RFCab}	
	RFMab	t_{RFCab}	
	RFMpb (any bank)	t_{RFCab}	
	ACTIVATE	t_{RFCab}	
RFMpb	REFab	t_{RFCpb}	
	REFpb (same bank)	t_{RFCpb}	
	REFpb (different bank)	t_{RREFD}	
	RFMab	t_{RFCpb}	
	RFMpb (same bank)	t_{RFCpb}	
	RFMpb (different bank)	t_{RREFD}	
	ACTIVATE (same bank)	t_{RFCpb}	
	ACTIVEATE (different bank)	t_{RREFD}	1
ACTIVATE	REFab	t_{RC}	2
	REFpb (same bank)	t_{RC}	2
	REFpb (different bank)	t_{RRD}	1
	RFMab	t_{RC}	2
	RFMpb (same bank)	t_{RC}	2
	RFMpb (different bank)	t_{RRD}	1

NOTE 1 t_{FAW} parameter must be observed as well.
NOTE 2 A bank must be in the idle state with tRP satisfied before it is refreshed.
NOTE 3 t_{RFCpb} parameter must be observed when the first REFpb command completes a set of 16 per-bank refresh operations within an SID and the second REFpb command initiates the next set of 16 per-bank refresh operations within an SID.
NOTE 4 t_{RFCpb} parameter must be observed when the REFpb command completes a set of 16 per-bank refresh operations within an SID and the following RFMpb command operations.

6.3.2.5.3 Refresh Management (RFM)

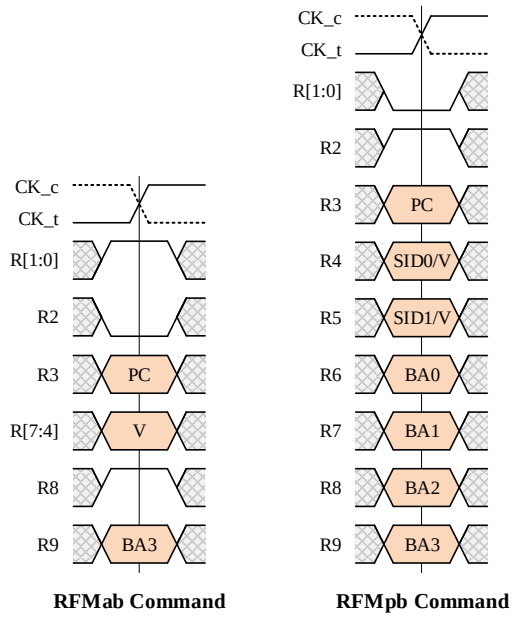
Periods of high DRAM activity may require additional refresh commands to protect the integrity of the stored data. The requirement for additional Refresh Management (RFM) is indicated in the RFM field of the DEVICE_ID WDR (see Table 126): RFM = 0 indicates that no additional refresh is needed beyond the refreshes specified in the REFRESH section of the standard; RFM = 1 indicates additional DRAM refresh management is required.

A suggested implementation of refresh management by the controller monitors 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 by the HBM4 DRAM in the RAAIMT field of the DEVICE_ID WDR (see Table 126), additional refresh management is needed. Executing a refresh management command allows additional time for the HBM4 DRAM 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.

The encoding of RFM related commands RFMab and RFMpb is shown in Figure 28. Both half-cycle commands are received on the R[9:0] inputs and latched with the rising CK clock edge. They must be followed either by RNOP, PREpb or PREab on the falling CK clock edge of the same cycle. Note that a PREab must be for the other pseudo channel. In case of a RFMpb command a PREpb command could be to any bank in the other pseudo channel as well as to a different bank in the same pseudo channel. In all cases the timing requirements for issuing these commands must be met.

An HBM4 DRAM not requiring refresh management will ignore RFMab and RFMpb commands and execute an RNOP command instead.



NOTE 1 BA = Bank Address; PC = Pseudo Channel 0 or 1; SID = Stack ID; V = Valid (H or L)

Figure 28 —RFMab and RFMpb Commands

6.3.2.5.3 Refresh Management (RFM) (cont'd)

The RFMab and RFMpb command scheduling shall meet the same minimum separation requirements as those for the REFab and REFpb commands, respectively (see Table 39 — REFab and REFpb Command Scheduling Requirements). The RFMab command period is the same as the REFab command period (t_{RFCab}), and the RFMpb command period is the same as the REFpb command period (t_{RFCpb}). The requirement for REFpb commands to be issued to all banks in a rolling fashion does not apply to RFMpb commands.

A REFpb command and/or RFMpb command must not be issued during t_{RFCpb} succeeding the last PER-BANK REFRESH (REFpb #N) of a single cycle as shown in Figure 26.

When an RFM command is issued to the HBM4 DRAM, the RAA counter 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 {SID[1:0], BA[3:0]} to be decremented by the RAAIMT value.

RFM commands are allowed to accumulate or "postpone", but the RAA counter shall never exceed a vendor specified RAA Maximum Management Threshold (RAAMMT), which is indicated by the HBM4 DRAM in the RAAMMT field of the DEVICE_ID WDR (see Table 126). If the RAA counter reaches RAAMMT, no additional ACTIVATE commands are allowed to the bank until one or more REF or RFM commands have been issued to reduce the RAA counter below the maximum value.

An RFM command does not replace the requirement for the controller to issue periodic REF commands to the HBM4 DRAM, nor does an RFM command affect internal refresh counters. The RFM commands are bonus time for the HBM4 DRAM to manage refresh internally. However, issuing a REF command also allows decrementing the RAA counter by a value indicated the RAA_CNT_DEC field of the DEVICE_ID WDR (see Table 126). Hence, any periodic REF command issued to the HBM4 DRAM allows the RAA counter of the banks being refreshed to be decremented by that value. Issuing a 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 {SID[1:0], BA[3:0]} to be decremented by that value.

The per-bank RAA count values may be reset to 0 when the HBM4 DRAM is held in self refresh for at least t_{RAASRF} time. No decrement to the per-bank RAA count values is allowed for entering or exiting self refresh and when the HBM4 DRAM is held in self refresh for less than t_{RAASRF} time.

6.3.2.5.4 Adaptive Refresh Management (ARFM)

HBM4 DRAMs optionally support a refresh management mode called Adaptive Refresh Management (ARFM). The HBM4 DRAM indicates the support of ARFM via the ARFM bit in the IEEE1500 DEVICE_ID WDR. 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 DRAM internal management of these commands. MR8 OP[5:4] select the RFM level as shown in Table 40.

Table 40 — Mode Register Definition for Adaptive RFM Levels

MR8 OP[5:4]	RFM Level	RFM Requirement	RAAIMT	RAAMMT	RAA Decrement per REF Command	Notes
00	Default	Default	RAAIMT	RAAMMT	RAADEC	1
01	Level A	RFM is required	RAAIMT_A	RAAMMT_A	RAADEC_A	
10	Level B	RFM is required	RAAIMT_B	RAAMMT_B	RAADEC_B	
11	Level C	RFM is required	RAAIMT_C	RAAMMT_C	RAADEC_C	

NOTE 1 RAAIMT, RAAMMT and RAADEC values are set by DRAM vendor in the IEEE1500 DEVICE_ID WDR.

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 level A to C can be retrieved from the corresponding fields of the IEEE1500 DEVICE_ID WDR.

Setting the bits in MR8 OP[5:4] to something other than the default "00" 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 HBM4 DRAM.

Adaptive RFM also allows an HBM4 DRAM shipped with 'RFM not required' (RFM bit in IEEE1500 DEVICE_ID WDR = 0) to override that initial setting and enable RFM by programming a non-default ARFM level. The HBM4 DRAM internally manages the change to treat the RFM command as an RFM command in this special override case as shown in Table 41.

6.3.2.5.4 Adaptive Refresh Management (ARFM) (cont'd)

Table 41 — RFM Commands Perceived by HBM4 DRAM

Command	Bit in DEVICE_ID WDR		RFM Level MR8 OP[5:4]	Command Perceived by HBM4 DRAM	Notes
	RFM	ARFM			
RFMab / RFMpb	0 (RFM not required)	0 (ARFM not supported)	00	RNOP	
			01, 10 or 11	Illegal	1
		1 (ARFM supported)	00	RNOP	
			01, 10 or 11	RFMab / RFMpb	2
	1 (RFM required)	0 (ARFM not supported)	00	RFMab / RFMpb	
			01, 10 or 11	Illegal	1
		1 (ARFM supported)	00, 01, 10 or 11	RFMab / RFMpb	

NOTE 1 These cases are marked as 'Illegal' because HBM4 DRAMs not supporting Adaptive RFM do not support the selection of an ARFM level via MR8 OP[5:4] and therefore define these bits as RFU which implies that the only supported setting for these bits is 00.

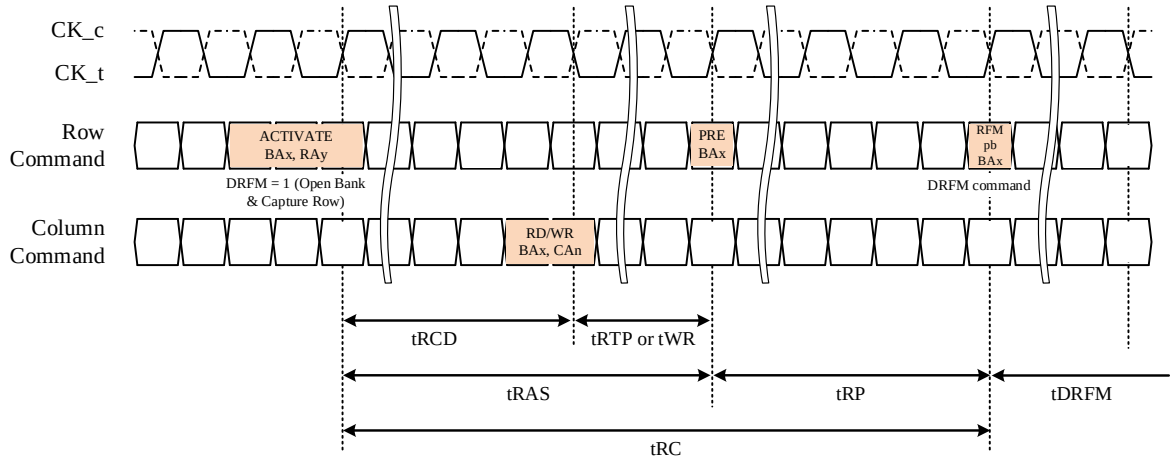
NOTE 2 Adaptive RFM enables an HBM4 DRAM shipped with RFM = 0 (RFM not required) to override the initial setting and enable Adaptive RFM by programming a non-default RFM level.

6.3.2.5.5 Directed Refresh Management (DRFM)

Directed Refresh Management (DRFM) is a feature that gives the controller additional flexibility for maintaining data integrity within the HBM4 DRAM. The DRFM feature allows the device to capture a host-requested row address, which then is followed by a host-directed RFMpb command allowing the device 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 MR0 OP3 to 1.

When DRFM is enabled, executing an ACT command with the DRFM bit set to 1_B will instruct the device to not only open the row but also capture the activated row address for DRFM as shown in Figure 29.

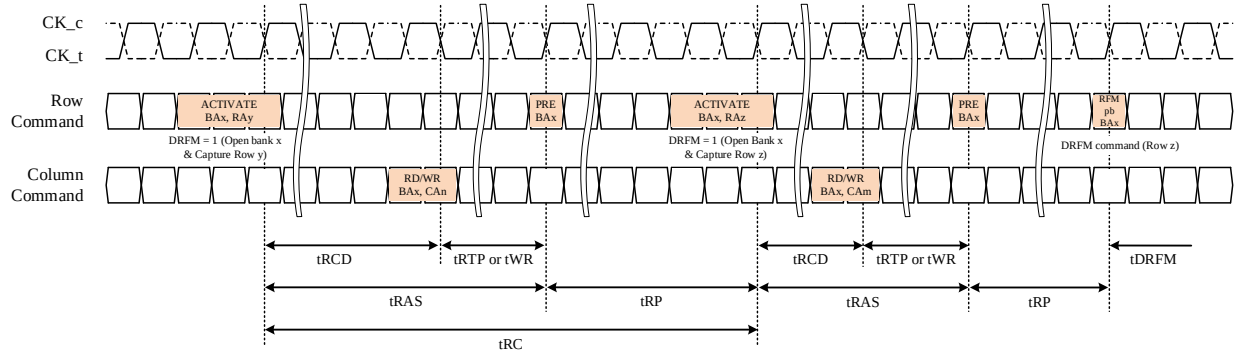
After the DRFM address capture, the host can issue an RFMpb command to the bank (referred to as DRFMpb command) to service the captured DRFM address. This DRFMpb command is supplemental to the device's RFM requirements and does not allow RAA count to be decremented. A RFMpb command issued to a bank without a valid address sample will be executed as a regular RFMpb command.



NOTE 1 PRE shown for illustration purposes.

Figure 29 — ACTIVATE with DRFM bit

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 as shown in Figure 30.



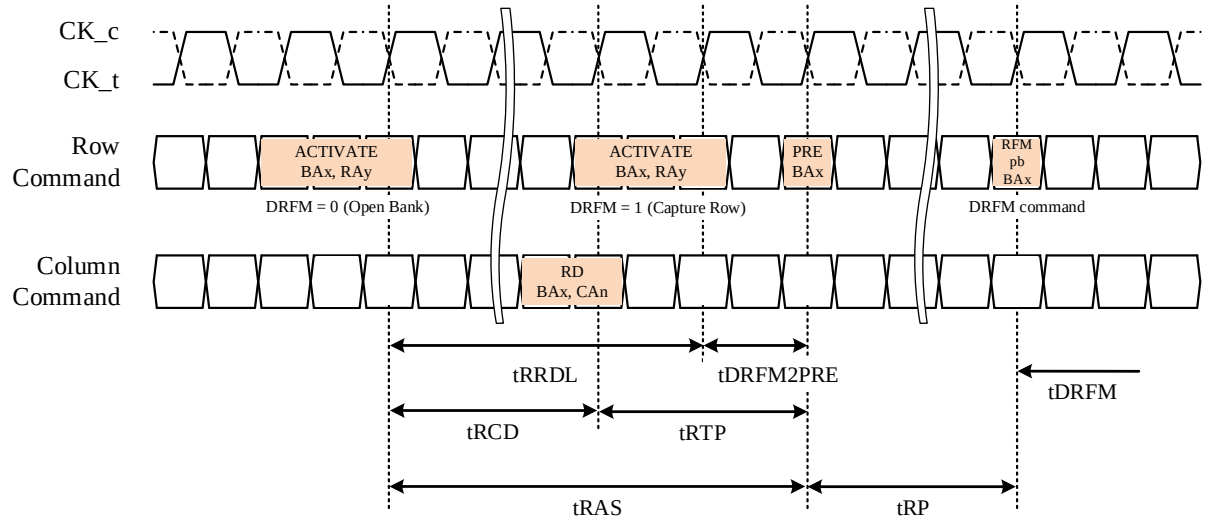
NOTE 1 PRE shown for illustration purposes.

Figure 30 — Multiple ACTIVATE with DRFM bit to same bank before DRFM command

Following a DRFMpb command, the DRFM address register for 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 capture addresses will be retained during Self Refresh mode, requiring the host to resample prior to issuing a DRFMpb command if the address retained in a bank's DRFM address register is no longer relevant. Additionally, no RAA credit is given to banks with DRFM sampled addresses, regardless of relevancy.

HBM4 also permits an ACTIVATE to an already open row with the DRFM flag set (DRFM=1) to capture the row. In this case the host must issue the second ACTIVATE to the same row address in the opened bank and must issue the additional ACTIVATE command tRRDL cycles after the initial ACTIVATE command that opened the row and must be issued no later than tDRFM2PRE cycles before a PREab/PREpb/WRA/RDA command to close the page as shown in Figure 31. The HBM4 DRAM will not open the bank as it is already open and simply c apture the row address for DRFM. A second ACTIVATE command with DRFM flag set to a different row address in the opened bank is illegal. If the initial ACTIVATE command opened the bank and captured the row address for DRFM then a second ACTIVATE with DRFM flag set to the same bank and row address while the bank is opened will capture the address again and all timings must be met.



NOTE 1 PRE shown for illustration purposes. tDRFM2PRE applies to the second ACTIVATE to the PREab, PREpb, WRA or RDA

Figure 31 — ACTIVATE with DRFM bit to open page (same bank and row address)

The DRFMpb command scheduling shall meet the same minimum separation requirements like tRP, tRRD or tRREFD as for a RFMpb command (see 6.3.2.5.3 Refresh Management (RFM) section). On average, any row/bank address combination is allowed to be sampled once per DRFM command interval, tDRFMI. tDRFMI is 2×tREFI, resulting in no more than TBD DRFM commands to the same row/bank address combination within tREF.

6.3.2.5.5.1 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[7:6]. The HBM4 DRAM is responsible for applying a refresh ratio to the outermost rows being refreshed to

protect the HBM4 DRAM 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 HBM4 DRAM. Likewise, if BRC4 is programmed, the HBM4 DRAM 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 Device ID.

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 42.

Table 42 — Bounded Refresh Configuration and tDRFM Timings

BRC	MR8 OP[7:6]	Rows Refreshed	tDRFM
2	00	Always ±1, Ratio ±2	4× tRRF
3 (Optional)	01	Always ±1, ±2, Ratio ±3	6 × tRRF
4 (Optional)	10	Always ±1, ±2, ±3, Ratio ±4	8 × tRRF
RFU	11	RFU	RFU

6.3.3 Column Commands

The column commands consist of CNOP, Read, Read with Auto Precharge, Write, Write with Auto Precharge, MRS. The column commands utilize C[7:0] inputs. All column commands are transmitted in a single clock cycle.

6.3.3.1 Column No Operation (CNOP)

The COLUMN NO OPERATION (CNOP) command is a 1-cycle command as shown in Figure 32 and is used to instruct the HBM4 DRAM to perform a NOP as the column command; this prevents unwanted column commands from being registered during idle or wait states. Operations already in progress are not affected.

Parity is evaluated with the CNOP command when the parity calculation is enabled in the Mode Register.

CNOP is assumed for the C[7:0] inputs on subsequent timing diagrams unless other column commands are explicitly shown.

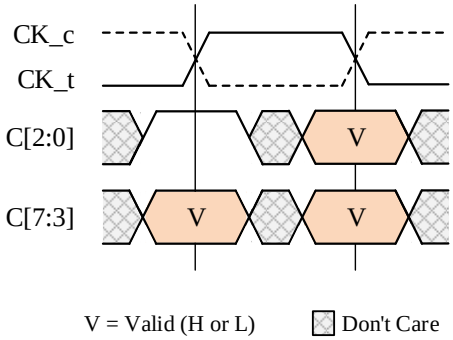
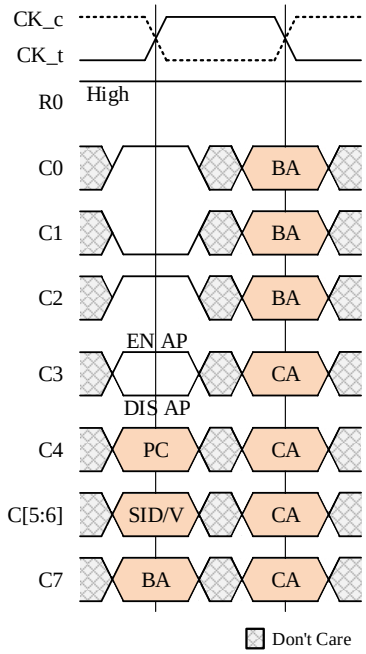


Figure 32 — CNOP Command

6.3.3.2 Read Command (RD, RDA)

A read burst is initiated with a READ command; READ is an one-cycle command received on the column command inputs C[7:0] and latched with the rising and falling CK clock edges as shown in Figure 33. The bank, PC, SID and column addresses are provided with the READ command and auto precharge is either enabled or disabled for that access.

Parity is evaluated with the READ command when CA parity is enabled in the Mode Register.



NOTE 1 BA = Bank Address; CA = Column Address; SID = Stack ID; PC = Pseudo Channel 0 or 1;
NOTE 2 EN AP = Enable Auto Precharge; DIS AP = Disable Auto Precharge

Figure 33 — READ Command

The length of the burst initiated with a READ command is eight. The column address is unique for the burst eight. There is no interruption nor truncation of read bursts.

The read latency (RL) is defined from the rising CK edge on which the READ command is issued to the rising CK edge from which the t_{DQSS} delay is measured, and the RL field of MR2 OP[7:0] (see Table 13). The first valid data is available RL × t_{CK} + t_{DQSS} + $t_{WDQS2DQ\_O}$ + t_{DQSQ} after the rising CK edge when the READ command was issued.

The write strobe(WDQS) is the source to trigger read data (DQ, DBI, ECC, SEV) and the read data strobe. The output drivers are enabled and begin driving either HIGH or LOW nominally two RDQS pulses (odd bytes) and one RDQS pulses (even bytes) prior to the first valid data bit. Bus pre-condition is Low regardless of RDBI enabled and disabled modes on a first READ command.

The output drivers will drive Hi-Z nominally one-half of RDQS pulse or less after the completion of the burst provided no other READ command has been issued.

6.3.3.2 Read Command (RD, RDA) (cont'd)

The write data strobe should be provided with a fixed four-pulse preamble and fixed two-pulse postamble before The read data strobe start to toggle because RDQS is generated from WDQS. The first WDQS edge occurs (RL-2) × t_{CK} + t_{DQSS}. The read data strobe provides a fixed two-pulse preamble and fixed two-pulse postamble; the first RDQS edge occurs (RL-1) × t_{CK} + t_{DQSS} + $t_{WDQS2DQ\_O}$ after the rising CK edge when the READ command was issued. The first data bit of the read burst is synchronized with the third rising edge of the RDQS strobe. Each subsequent data-out is edge-aligned with the data strobe. Timings for the data strobe are measured relative to the crosspoint of RDQS_t and its complement, RDQS_c.

6.3.3.2.1 Clock to Write Data Strobe Timings

The Write Data Strobe(WDQS) to Clock(CK) relationship is shown in Figure 34. Related parameters:

- t_{DQSS}(min/max) describes the allowed range for rising or falling WDQS edge relative to CK.
- t_{DQSS} is the actual position of a WDQS edge relative to CK.
- t_{WQSH} describes the WDQS HIGH pulse width
- t_{WQSL} describes the WDQS LOW pulse width

6.3.3.2.2 Write Data Strobe and Data Out Timings

The Write Data Strobe to Read Data Strobe(RDQS) relationship is shown in Figure 34. Related parameters:

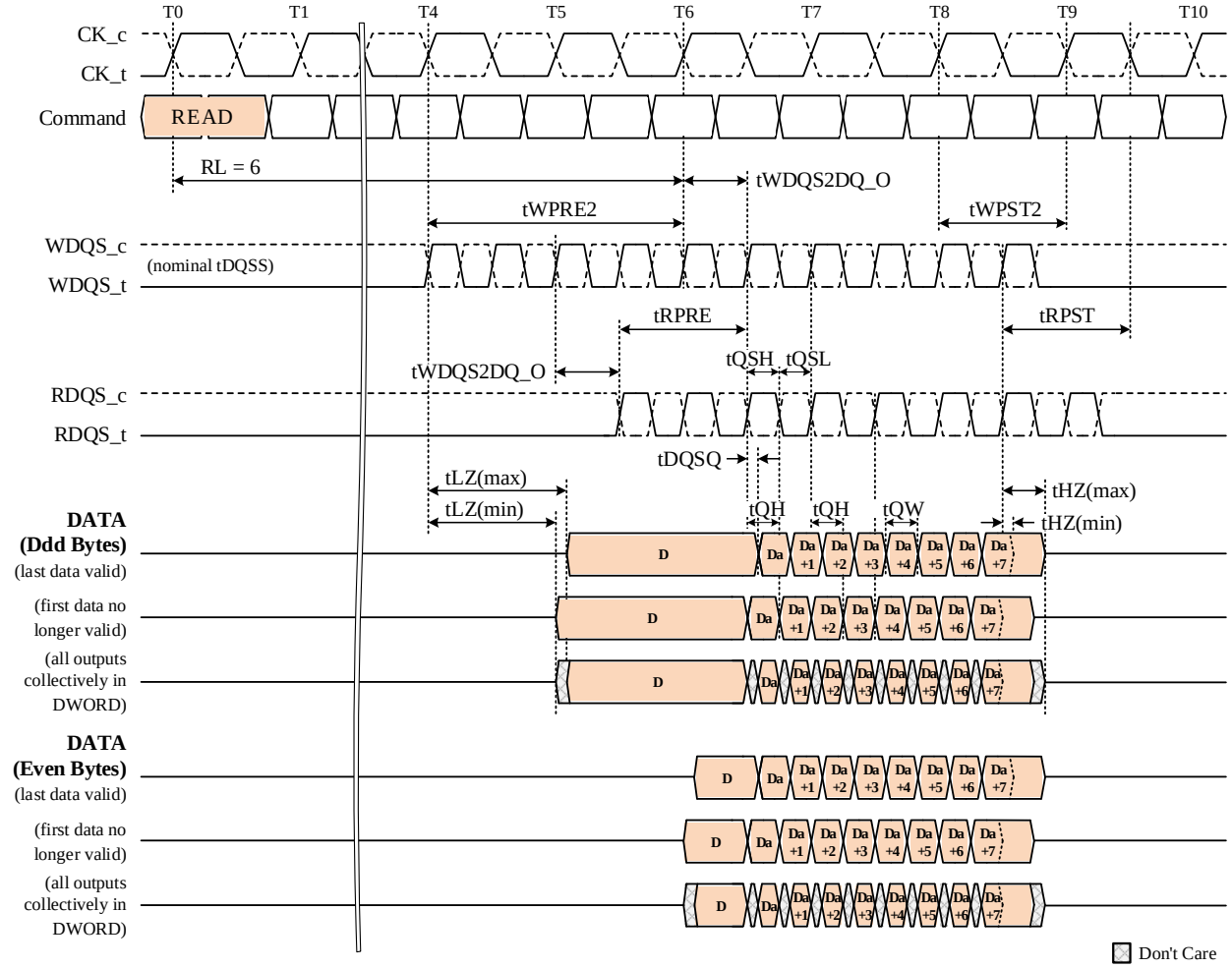
- $t_{WDQS2DQ\_O}$(min/max) describes the allowed range for a rising or falling RDQS edge relative to WDQS.
- $t_{WDQS2DQ\_O}$ is the actual position of a RDQS edge relative to WDQS.
- t_{QSH} describes the RDQS HIGH pulse width.
- t_{QSL} describes the RDQS LOW pulse width.
- t_{LZ}(min/max) describe the allowed range for the data output Hi-Z to low impedance transition relative to WDQS.
- t_{HZ}(min/max) describe the allowed range for the data output low impedance to Hi-Z transition relative to WDQS.

6.3.3.2.3 Read Data Strobe and Data Out Timings

The Read Data Strobe(RDQS) to Data Out(DQ, ECC, SEV, DBI) relationship is shown in Figure 34. Related parameters:

- tDQSQ describes the latest valid transition of any associated DQ or ECC or SEV or DBI pin for both rising and falling RDQS edges.
- tQH describes the earliest invalid transition of any associated DQ or ECC or SEV or DBI pin for both rising and falling RDQS edges.
- tQW describes the valid data output window of any associated DQ or ECC or SEV or DBI pin for both rising and falling RDQS edges.
- tDQ2DQ describes Read DQ to DQ skew of any associated DQ or ECC or SEV or DBI pin for both rising and falling RDQS edges.

6.3.3.2.3 Read Data Strobe and Data Out Timings (cont'd)

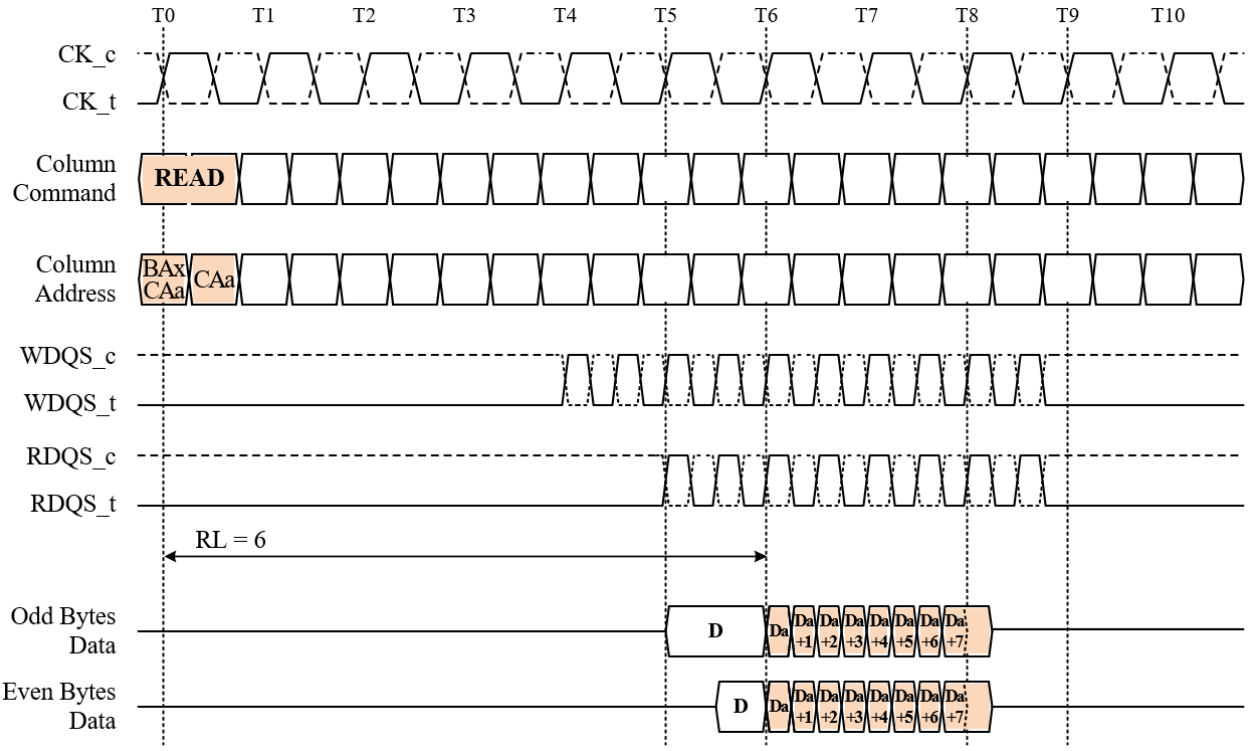


NOTE 1 $t_{WDQS2DQ\_O}$ may span multiple clock periods.
NOTE 2 A burst length of 8 is shown.
NOTE 3 Early/late data transition of a DQ or SEV or ECC or DBI can vary within a burst.
NOTE 4 Da...a+7 = data-out for READ command a.
D = last data-out from previous READ command (not if first READ after reset, MRS, self refresh or write-to-read).
NOTE 5 t_{WPRE2} = Read preamble for WDQS, t_{WPST2} = Read postamble for WDQS
NOTE 6 t_{RPRE} = Read preamble for RDQS, t_{RPST} = Read postamble for RDQS

Figure 34 — Clock to RDQS and Data Out Timings

6.3.3.2.4 Read Operation

Single read bursts are shown in Figure 35 for BL=8.

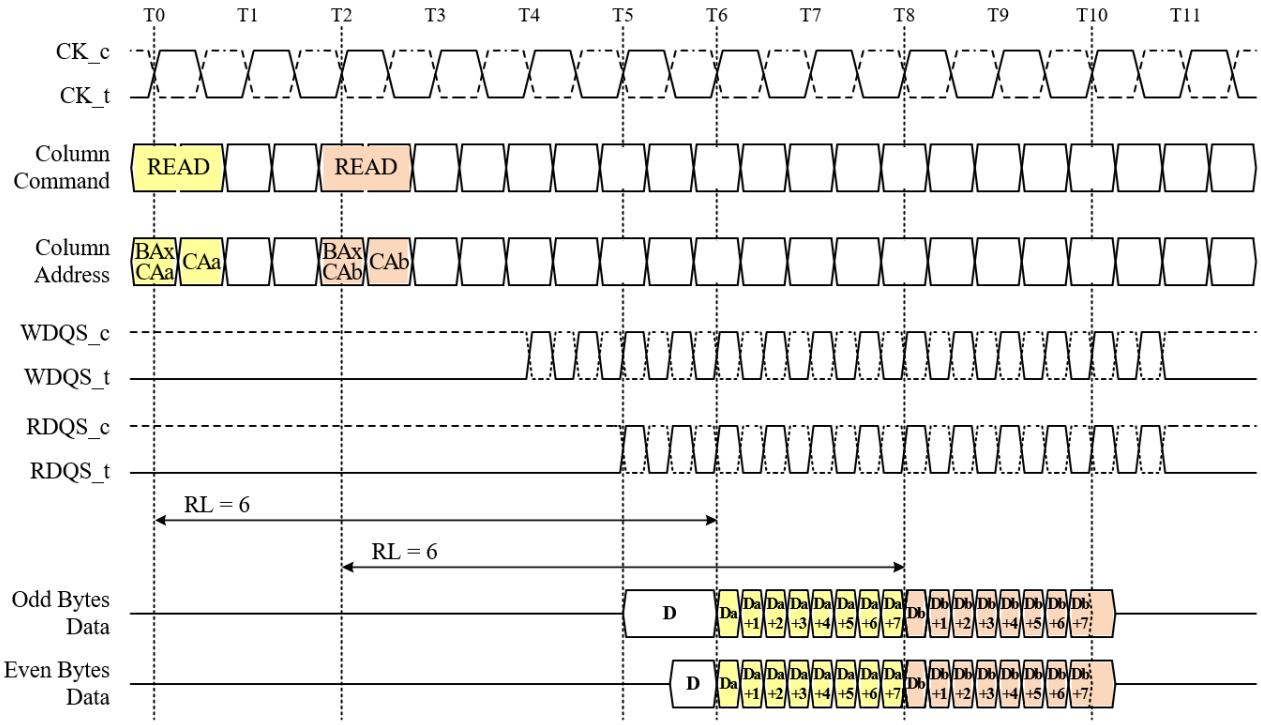


NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 RL = 6 is shown as an example.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0]. SEV[1:0] for P C0, and DQ[63:32], DBI[7:4],ECC[3:2]. SEV[3:2] for PC1.
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1.
NOTE 4 Da...a+7 = data-out for READ command a.
D = last data-out from previous READ command (not if first REA D after reset, MRS, self refresh or write-to-read).
NOTE 5 $t_{WDQS2DQ\_O}$ = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 6 RDBI could be on or off and is controlled with MR0 OP0.

Figure 35 — Single Read Burst with BL=8

6.3.3.2.4 Read Operation (cont'd)

Data from any read burst may be concatenated with data from a subsequent READ command. A continuous flow of data can be maintained as shown in Figure 36. 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 t_{CCD} timing. If that READ command is to another idle bank then an ACTIVATE command must precede the READ command and t_{RCDRD} also must be met.



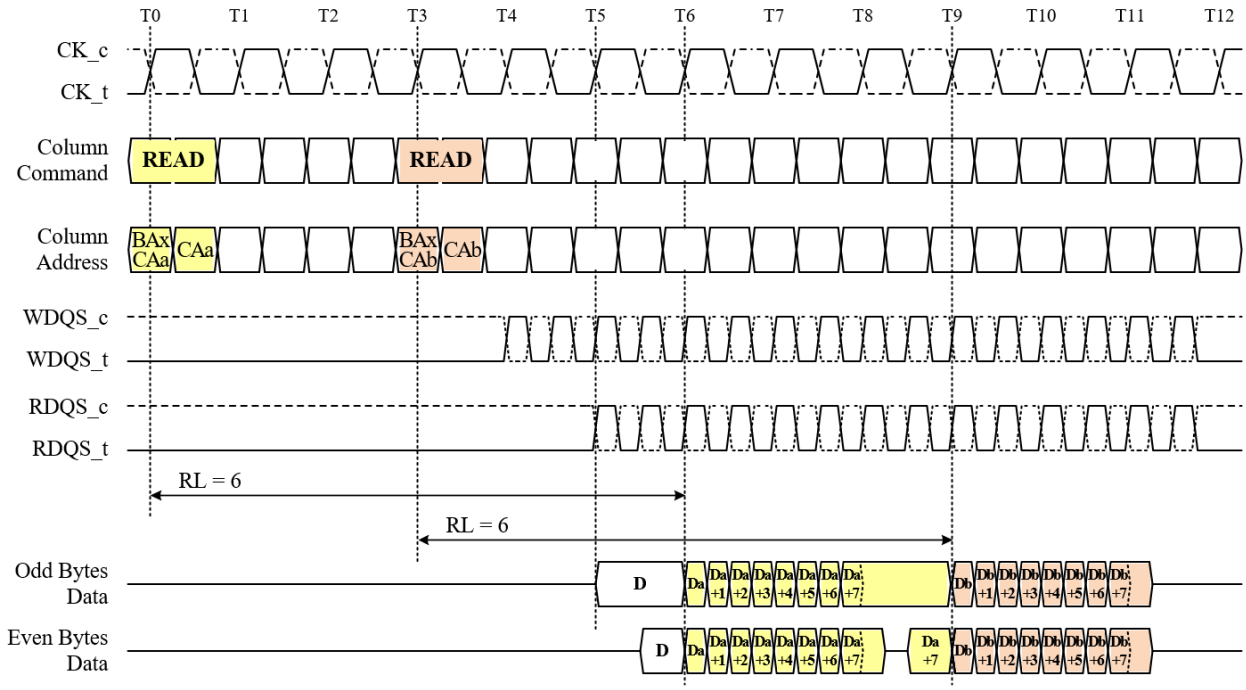
NOTE 1 BAx = bank address x; CAa,b = column address a,b.
NOTE 2 RL = 6 is shown as an example.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0]. SEV[1:0] for PC0, and DQ[63:32], DBI[7:4],ECC[3:2]. SEV[3:2] for PC1.
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1.
NOTE 4 Da,Da+1..Da+7,Db,Db+1..Db+7 = output data for READ commands a,b.
D = last data-out from previous READ command (not if first READ after reset, MRS, self refresh or write-to-read).
NOTE 5 $t_{WDQS2DQ\_O}$ = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 6 RDBI could be on or off and is controlled with MR0 OP0.

Figure 36 — Seamless Read Bursts with BL=8

6.3.3.2.4 Read Operation (cont'd)

Examples of non-seamless read bursts are shown in Figure 37 for t_{CCD}=3 and Figure 38 for t_{CCD}=4. The RDQS pulse at clock edge T8 in Figure 37 represents the read postamble of the first read burst as well as the read preamble of the second read burst. The chosen t_{CCD} value leads to a continuous series of RDQS pulses over both read bursts, and the data bus does not return to Hi-Z between the read bursts (for odd bytes), and the last data out of the first read burst (Da+7) is re-driven at the RDQS at clock edge T8+a half (for even bytes) preceding the second read burst.

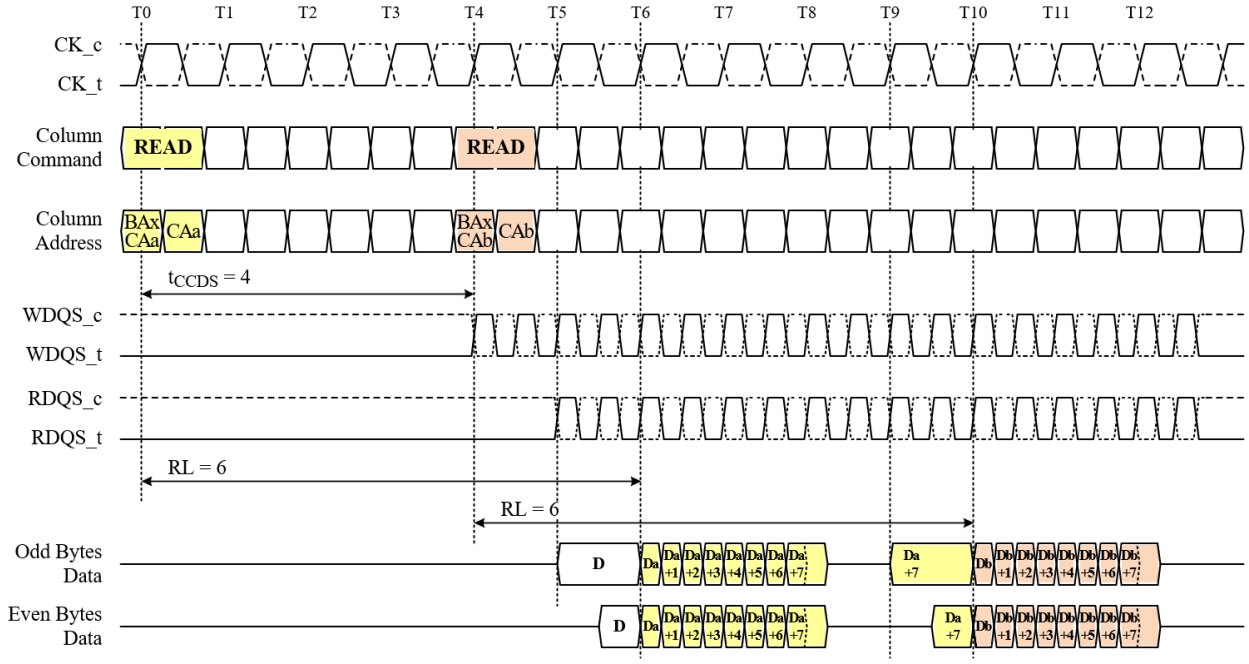
With t_{CCD}=4 as shown in Figure 38 the timing of each of the two read bursts is identical to a single read burst as shown in Figure 35. The data bus returns to Hi-Z between the read bursts, and the last data out of the first read burst (Da+7) is re-driven at the RDQS pulse at clock edge T9 (for odd bytes) and T9+a half (for even bytes) preceding the second read burst.



NOTE 1 BAx = bank address x; CAa,b = column address a,b.
NOTE 2 RL = 6, t_{CCD} = 3 are shown as an example.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0]. SEV[1:0] for P C0, and DQ[63:32], DBI[7:4],ECC[3:2]. SEV[3:2] for P C1.
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1.
NOTE 4 Da,Da+1..Da+7,Db,Db+1..Db+7 = output data for READ commands a,b.
D = last data-out from previous READ command (not if first READ after reset, MRS, self refresh or write-to-read).
NOTE 5 $t_{WDQS2DQ\_O}$ = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 6 RDBI could be on or off and is controlled with MR0 OP0.

Figure 37 — Non-Seamless Read Bursts with t_{CCD}=3 and BL=8

6.3.3.2.4 Read Operation (cont'd)

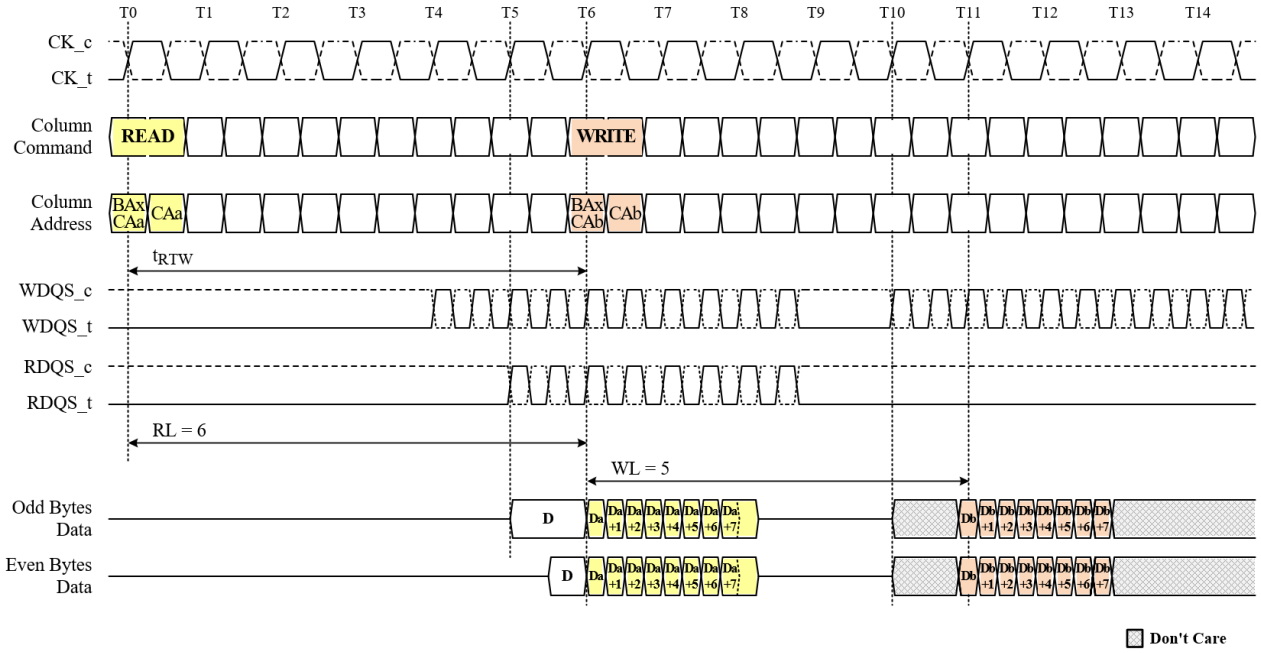


NOTE 1 BAx = bank address x; CAa,b = column address a,b.
NOTE 2 RL = 6, t_{CCD} =4 are shown as an example.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0]. SEV[1:0] for PC0, and DQ[63:32], DBI[7:4],ECC[3:2]. SEV[3:2] for PC1.
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1.
NOTE 4 Da,Da+1..Da+7,Db,Db+1..Db+7 = output data for READ commands a,b.
D = last data-out from previous READ command (not if first READ after reset, MRS, self refresh or write-to-read).
NOTE 5 $t_{WDQS2DQ\_O}$ = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 6 RDBI could be on or off and is controlled with MR0 OP0.

Figure 38 — Non-Seamless Read Burst with t_{CCD}=4 and BL=8

6.3.3.2.4 Read Operation (cont'd)

A WRITE can be issued any time after a READ command as long as the bus turnaround time t_{RTW} is met as shown in Figure 39. If that WRITE command is to another idle bank, then an ACTIVATE command must precede the WRITE command and t_{RCDWR} also must be met.

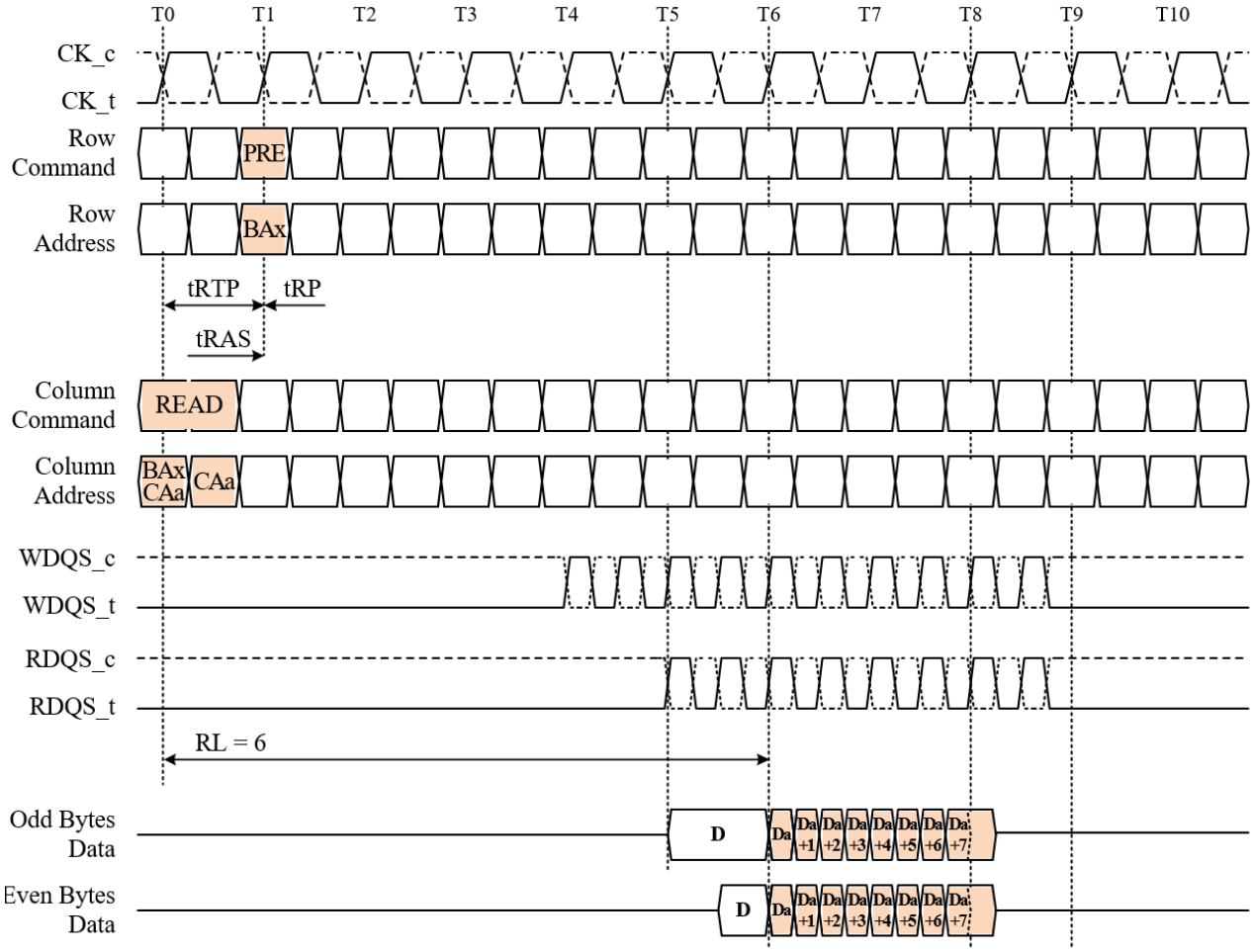


NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 RL=6 and WL=5 are shown as examples.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0]. SEV[1:0] for PC0, and DQ[63:32], DBI[7:4],ECC[3:2]. SEV[3:2] for PC1.
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1.
NOTE 4 Da...a+7 = data-out for READ command a.
D = last data-out from previous READ command (not if first READ after re set, MRS, self refresh or write-to-read).
NOTE 5 Db...b+7 = data-in for WRITE command b.
NOTE 6 $t_{WDQS2DQ\_O}$ = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 7 t_{RTW} is not a device limit but determined by the system bus turnaround time.
NOTE 8 RDBI and WDBI could be on or off. RDBI is controlled with MR0 OP0, and WDBI is controlled with MR0 OP1.

Figure 39 — Read to Write

6.3.3.2.4 Read Operation (cont'd)

A PRECHARGE can be issued t_{RTP} after the READ command as shown in Figure 40. After the PRECHARGE command, a subsequent ACTIVATE command to the same bank cannot be issued until t_{RP} is met.



NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 RL = 6 is shown as an example.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0]. SEV[1:0] for P C0, and DQ[63:32], DBI[7:4],ECC[3:2]. SEV[3:2] for P C1.
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1.
NOTE 4 Da...a+7 = data-out for READ command a.
D = last data-out from previous READ command (not if first READ after reset, MRS, self refresh or write-to-read).
NOTE 5 $t_{WDQS2DQ\_O}$ = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 6 t_{RTP} = 1 nCK is shown as an example. t_{RTP} = t_{RTPL} when bank groups are enabled and the PRECHARGE command accesses the same bank; otherwise t_{RTP} = t_{RTPS}.
NOTE 7 RDBI could be on or off and is controlled with MR0 OP0.

Figure 40 — Read to Precharge

6.3.3.2.5 Per-Signal-Group for Read De-Skew

The internal WDQS clock tree is optimized for lowest signal skew among signals within a group as outlined in Table 43 . The grouping is aligned with the physical location of signals in a DWORD (see HBM4 Ballout) with no lane being repaired.

Each group contains 6 to 8 signals. The internal WDQS clock tree however compensates the different loading by e.g. adding dummy loads. The per-group de-skew is also deterministic and not frequency dependent. A larger signal skew should be expected between different groups. A per-group de-skew is recommended to achieve the largest signaling margin for read data.

In this context, RDQS_t and RDQS_c are treated as regular out signals within group T4.

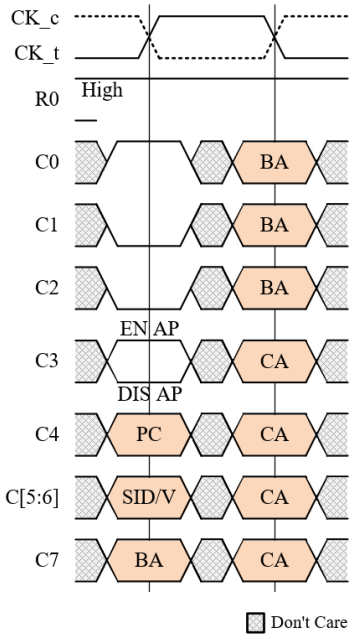
Table 43 — Signal Groups for Read Data De-Skew

Group	Signal List (DWORD0)	Signal List (DWORD1)
T0	DQ0, DQ1, DQ2, DQ8, DQ9, DQ10, ECC0, ECC1	DQ32, DQ33, DQ34, DQ40, DQ41, DQ42, ECC2, ECC3
T1	DQ3, DQ4, DQ11, DQ12, RD0, DPAR0	DQ35, DQ36, DQ43, DQ44, RD2, DPAR1
T2	DQ5, DQ6, DQ7, DQ13, DQ14, DQ15, DBI0, DBI1	DQ37, DQ38, DQ39, DQ45, DQ46, DQ47, DBI4, DBI5
T3	DQ16, DQ17, DQ18, DQ24, DQ25, DQ26, SEV0, SEV1	DQ48, DQ49, DQ50, DQ56, DQ57, DQ58, SEV2, SEV3
T4	DQ19, DQ20, DQ27, DQ28, RD1, RDQS0_t, RDQS0_c	DQ51, DQ52, DQ59, DQ60, RD3, RDQS1_t, RDQS1_c
T5	DQ21, DQ22, DQ23, DQ29, DQ30, DQ31, DBI2, DBI3	DQ53, DQ54, DQ55, DQ61, DQ62, DQ63, DBI6, DBI7

6.3.3.3 Write Command (WR, WRA)

A Write burst is initiated with a WRITE command. WRITE is a one-cycle command received on the column command inputs C[7:0] and latched with the rising and falling CK clock edges as shown in Figure 41. The bank, PC, SID and column addresses are provided with the WRITE command and auto precharge is either enabled or disabled for that access.

Parity is evaluated with the WRITE command when CA parity is enabled in MR0 OP5 (Table 11).



NOTE 1 BA = Bank Address; CA = Column Address; SID = Stack ID; PC = Pseudo Channel 0 or 1;
NOTE 2 EN AP = Enable Auto Precharge; DIS AP = Disable Auto Precharge

Figure 41 — Write Command

The length of the burst initiated with a WRITE command is eight. The column address is unique for this burst of eight. There is no interruption nor truncation of write bursts.

The write latency (WL) is defined from the rising CK edge on which the WRITE command is issued to the rising CK edge from which the t_{DQSS} delay is measured, and the WL field of MR1 OP[4:0]. The first valid data must be driven WL × t_{CK} + t_{DQSS} after the rising CK edge when the WRITE command was issued.

The write data strobe provides a fixed two-pulse preamble and two-pulse postamble; the first WDQS edge must be driven (WL-1) × t_{CK} + t_{DQSS} after the rising CK edge when the WRITE command was issued.

The HBM4 uses an un-matched WDQS-DQ path, so WDQS must stay within t_{DQSS} and the DQ can be trained to stay center aligned to the WDQS with satisfying t_{DIVW}. The DQ-data must be held for t_{DIVW} (data input valid window) and the WDQS can be periodically trained to stay center aligned to DQ in the t_{DIVW} window to compensate for timing changes due to temperature and voltage variation. Burst data is captured by the HBM on successive edges of WDQS until the burst length is complete. Pin timings for the data strobe are measured relative to the crosspoint of WDQS_t and its complement, WDQS_c.

6.3.3.3.1 Clock to Write Data Strobe Timings

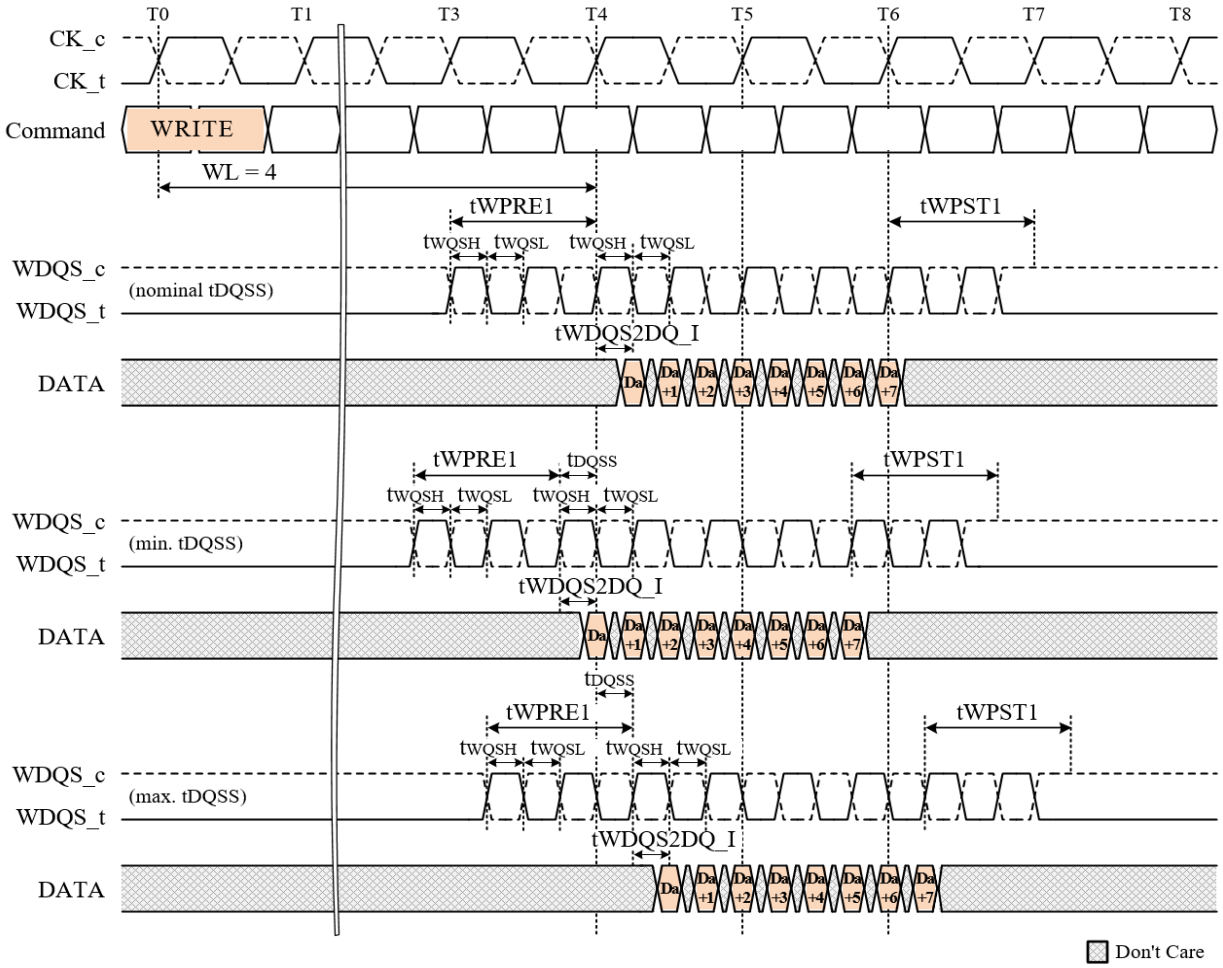
The clock to write data strobe (WDQS) relationship is shown in Figure 42. Related parameters:

- t_{DQSS}(min/max) describes the allowed range for a rising or falling WDQS edge relative to CK.
- t_{DQSS} is the actual position of a WDQS edge relative to CK.
- t_{WQSH} describes the WDQS HIGH pulse width.
- t_{WQSL} describes the WDQS LOW pulse width.

6.3.3.3.2 Write Data Strobe and Data In Timings

The write data strobe (WDQS) to data in relationship is shown in Figure 42. Related parameters:

- $t_{WDQS2DQ\_I}$ describes the allowed range for a DQ to a rising or falling WDQS edge.
- t_{DIVW} describes allowed range for receiver minimum setup/hold time for sampling at DQ.
- V_{DIVW} describes allowed range for receiver voltage peak to peak size.

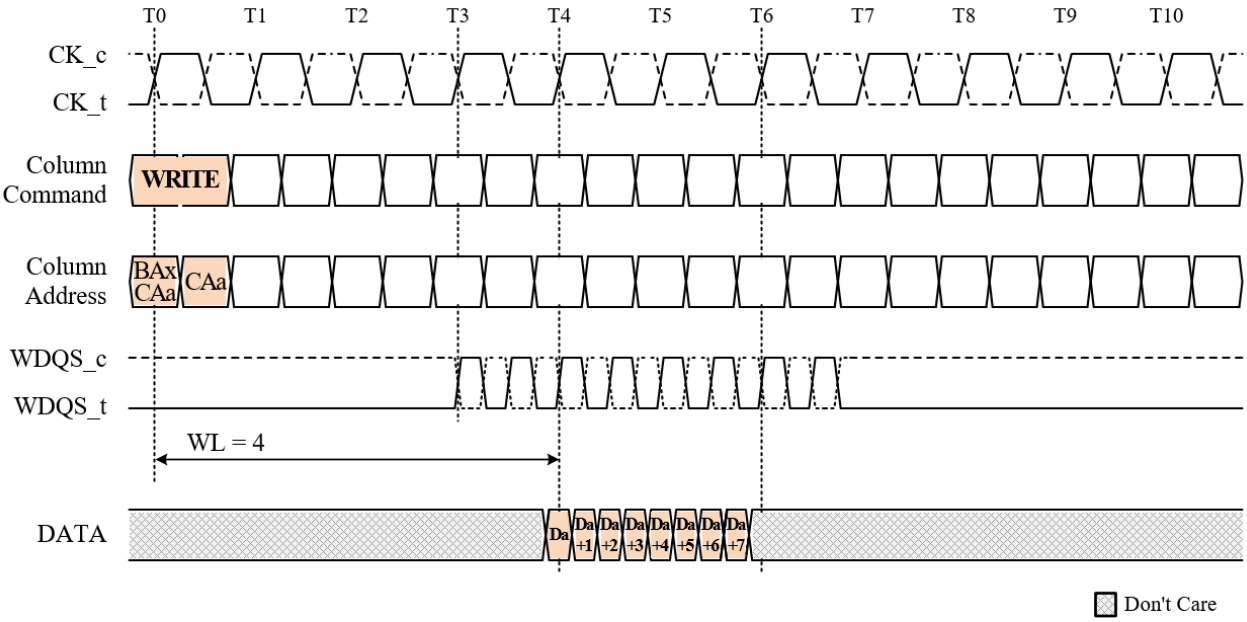


NOTE 1 DATA = DQ[63:0], DBI[3:0]
NOTE 2 Da, …, Da+7 = data-in for WRITE command a
NOTE 3 tWPRE1 = Write preamble for WDQS
NOTE 4 tWPST1 = Write postamble for WDQS

Figure 42 — Clock to WDQS and Data Input Timings

6.3.3.3.3 Write Operation

Single write bursts are shown in Figure 43.

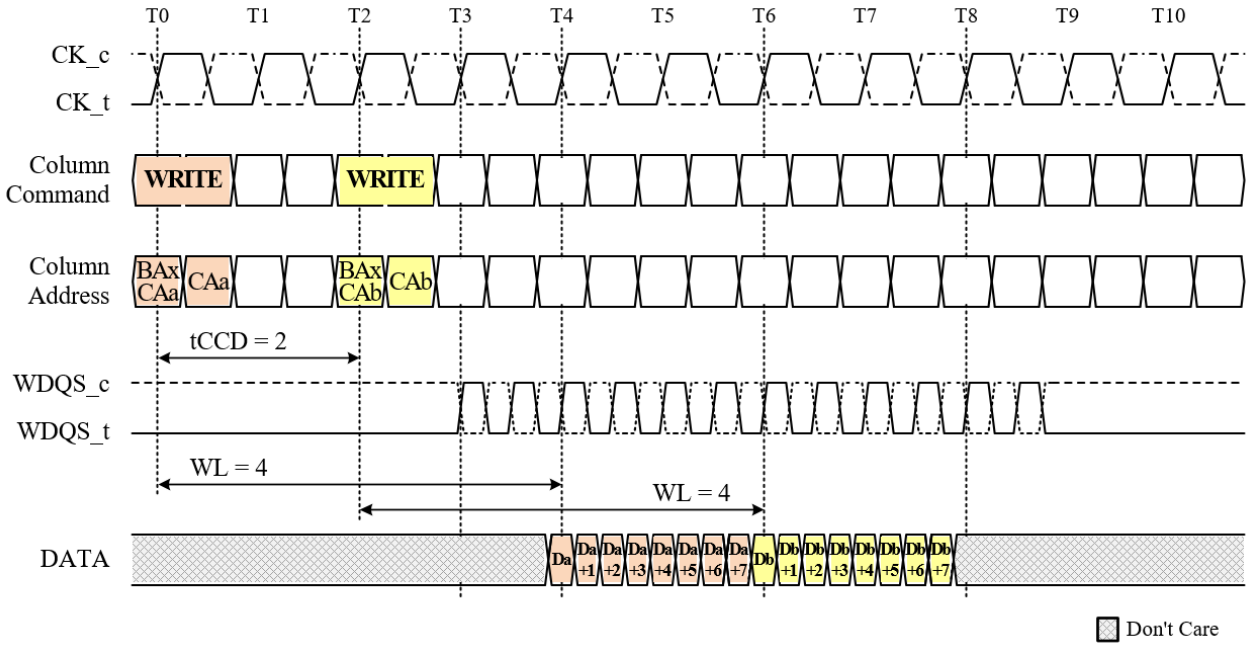


NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 WL = 4 is shown as an example.
NOTE 3 DATA = DQ[31:0]. DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4],ECC[3:2] for PC1.
DPAR = DPAR 0 for PC0 and DPAR1 for PC1 (if applicable).
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1..
NOTE 4 Da...Da+7 = data-in for WRITE command a.
NOTE 5 t_{DQSS} = 0 is shown for illustration purposes.
NOTE 6 WDBI could be on or off and is controlled with MR0 OP1.

Figure 43 — Single Write Burst with BL=8

6.3.3.3.3 Write Operation (cont'd)

Data from any write burst may be concatenated with data from a subsequent WRITE command. A continuous flow of data can be maintained as shown in Figure 44. 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 t_{CCD} timing. If that WRITE command is to another idle bank then an ACTIVE command must precede the WRITE command and t_{RCDWR} also must be met.

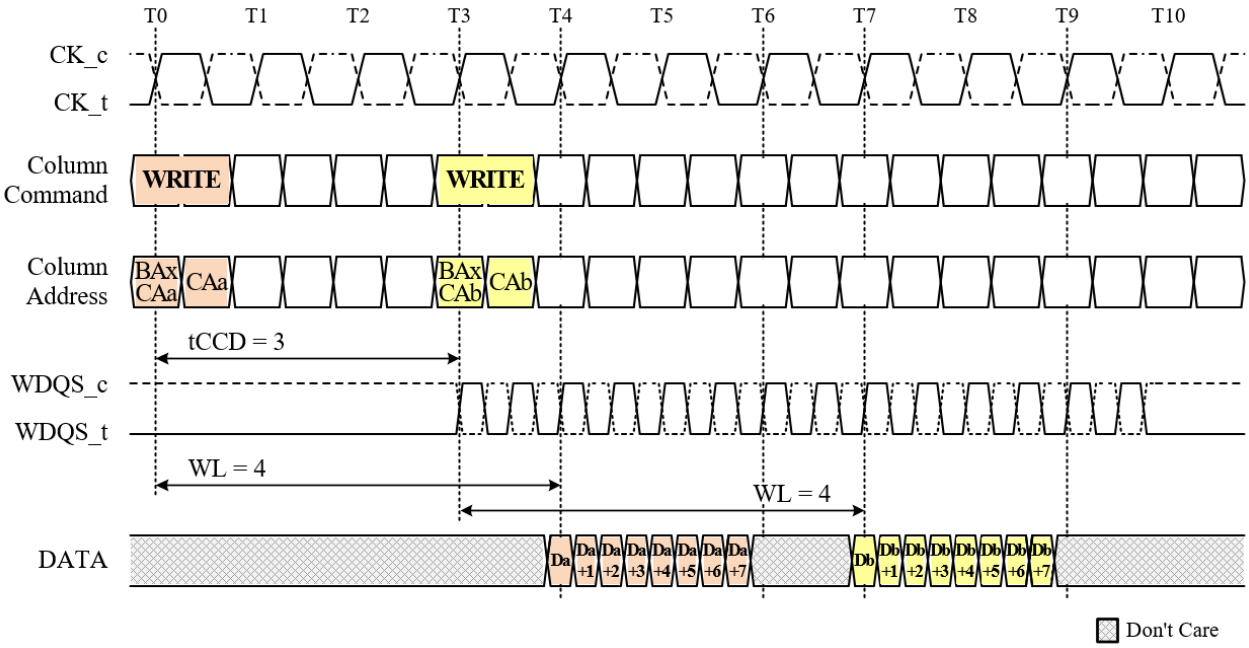


NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 WL = 4 is shown as an example.
NOTE 3 $t_{CCD} = t_{CCDS}$ when the second WRITE is to a different bank group, otherwise $t_{CCD}=t_{CCDL}$.
NOTE 4 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0 and DQ[63:32], DBI[7:4], ECC[3:2] for PC1.
DPAR = DPAR0 for PC0 and DPAR1 for PC1 (if applicable).
WDQS_t/_c is WDQS0_t/_c for PC0 and WDQS1_t/_c for PC1.
NOTE 5 Da...Da+7 = data-in for WRITE command a, Db...Db+7 = data-in for WRITE command b
NOTE 6 t_{DQSS} = 0 And t_{CCDS} = 2 are shown for illustration purposes.
NOTE 7 WDBI could be on or off and is controlled with MR0 OP1.

Figure 44 — Seamless Write Bursts with BL=8

6.3.3.3.3 Write Operation (cont'd)

Examples of non-seamless write bursts are shown in Figure 45.

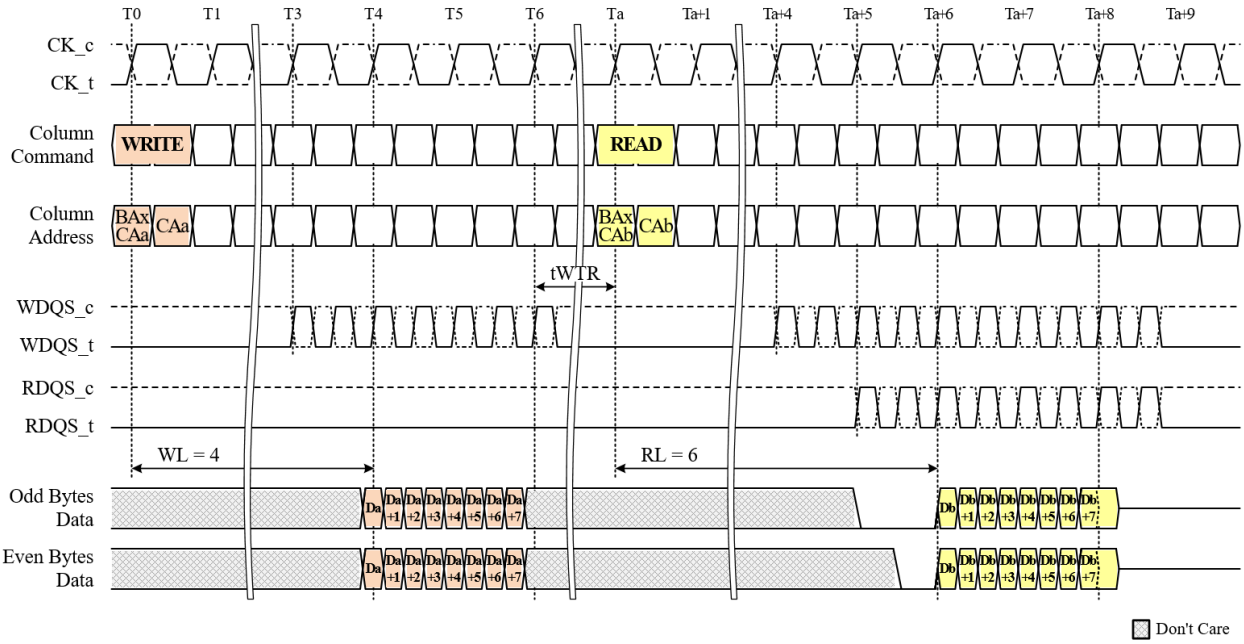


NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 WL = 4 is shown as an example.
NOTE 3 t_{CCD} = t_{CCDS} when the second WRITE is to a different bank group, otherwise t_{CCD}=t_{CCDL}.
NOTE 4 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0 and DQ[63:32], DBI[7:4], ECC[3:2] for PC1.
DPAR = DPAR0 for PC0 and DPAR1 for PC1 (if applicable).
WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
NOTE 5 Da...Da+7 = data-in for WRITE command a, Db...Db+7 = data-in for WRITE command b.
NOTE 6 t_{DQSS} = 0 And t_{CCDS} = 3 are shown for illustration purposes.
NOTE 7 WDBI could be on or off and is controlled with MR0 OP1.

Figure 45 — Non-Seamless Write Bursts

6.3.3.3.3 Write Operation (cont'd)

A READ can be issued any time after a WRITE command as long as the bus turnaround time t_{WTR} is met as shown in Figure 46. If that READ command is to another idle bank, then an ACTIVATE command must precede the READ command and t_{RCDRD} also must be met. The bus is preconditioned for the first read burst by being driven LOW two RDQS pulses (Odd bytes) and one RDQS pulse (Even bytes) prior to the first valid data element of the read burst regardless whether RDBI is enabled in MR0 OP0 or not.

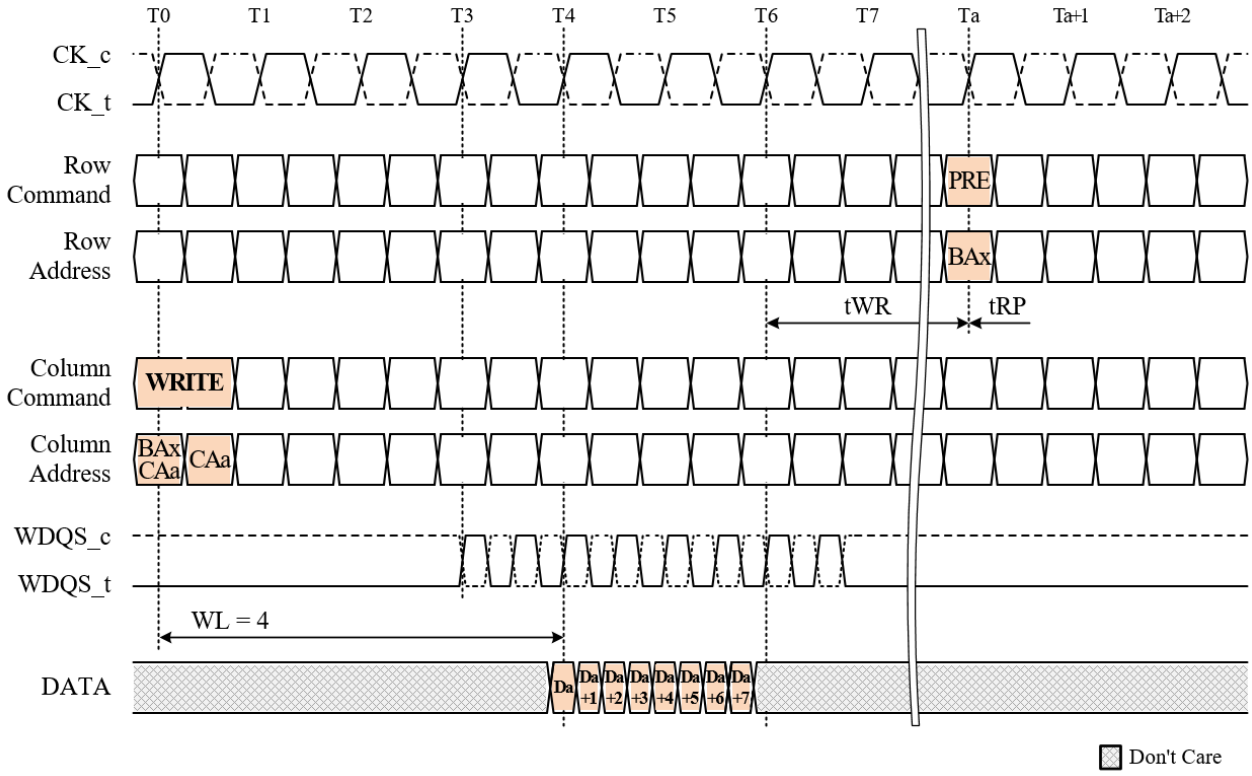


NOTE 1 BAx = bank address x; CAa,b = column address a,b.
NOTE 2 WL = 4 and RL = 6 are shown as examples.
NOTE 3 DATA = DQ[31:0], DBI[3:0], ECC[1:0], SEV[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2], SEV[3:2] for PC1.
DPAR = DPAR0 for PC0 and DPAR1 for PC1 (if applicable).
WDQS_t/_c is WDQS0_t/_c for PC0 and WDQS1_t/_c for PC1.
RDQS_t/_c is RDQS0_t/_c for PC0 and RDQS1_t/_c for PC1.
NOTE 4 Da...Da+7 = data-in for WRITE command b. Db...Db+7 = data-out for READ command a.
NOTE 5 $t_{WDQS2DQ\_O}$, t_{DQSS} = 0 and nominal t_{QW} is shown for illustration purposes.
NOTE 6 t_{WTR} = t_{WTRL} when both WRITE and READ access banks in the same bank group,
otherwise t_{WTR} = t_{WTRS}.
NOTE 7 WDBI could be on or off and is controlled with MR0 OP1.
NOTE 8 READ operation shown with RDBI enabled. RDBI is enabled/disabled with MR0 OP0.

Figure 46 — Write to Read

6.3.3.3.3 Write Operation (cont'd)

The write recovery time t_{WR} must have elapsed before a PRECHARGE command can be issued to that bank as shown in Figure 47; the t_{WR} interval begins with the completion of the write burst at WL + BL/4 clock cycles after the WRITE command was issued. Also, t_{RAS} must be met when the PRECHARGE is issued. After the PRECHARGE command, a subsequent ACTIVATE command to the same bank cannot be issued until t_{RP} is met.



NOTE 1 BAx = bank address x; CAa = column address a.
NOTE 2 WL = 4 is shown as an example.
NOTE 3 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2] for PC1.
DPAR = DPAR0 for PC0 and DPAR1 for PC1 (if applicable).
WDQS_t/_c is WDQS0_t/_c for PC0 and WDQS1_t/_c for PC1
NOTE 4 Da...Da+7 = data-in for WRITE command a.
NOTE 5 t_{DQSS} = 0 is shown for illustration purposes.
NOTE 6 WDBI could be on or off and is controlled with MR0 OP1.

Figure 47 — Write to Pre-charge

6.3.3.3.4 Per-Signal-Group for Write De-Skew

The internal WDQS clock tree is optimized for lowest signal skew among signals within a group as outlined in Table 44. The grouping is aligned with the physical location of signals in a DWORD (see HBM4 Ballout) with no lane being repaired.

Each group contains 5 to 8 signals. The internal WDQS clock tree however compensates the different loading by e.g. adding dummy loads. The per-group de-skew is also deterministic and not frequency dependent. A larger signal skew should be expected between different groups. A per-group de-skew is recommended to achieve the largest signaling margin for write data.

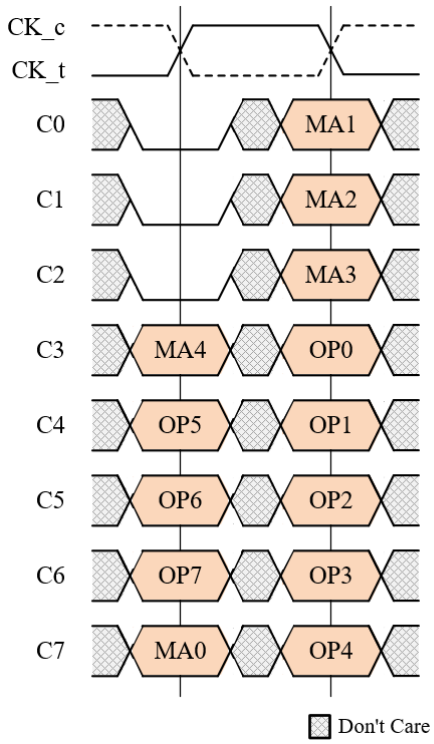
Table 44 — Signal Groups for Write Data De-Skew

Group	Signal List (DWORD0)	Signal List (DWORD1)
T0	DQ0, DQ1, DQ2, DQ8, DQ9, DQ10, ECC0, ECC1	DQ32, DQ33, DQ34, DQ40, DQ41, DQ42, ECC2, ECC3
T1	DQ3, DQ4, DQ11, DQ12, RD0, DPAR0	DQ35, DQ36, DQ43, DQ44, RD2, DPAR1
T2	DQ5, DQ6, DQ7, DQ13, DQ14, DQ15, DBI0, DBI1	DQ37, DQ38, DQ39, DQ45, DQ46, DQ47 DBI4, DBI5
T3	DQ16, DQ17, DQ18, DQ24, DQ25, DQ26	DQ48, DQ49, DQ50, DQ56, DQ57, DQ58
T4	DQ19, DQ20, DQ27, DQ28, RD1	DQ51, DQ52, DQ59, DQ60, RD3
T5	DQ21, DQ22, DQ23, DQ29, DQ30, DQ31, DBI2, DBI3	DQ53, DQ54, DQ55, DQ61, DQ62, DQ63, DBI6, DBI7

6.3.3.4 Mode Register Set (MRS) Command

The MODE REGISTER SET (MRS) command is a 1-cycle command as shown in Figure 48 and is used to load the Mode Registers of the HBM4 DRAM. The command is received on the column command inputs C[7:0] and requires an RNOP command on the row command inputs R[9:0].

Inputs MA[4:0] select one of the sixteen Mode Registers, and inputs OP[7:0] determine the opcode to be loaded. Refer to the Mode Registers section for the register definition.



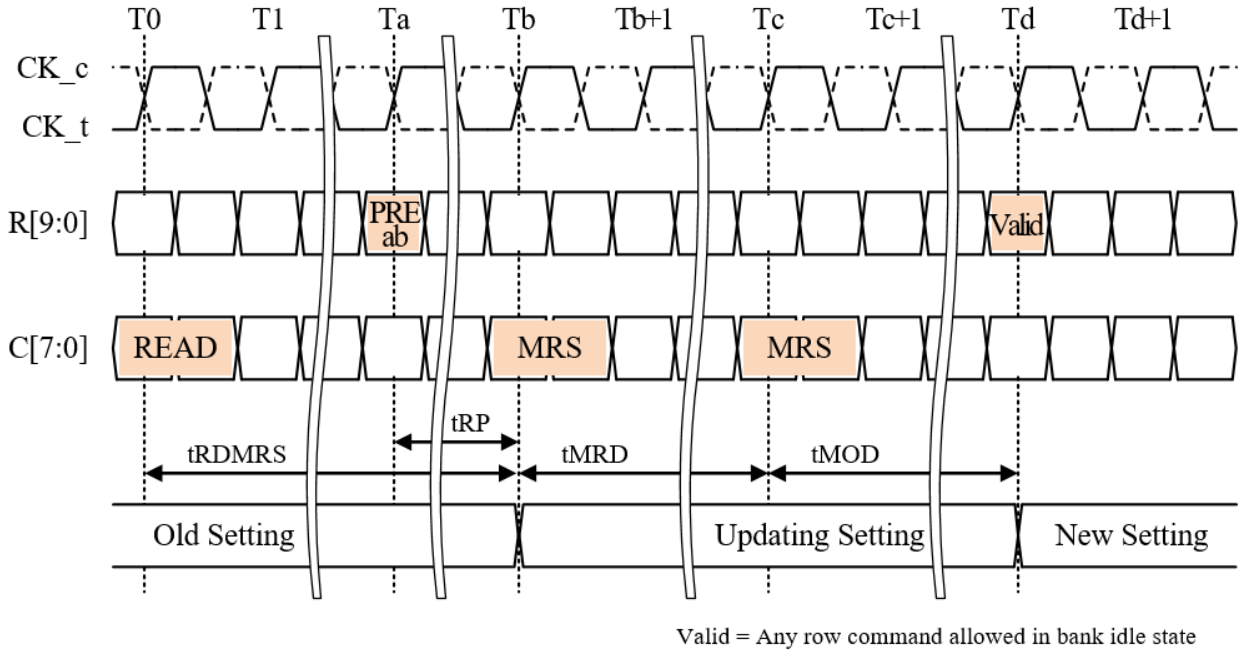
NOTE 1 MA = Mode Register Address; OP = Opcode; V = Valid (H or L)

Figure 48 — Mode Register Set Command (MRS)

The MODE REGISTER SET (MRS) command can only be issued when all banks are idle, the time t_{RDMRS} from a preceding READ command has elapsed and the time t_{WRMRS} from a preceding WRITE command has elapsed. The MRS command cycle time t_{MRD} is required to complete the write operation to the Mode Register and is the minimum time required between two MRS commands. The MRS command to Non-MRS command delay, t_{MOD}, is required by the HBM4 DRAM to update the features, and is the minimum time required from an MRS command to a non-MRS command excluding RNOP and CNOP.

Parity is evaluated with the MODE REGISTER SET command when CA parity has already been enabled in the Mode Register prior to this MODE REGISTER SET command. When CA parity is enabled by a MODE REGISTER SET command, the HBM4 DRAM requires all subsequent commands including RNOP and CNOP to be issued with correct parity until t_{MOD} has expired for the MODE REGISTER SET command that disables CA parity.

6.3.3.4 Mode Register Set (MRS) Command (cont'd)



NOTE 1 Valid shown as half cycle for illustration purposes.

Figure 49 — Mode Register Set Timings

6.3.4 Power-Mode Commands

6.3.4.1 Power-Down (PDE, PDX)

HBM4 devices enter Power-down with a Power-down Entry command as shown in Figure 50.

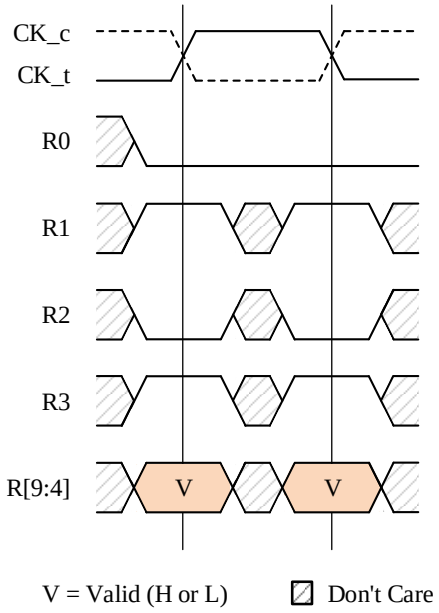


Figure 50 — Power-Down Entry Command

Power-down Entry must not be issued when read or write operations are in progress on either PC. A read operation is completed when the last data element including parity (when enabled) and RDQS postamble has been transmitted on the outputs. A write operation is completed when the last data element including parity (when enabled) has been written to the memory array with t_{WR} satisfied; for writes with auto-precharge, the number of clock cycles programmed in the mode register for WR must have elapsed instead.

Power-down Entry can be issued while any other operations such as row activation, precharge, auto precharge, or refresh are in progress, but the power-down IDD specification will not apply until such operations are complete.

If power-down occurs when all banks are idle, this mode is referred to as precharge power-down; if power-down occurs when there is a row active in any bank, this mode is referred to as active power-down.

To ensure that there is enough time to internally process the power-down entry, POWER DOWN ENTRY and CNOP commands have to be maintained for t_{CPDED} period. Also, the CK clock must be held stable for t_{CKPDE} cycle.

6.3.4.1 Power-Down (PDE, PDX) (cont'd)

Once t_{CPDED} and t_{CKPDE} have been met, the pins shall have the following states (see Table 45):

- The RESET_n and R0 receiver remains active; RESET_n = HIGH and R0 = LOW must be maintained to keep the HBM4 DRAM in power-down;
- The CK clock receiver remains active. The clock may be stopped with CK_t and CK_c being driven to static LOW and HIGH levels, respectively; in that case the clock must be stable again with t_{CH}(min) and t_{CL}(min) satisfied at least t_{CKPDX} cycles prior to power-down exit;
- WDQS_t = static LOW and WDQS_c = static HIGH levels must be maintained, respectively;
- RDQS_t and RDQS_c continue driving static LOW and HIGH levels, respectively;
- AERR, DERR continue driving static LOW levels;
- CATTRIP continues driving valid HIGH or LOW levels;
- All other input and output buffers are deactivated.

No refresh operations are performed in power-down mode. The maximum duration in power-down mode is limited by the refresh requirements of the device.

While in power-down the device will maintain the internal DBI state for the DBI(ac) calculation when DBI is enabled in the Mode Register. The device will also continue driving RDQS_t and RDQS_c to LOW and HIGH static levels, respectively, and CATTRIP to valid HIGH or LOW levels.

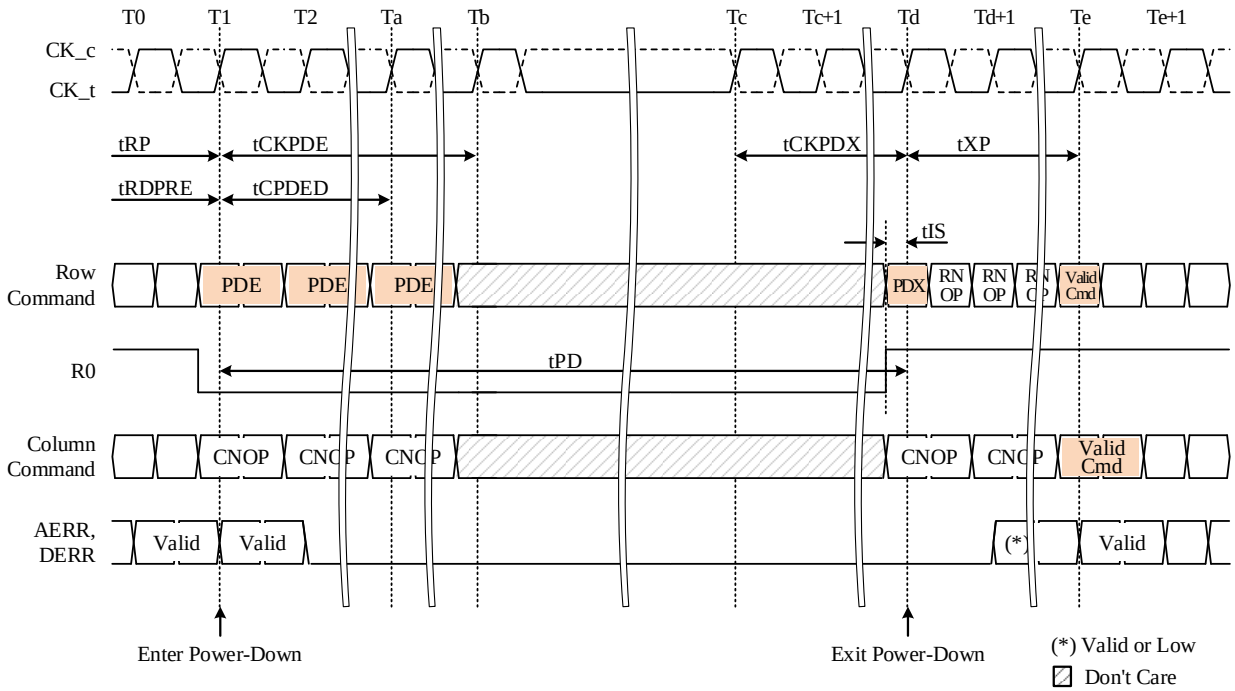
Power-down is synchronously exited when R0 is registered HIGH (in conjunction with CNOP commands). A valid executable command may be applied t_{XP} cycles later. The minimum power-down duration is specified by t_{PD}.

If CA parity is enabled, parity is evaluated for the POWER-DOWN ENTRY command. The HBM4 device requires PDE and CNOP commands with valid parity for the entire t_{CPDED} period, while it will suspend parity checking after power-down entry and drive AERR to a static LOW. DERR remains LOW as there are no data bursts in progress at this time.

Parity is not evaluated for the POWER-DOWN EXIT command. The HBM4 device requires RNOP and CNOP commands with valid parity for the entire t_{XP} period, while within t_{XP} period it will resume parity checking and indicating parity errors on AERR. DERR remains LOW as there are no data bursts in progress at this time.

Power-down is entered when R[3:0] are registered HIGH, LOW, HIGH, LOW along with CNOP commands as shown in Figure 51. PDE and CNOP commands are required for t_{CPDED} period after power-down entry.

6.3.4.1 Power-Down (PDE, PDX) (cont'd)



NOTE 1 Only PDE and CNOP commands are allowed during t_{CPDED} period. PDX, RNOP and CNOP commands are allowed during t_{XP} periods.

NOTE 2 Write bursts must have been completed with t_{WRPDE} (or t_{WRAPDE} in the case of write with auto pre-charge) satisfied prior to power-down entry.

NOTE 3 Read bursts must have been completed with t_{RDPDE} satisfied prior to power-down entry.

NOTE 4 Address inputs are "Don't Care" for power-down entry and exit.

NOTE 5 AERR, DERR are driven LOW when parity check is suspended during power-down. Signals are shown with t_{PARAC}=0 and t_{PARDQ}=0 for illustration purpose.

NOTE 6 The CK clock may be stopped during power-down as shown, or toggling.

NOTE 7 t_{CKPDE} means valid CK clocks required after first power-down entry.

NOTE 8 t_{CKPDX} means valid CK clocks required before power-down exit.

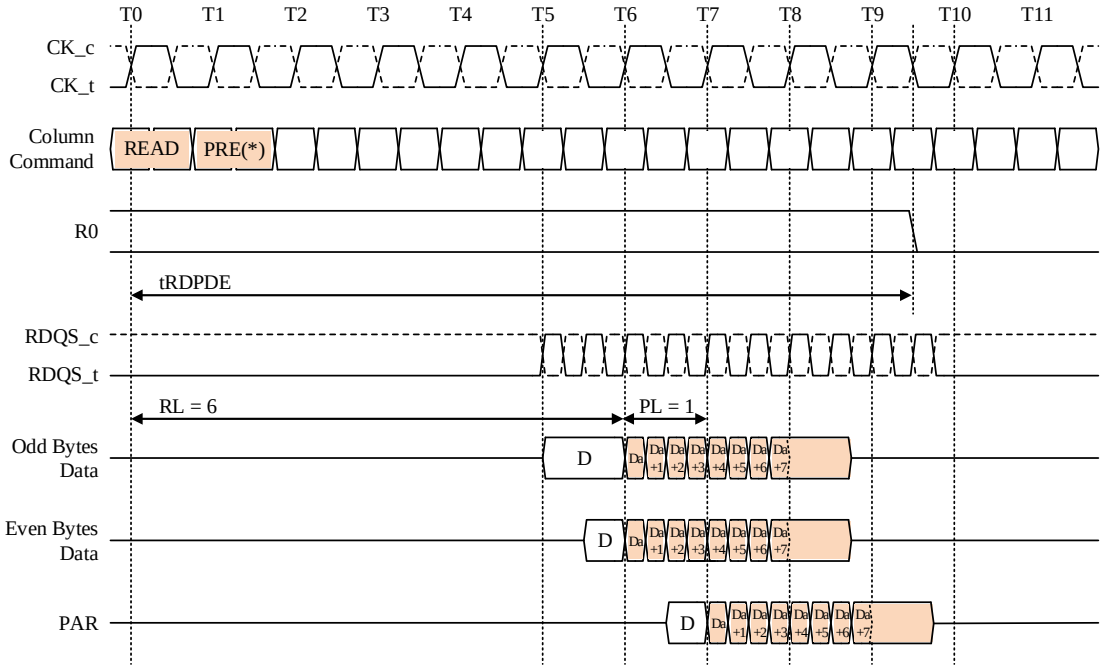
NOTE 9 Second PDE and third PDE after first PDE are treated as a RNOP and does not issue a power down entry.

Figure 51 — Power-Down Entry and Exit

6.3.4.1 Power-Down (PDE, PDX) (cont'd)

Table 45 — Pin State Description in Power Down

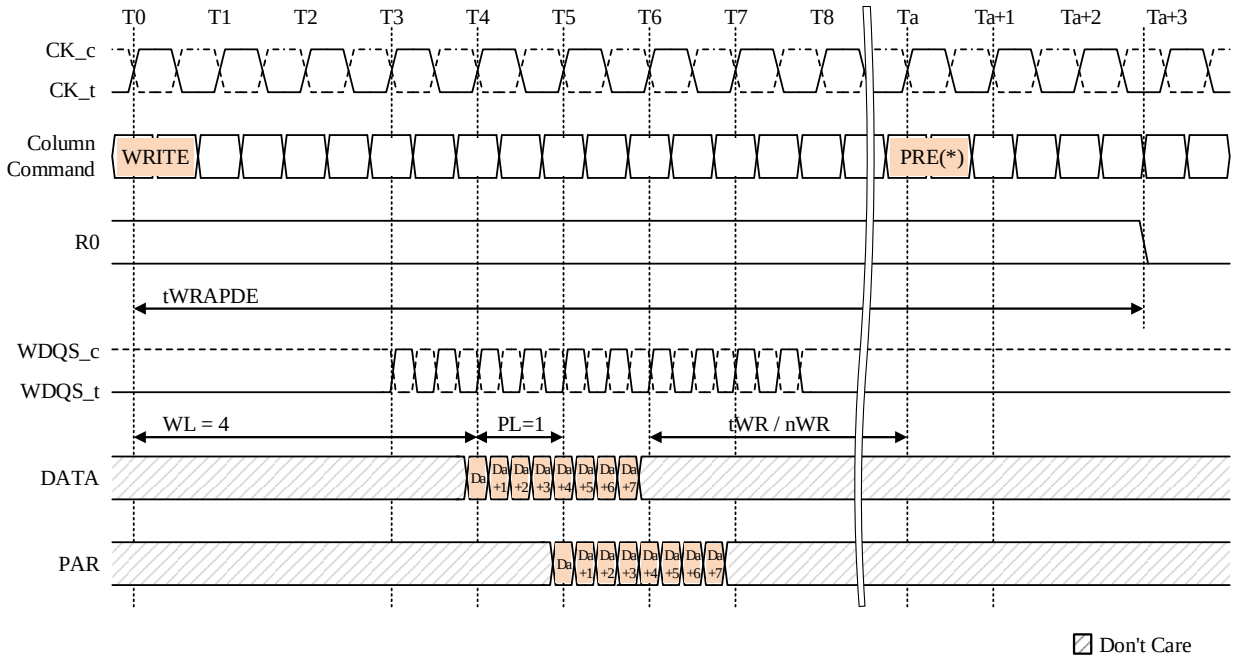
Pin Group	Pin State
RESET_n	H
CK_t, CK_c	L/H or Toggling
R0	L
R[9:1]	X
C[7:0]	X
APAR, ARFU	X
AERR	L
DQ, DBI, ECC, SEV, DPAR	X
WDQS_t, WDQS_c	L/H
RDQS_t, RDQS_c	L/H
DERR	L
CATTRIP	V
NOTE 1 For the pin state description, the following definitions apply: a) "L" is defined as "LOW", and "H" is defined as "HIGH" b) "X" is defined as "Don't Care", and "V" is defined as "Valid"	



NOTE 1 PRE indicates the internal auto-precharge for RDA commands.
NOTE 2 BL = 8, RL = 6 and PL =1 are shown as examples.
NOTE 3 R0 must be used for command except for PDE or address until the end of the read burst operation.

Figure 52 — READ or READ with Auto Precharge to Power-Down Entry Timing

6.3.4.1 Power-Down (PDE, PDX) (cont'd)



NOTE 1 PRE indicates the internal auto-precharge for WRA commands.
NOTE 2 BL = 8, WL = 4 and PL = 1 are shown as examples.
NOTE 3 R0 must be used for address until the end of the write burst operation.
NOTE 4 t_{WR} is the analog value used with WR commands.
NOTE 5 nWR is the number of clock cycles programmed for WR in the Mode Register and used with WRA commands.

Figure 53 — WRITE or WRITE with Auto Precharge to Power-Down Entry Timing

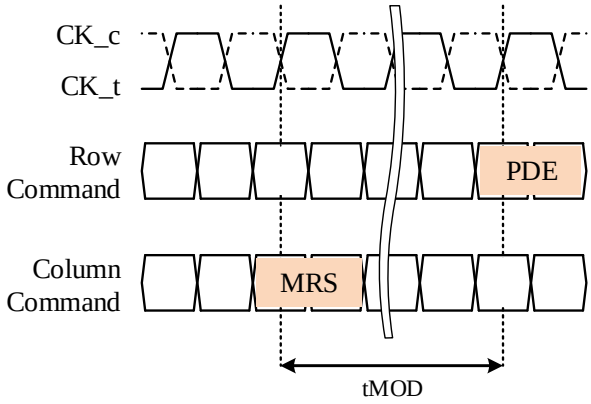
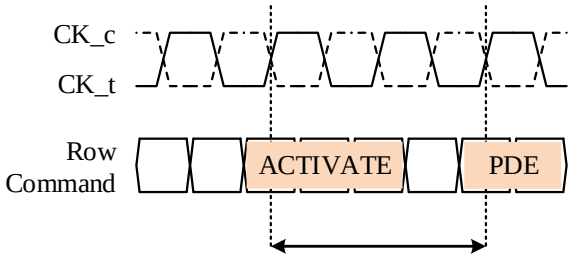


Figure 54 — MODE REGISTER SET to Power-Down Entry Timing

6.3.4.1 Power-Down (PDE, PDX) (cont'd)



NOTE 1 Upon power-down entry the clock must be kept active for the number of clock cycles programmed for RAS in the Mode Register.

Figure 55 — ACTIVATE to Power-Down Entry Timing

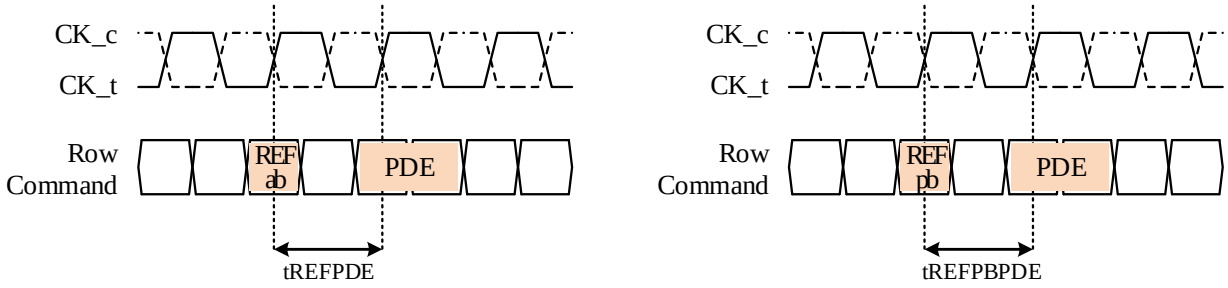


Figure 56 — REFab or REFpb to Power-Down Entry Timing

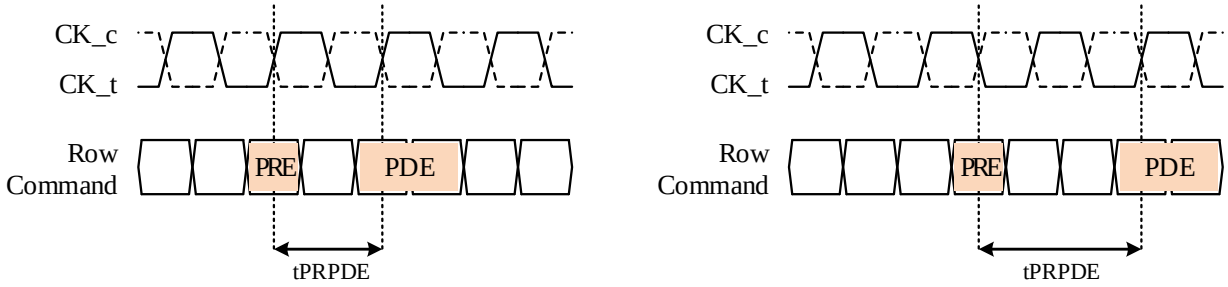


Figure 57 — PRECHARGE to Power-Down Entry Timing

6.3.4.2 Self Refresh (SRE, SRX)

Self refresh can be used to retain data in the HBM4 device, even if the rest of the system is powered down. When in the self refresh mode, the HBM4 device retains data without external clocking. The command is received on the row command inputs R[9:0] as shown in Figure 58 and requires a CNOP command on the column command inputs C[7:0].

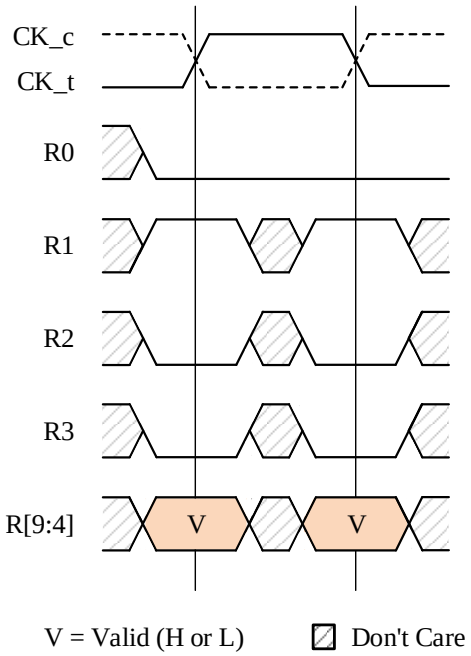


Figure 58 — Self-Refresh Entry Command

Self refresh entry is only allowed when all banks in both pseudo channels are precharged with tRP satisfied, the last data elements from a preceding READ command have been pushed out (t_{RDSRE}), or t_{MOD} from a preceding MODE REGISTER SET command is met. PDE and CNOP commands are required after entering self refresh mode until t_{CPDED} is met.

Once the SELF REFRESH-ENTRY command is registered, R0 must be held LOW to keep the device in self refresh mode. For proper self refresh operation, all power supply pins (VDDC, VDDQ, VPP, VDDQL) must be at valid levels. The HBM4 device initiates a minimum of one internal refresh within t_{CKSR} period once it enters self refresh mode.

The clocks are internally disabled during self refresh operation to save power. The minimum time that the HBM4 device must remain in self refresh mode is t_{CKSR}. The user may halt the external clock or change the external clock frequency t_{CKSRE} after self refresh entry is registered. However, the clock must be restarted and stable t_{CKSRX} before the device can exit self refresh operation.

To ensure that there is enough time to internally process the self refresh entry, POWER DOWN ENTRY and CNOP commands have to be maintained for t_{CPDED} period following the SELF REFRESH ENTRY command. Also, the CK clock must be held stable for t_{CKSRE} cycle.

6.3.4.2 Self Refresh (SRE, SRX) (cont'd)

Once t_{CPDED} and t_{CKSRE} have been met, the pins shall have the following states (see Table 46):

- The RESET_n and R0 receiver remains active; RESET_n = HIGH and R0 = LOW must be maintained to keep the HBM4 DRAM in self refresh;
- The CK clock receiver is disabled; the clock may be stopped, or the clock frequency may be changed; MRS required to set after t_{XSMRSF} in case of frequency changed; the clock must be stable again with t_{CH}(min) and t_{CL}(min) satisfied at least t_{CKSRX} cycles prior to self refresh exit;
- WDQS_t = static LOW and WDQS_c = static HIGH levels must be maintained, respectively;
- RDQS_t and RDQS_c continue driving static LOW and HIGH levels, respectively;
- AERR, DERR continue driving static LOW levels;
- CATTRIP continues driving valid HIGH or LOW levels;
- All other input and output buffers are deactivated.

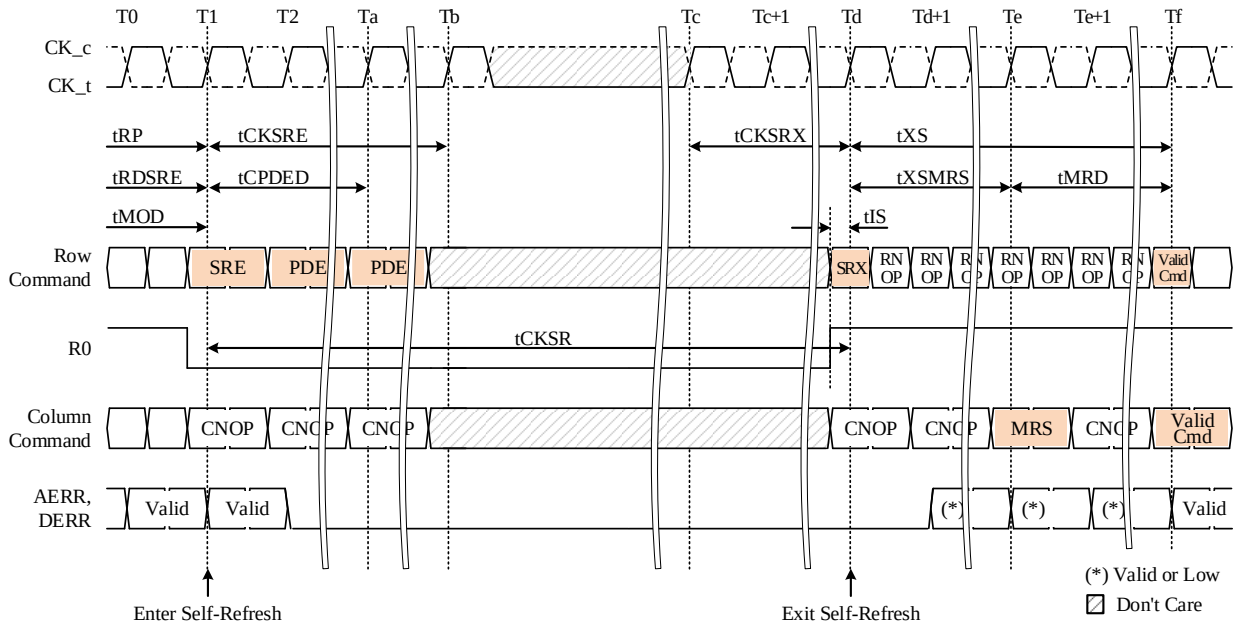
If CA parity is enabled, parity is evaluated for the SELF REFRESH ENTRY command. The HBM4 device requires PDE and CNOP commands with valid parity for the entire t_{CPDED} period, while it will suspend parity checking after self refresh entry and drive AERR to a static LOW. DERR remains LOW as there are no data bursts in progress at this time.

Parity is not evaluated for the SELF REFRESH EXIT command. The HBM4 device requires RNOP and CNOP commands with valid parity for the entire t_{XS} period, while within t_{XS} period it will resume parity checking and indicating parity errors on AERR. DERR remains LOW as there are no data bursts in progress at this time.

The procedure for exiting self refresh requires a sequence of events. First, the CK clock must be stable prior to R0 going back HIGH. A delay of at least t_{XS} must be satisfied before a valid command can be issued to the device to allow for completion of any internal refresh in progress.

Upon exit from self refresh, the HBM4 device can be put back into self refresh mode after waiting at least t_{XS} period.

6.3.4.2 Self Refresh (SRE, SRX) (cont'd)



NOTE 1 Only PDE and CNOP commands are allowed during t_{CPDED} period. Only RNOP and CNOP commands are allowed during t_{XS} periods, except for MRS commands which are allowed t_{XSMRS}(or t_{XSMRSF} when in case of frequency changed, t_{XSMRSF} can be required longer than t_{XS}) after self-refresh exit.
NOTE 2 Write bursts must have been completed with t_{RP} satisfied prior to self-refresh entry.
NOTE 3 Read bursts must have been completed with t_{RDSRE} satisfied prior to self-refresh entry.
NOTE 4 Address inputs are "Don't Care" for self-refresh entry and exit.
NOTE 5 AERR, DERR are driven LOW when parity check is suspended during self-refresh. Signals are shown with t_{PARAC}=0 and t_{PARDQ}=0 for illustration purpose.
NOTE 6 PDE commands after SRE are treated as a RNOP and does not issue a power down entry.

Figure 59 — Self-Refresh Entry and Exit

6.3.4.2 Self Refresh (SRE, SRX) (cont'd)

Table 46 — Pin State Description in Self Refresh

Pin Group	Pin State
RESET_n	H
CK_t, CK_c	X
R0	L
R[9:1]	X
C[7:0]	X
APAR, ARFU	X
AERR	L
DQ, DBI, ECC, SEV, DPAR	X
WDQS_t, WDQS_c	L/H
RDQS_t, RDQS_c	L/H
DERR	L
CATTRIP	V
NOTE 1 For the pin state description, the following definitions apply: a) "L" is defined as "LOW", and "H" is defined as "HIGH" b) "X" is defined as "Don't Care", and "V" is defined as "Valid"	

6.4 Parity

6.4.1 Command/Address Parity

The HBM4 DRAM includes a command/address parity checking function controlled by the CAPAR bit in MR0 OP6. The function is disabled by default. The APAR input and AERR output are associated with the function. APAR is enabled only when the function is enabled.

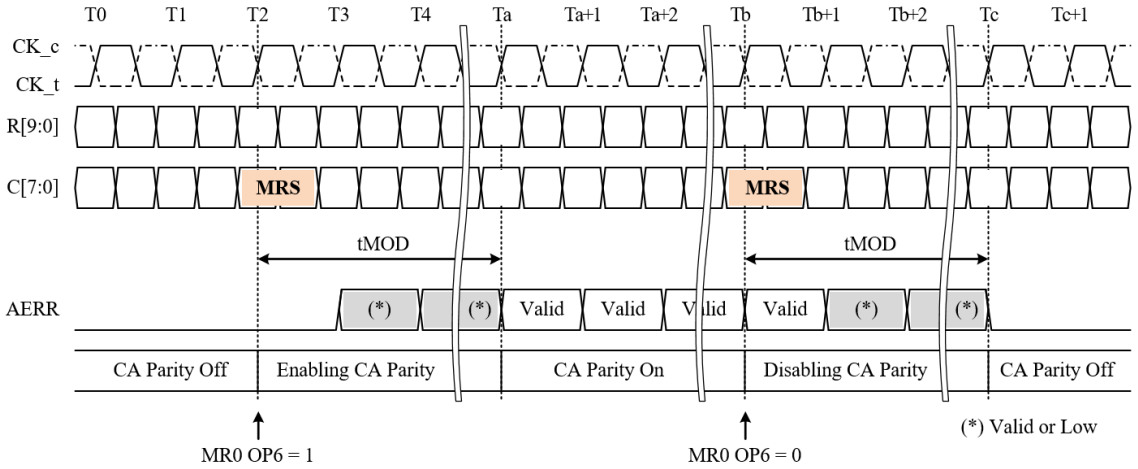
If enabled, the parity is calculated every CK clock cycle separately on both the rising and falling CK clock edges over input signals R[9:0], C[7:0], ARFU and APAR as summarized in Table 47. The AERR output indicates whether a parity error has occurred or not on either the rising or falling CK clock edge (or both edges). The HBM4 DRAM executes commands regardless of command/address parity errors.

Table 47 — Command/Address Parity Function Table

INPUTS	Sum of Inputs Received HIGH	AERR
R[9:0], C[7:0], ARFU, APAR	Even	LOW
	Odd	HIGH
NOTE 1 See Command Truth Tables for command and device state exceptions.		

The HBM4 DRAM may begin to check parity on the next clock cycle following the MODE REGISTER SET command that enables the parity checking function; it will have the parity check enabled latest when t_{MOD} has expired after that MODE REGISTER SET command. The HBM4 DRAM therefore requires all subsequent commands including RNOP and CNOP to be issued with correct parity until when t_{MOD} has expired for the MODE REGISTER SET command that disables the parity calculation. See also the Power-Down and Self Refresh sections. AERR is driven LOW by the HBM4 DRAM at reset.

For every parity error, AERR is driven HIGH for 1 t_{CK}, t_{PARAC} after the corresponding cycle of the error inputs. In the case of consecutive errors, the AERR signal will stay HIGH during the next cycle. The parity function should not be disabled within t_{PARAC} after an access command.

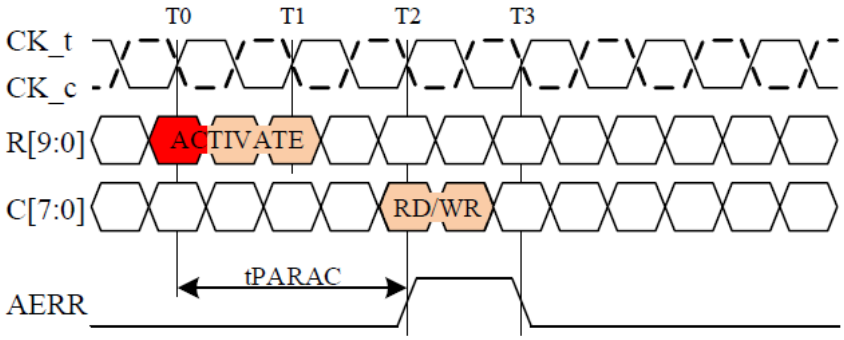


NOTE 1 For illustration purpose, t_{PARAC} is shown with 0 t_{CK} digital and 0 ns analog output delay.
NOTE 2 See Power-Down and Self Refresh sections for details on disabling and enabling parity check in conjunction with power-down and self refresh entry and exit.

Figure 60 — Enabling and Disabling Command/Address Parity

6.4.1 Command/Address Parity (cont'd)

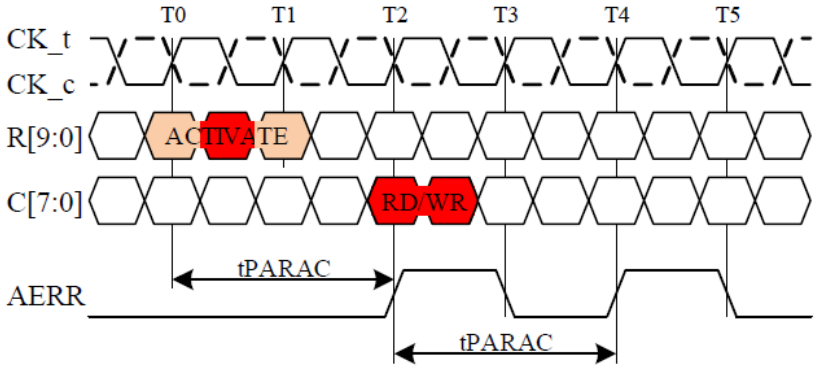
Figure 61 illustrates a single parity error occurrence on the R inputs. In this case, the error occurs at the rising edge of the first cycle of the ACTIVATE command at time T0. After t_{PARAC}, AERR is driven HIGH for 1 t_{CK} and then LOW since no subsequent errors occur.



NOTE 1 For illustration purpose, t_{PARAC} is shown with 2 t_{CK} digital and 0 ns analog output delay.
NOTE 2 MR0 OP6 shall be maintained as 1 for at least t_{PARAC} after the access command.

Figure 61 — Single Command/Address Parity Error

Figure 62 illustrates parity error occurrences on the R and the C inputs. In this case, the error occurs at the falling edge of the first cycle of the ACTIVATE command at time T0. After t_{PARAC}, AERR is driven HIGH for 1 t_{CK} and then LOW for 1 t_{CK}. Since an error also occurs in T2 at both the rising and the falling edges of the READ or WRITE command, the AERR is again driven HIGH for 1 t_{CK} and then LOW since no subsequent errors occur.

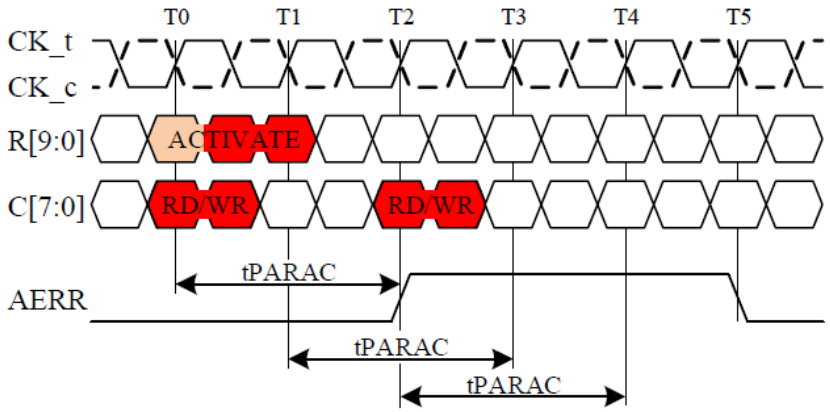


NOTE 1 For illustration purpose, t_{PARAC} is shown with 2 t_{CK} digital and 0 ns analog output delay.
NOTE 2 MR0 OP6 shall be maintained as 1 for at least t_{PARAC} after the access command.

Figure 62 — Separated Command/Address Parity Errors

6.4.1 Command/Address Parity (cont'd)

Figure 63 illustrates consecutive parity error occurrences on the R and the C inputs during the T0, T1, and T2 cycles and either the rising, the falling or both clock edges. Due to the common AERR output, parity error occurrences on both interfaces are indistinguishable.



NOTE 1 For illustration purpose, t_{PARAC} is shown with 2 t_{CK} digital and 0 ns analog output delay.
NOTE 2 MR0 OP6 shall be maintained as 1 for at least t_{PARAC} after the access command.

Figure 63 — Consecutive Command/Address Parity Errors

6.4.2 Data Parity

The HBM4 DRAM includes a data parity checking function for writes controlled by the WPAR bit in MR0 OP5, and a data parity generation function for reads controlled by the RPAR bit in MR0 OP4. Both WPAR and RPAR functions are disabled by default. There is one DPAR bidirectional DDR I/O and one DERR output signal per DWORD associated with the function. The DPAR input is enabled with WPAR during writes, and the DPAR output is enabled with RPAR during reads, otherwise DPAR is disabled.

The data parity function includes a programmable parity latency PL between the corresponding data and the DPAR signal. PL is programmed in MR1 OP[7:5], and is the same for writes and reads. The corresponding DPAR signal will be received and sent PL cycles later. The WDQS and RDQS strobes will have additional strobe cycles with the same preamble and postambles to accommodate the latching of the delayed DPAR signal at both ends. Examples of reads and writes with DQ parity enabled can be found in the Write Command and Read Command sections. The DRAM vendor's datasheet shall be consulted for the range of supported PL values.

On read transactions, the HBM4 DRAM generates parity and transmits the parity on the DPAR signal along with the corresponding data on DQ, DBI and ECC.

On write transactions, the HBM4 DRAM compares the DPAR input with the corresponding data received on DQ, DBI and ECC inputs as summarized in Table 48. The parity calculation is performed separately for each UI of a write burst.

If an error occurs in any single or in multiple UIs within one clock cycle of a write burst (D0 ... D3 or D4 ... D7), DERR is driven HIGH for 1 t_{CK}, t_{PARDQ} after the corresponding cycle of error inputs. The t_{PARDQ} interval for errors occurring during the first clock cycle of a write burst begins (WL + PL) clock cycles after the WRITE command was issued. In case of errors within the first and the second clock cycle of a write burst, DERR will stay HIGH during the next cycle. DERR is driven LOW by the HBM4 DRAM at reset.

When an error occurs, the HBM4 DRAM does not block the write data. The HBM4 DRAM completes the write transaction to the array as normal.

WPAR should not be disabled within (WL + PL + t_{PARDQ} + 2 t_{CK}) after the WRITE command in order to not create a conflict with any ongoing parity operation. For the same reason RPAR should not be disabled within t_{RDMRS} after the READ command.

As outlined in Table 48, meta data received and sent via the ECC signals are included in the parity check and parity generation only when these signals are enabled by the MD bit in MR9 OP0. Similarly, the DBI signals are included in the parity check and parity generation only when these signals are enabled by the WDBI and RDBI bits in MR0 OP[1:0]. The SEV signals are not included in the parity check or parity generation.

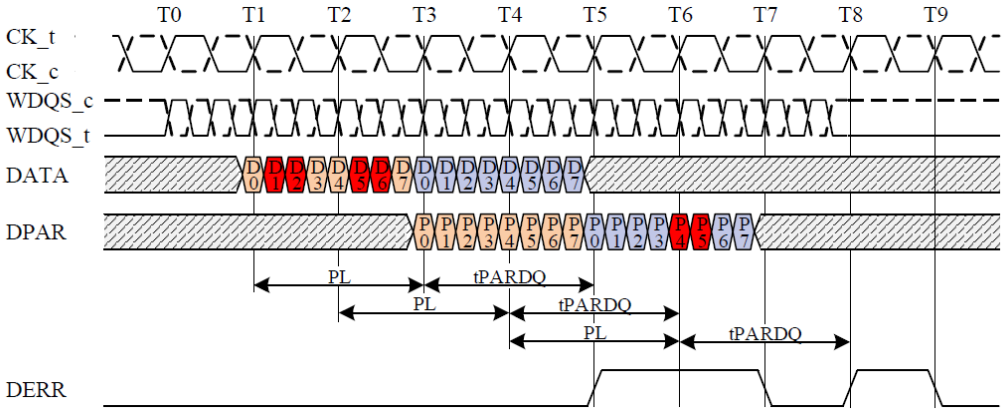
6.4.2 Data Parity (cont'd)

Table 48 — Data Parity Function Table

CONFIGURATION		INPUTS		Sum of Inputs Received HIGH	DERR
MD (MR9 OP0)	WDBI or RDBI (MR0 OP[1:0])	DWORD0	DWORD1		
Enabled	Enabled	DQ[31:0], ECC[1:0], DBI[3:0], DPAR0	DQ[63:32], ECC[3:2], DBI[7:4], DPAR1	Even	LOW
				Odd	HIGH
	Disabled	DQ[31:0], ECC[1:0], DPAR0	DQ[63:32], ECC[3:2], DPAR1	Even	LOW
				Odd	HIGH
Disabled	Enabled	DQ[31:0], DBI[3:0], DPAR0	DQ[63:32], DBI[7:4], DPAR1	Even	LOW
				Odd	HIGH
	Disabled	DQ[31:0], DPAR0	DQ[63:32], DPAR1	Even	LOW
				Odd	HIGH

NOTE 1 The DBI inputs are disabled and excluded from the parity check when WDBI is disabled in MR0 OP1. The DBI outputs are disabled and excluded from the parity generation when RDBI is disabled in MR0 OP0. The ECC I/Os are disabled and excluded from the parity check and parity generation when MD is disabled in MR9 OP0.

Figure 64 illustrates data parity error occurrences on two seamless write bursts. In this example errors occur in the second (D1), third (D2), sixth (D5) and seventh (D6) UI of the first write burst, and in the fifth (P4) and sixth (P5) UI of the DPAR input of the second write burst. After t_{PARDQ}, DERR is driven HIGH for 2 t_{CK} at T5 and T6, then driven LOW for 1 t_{CK} and again driven HIGH for 1 t_{CK} at T8.

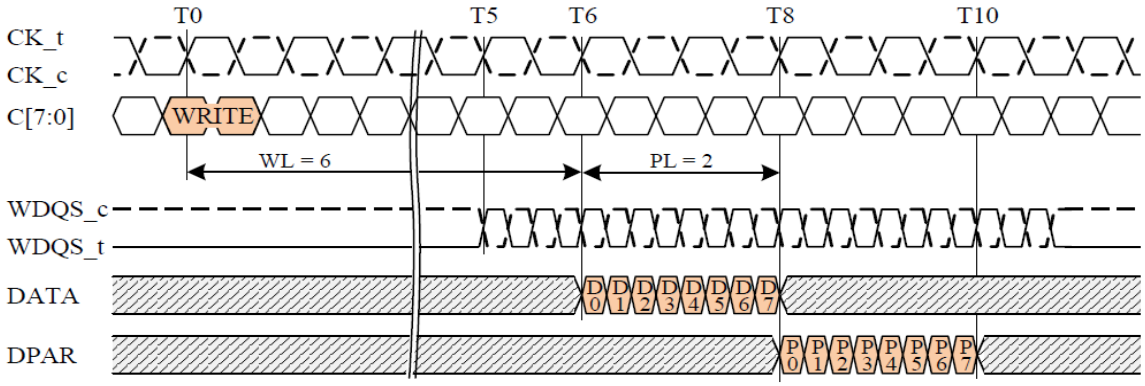


NOTE 1 D0 … D7 = data-in for WRITE command (BL8 burst). P0 … P7 = parity-in for WRITE command.
NOTE 2 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2] for PC1. WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
NOTE 3 Two seamless bursts are shown, with parity errors in the second (D1), third (D2), sixth (D5) and seventh (D6) UI of the first write burst, and in the fifth (P4) and sixth (P5) UI of the DPAR input of the second write burst.
NOTE 4 PL=2 is assumed.
NOTE 5 The parity check is performed separately for the first clock cycle (UI = D0 … D3) and the second clock cycle (UI = D4 … D7) of a BL8 burst.
NOTE 6 tPARDQ is shown with 2 tCK digital and 0 ns analog output delay.
NOTE 7 WDBI could be on or off and is controlled with MR0 OP1. WDBI shall be maintained enabled for at least tPARDQ after the access command.

Figure 64 — Write Parity Errors with PL = 2

6.4.2 Data Parity (cont'd)

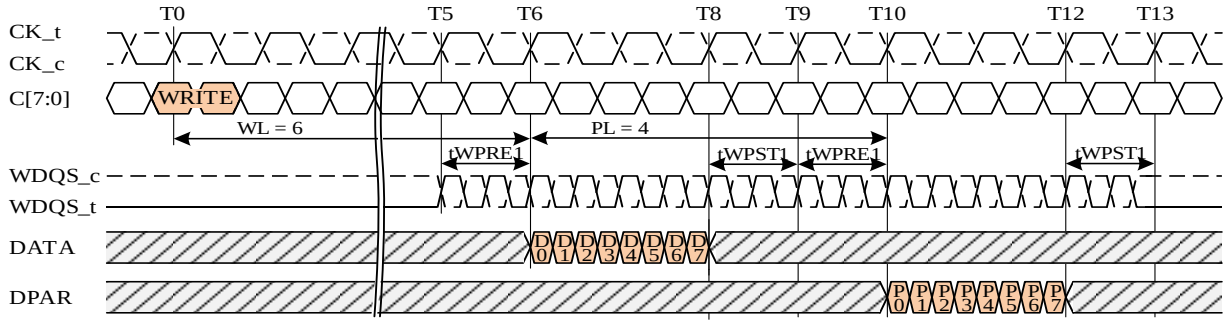
Examples of single write bursts with write data parity enabled are shown in Figure 65 – Write Parity Alignment with PL = 2 and Figure 66 – Write Parity Alignment with PL = 4. With PL= 2 four additional WDQS pulses are received at cycles T8 and T9 to latch the DPAR input.



NOTE 1 D0 … D7 = data-in for WRITE command (BL8 burst). P0 … P7 = parity-in for WRITE command.
NOTE 2 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2] for PC1. WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
NOTE 3 WL = 6 and PL=2 is assumed.
NOTE 4 WDBI could be on or off and is controlled with MR0 OP1.

Figure 65 – Write Parity Alignment with PL = 2

With PL = 4, the WDQS postamble for the write data at cycle T8 is immediately followed by the WDQS preamble for the write parity data at cycle T9, resulting in continuous WDQS pulses over the write data and data parity bursts.

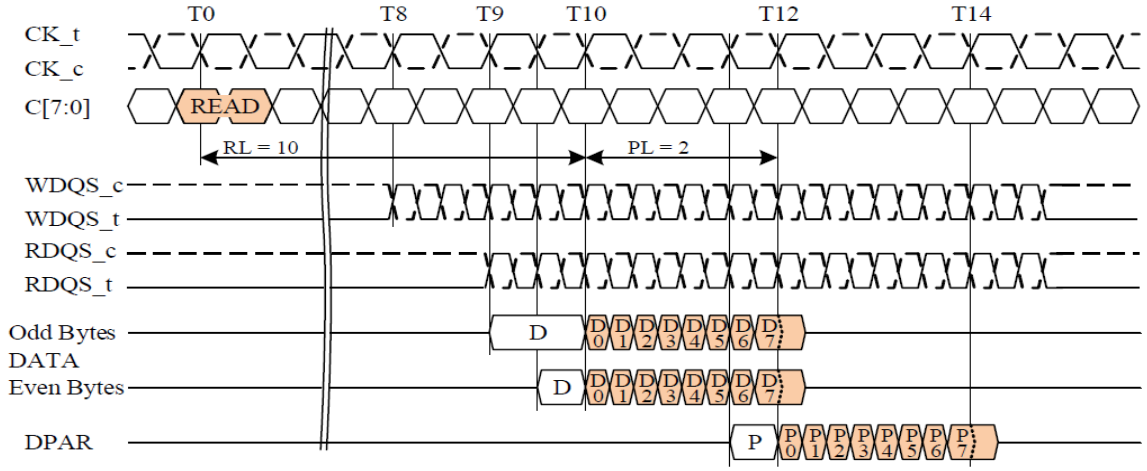


NOTE 1 D0 … D7 = data-in for WRITE command (BL8 burst). P0 … P7 = parity-in for WRITE command.
NOTE 2 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2] for PC1. WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
NOTE 3 WL = 6 and PL=4 is assumed.
NOTE 4 WDBI could be on or off and is controlled with MR0 OP1.

Figure 66 – Write Parity Alignment with PL = 4

Examples of a single read bursts with read data parity enabled are shown in Figure 67 – Read Parity Alignment with PL = 2 and Figure 68 – Read Parity Alignment with PL = 4. The DPAR output is

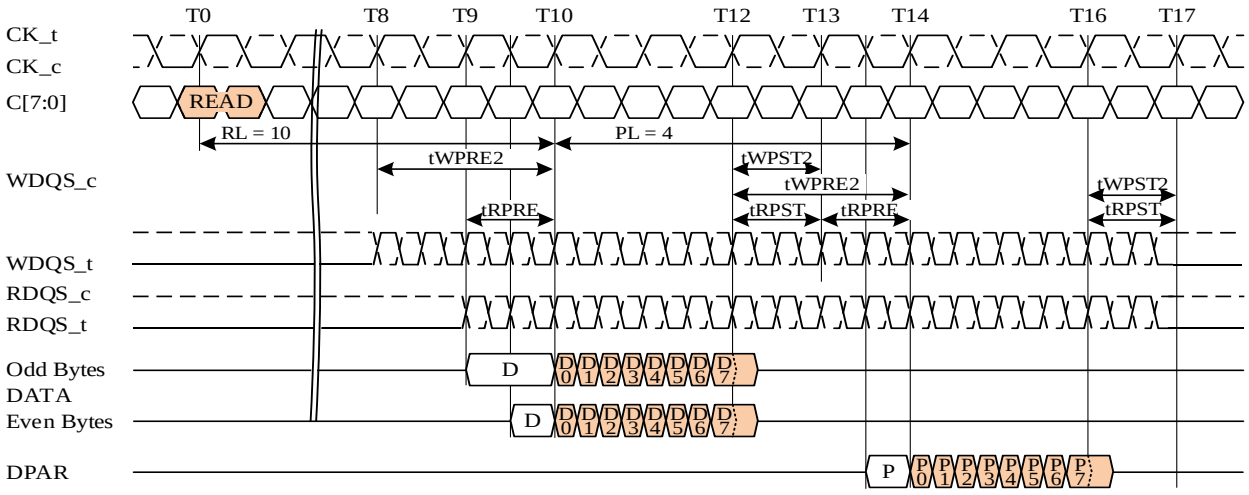
preconditioned over half a clock cycle like for the even data bytes. With PL = 2 four additional WDQS and RDQS pulses for DPAR are received and returned at cycles T12 and T13.



NOTE 1 D0 … D7 = data-out for READ command (BL8 burst). P0 … P7 = parity-out for READ command.
NOTE 2 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2] for PC1. RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1. WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
NOTE 3 RL = 10 and PL=2 is assumed.
NOTE 4 RDBI could be on or off and is controlled with MR0 OP0.

Figure 67 – Read Parity Alignment with PL = 2

With PL = 4, the WDQS postamble for the read data at cycle T12 overlaps with the WDQS preamble for the read parity data. The RDQS postamble for the read data at cycle T12 is immediately followed by the RDQS preamble for the read parity data at cycle T13, both resulting in continuous WDQS and RDQS pulses over the read data and data parity bursts.



NOTE 1 D0 … D7 = data-out for READ command (BL8 burst). P0 … P7 = parity-out for READ command.
NOTE 2 DATA = DQ[31:0], DBI[3:0], ECC[1:0] for PC0, and DQ[63:32], DBI[7:4], ECC[3:2] for PC1. RDQS_t/_c is RDQS0_t/_c for PC0, and RDQS1_t/_c for PC1. WDQS_t/_c is WDQS0_t/_c for PC0, and WDQS1_t/_c for PC1.
NOTE 3 RL = 10 and PL=4 is assumed.
NOTE 4 RDBI could be on or off and is controlled with MR0 OP0.

Figure 68 – Read Parity Alignment with PL = 4

6.5 Clock Frequency Change Sequence

Clock Frequency changes can occur during self refresh mode only. When the CK clock is stopped after self refresh entry, it can be restarted at a different frequency. If the change in clock-rate requires changes to configuration parameters, MRS commands immediately prior to or after self refresh mode may be required.

6.6 Temperature Compensated Refresh Reporting

TBD

6.6.1 Catastrophic Temperature Sensor

The CATTRIP sensor logic detects if the junction temperature of any die in the HBM4 stack exceeds a catastrophic trip-point level. The level is set by the DRAM vendor to a value below the temperature that would result in permanent damage of the device. If the junction temperature anywhere in the stack exceeds that catastrophic trip-point level, the HBM4 device will drive the CATTRIP pin to HIGH.

The CATTRIP output is sticky in that device power-off is required to clear the CATTRIP output to LOW. Sufficient time should be allowed for the device to cool after a CATTRIP event. See HBM4 Power-Up and Initialization Sequence for the initialization of the CATTRIP pin.

The circuits associated with the CATTRIP pin will operate correctly even if the catastrophic trip-point level has been exceeded, and regardless of whether the external or internal clocks have stopped. The functionality of CATTRIP can be verified by writing a “1” to MR7 OP7 to force CATTRIP to HIGH, and “0” to set CATTRIP back to LOW.

6.7 Interconnect Redundancy Remapping

The HBM4 DRAM supports interconnect lane remapping to help improve SIP assembly yield and recover functionality of the HBM4 stack. The SOFT_LANE_REPAIR and HARD_LANE_REPAIR instructions are used to perform lane remapping. Lane remapping is independent for each channel. WSO remapping is associated with Channel 0 for WSO0 thru WSO15 and Channel 16 for WSO16 thru WSO31.

SOFT_LANE_REPAIR and HARD_LANE_REPAIR instructions can only be issued as part of the device initialization and before normal memory operation has commenced, e.g. before the CK clock has started to toggle.

The HBM DRAM can be programmed to retain the remapped lane information even when power is completely removed from the HBM stack.

6.7.1 AWORD Remapping

There is one redundant AWORD lane per channel to either repair one lane in the row command bus or one lane in the column command bus. APAR and ARFU are associated with the column command bus repair as shown in Table 50. CK_c, CK_t, and AERR signals cannot be remapped. After a lane is remapped, the input buffer associated with the broken lane is turned off and the input buffer associated with the redundant bump (RA) is turned on. All functionalities are preserved with row or column bus lane remapping.

6.7.1.1 Row Command Bus – Remapping Table

Table 49 — AWORD - Row Command Bus Remapping

Description	Register Encoding	R1	R2	R3	R0	R4	R5	R6	R7	R8	R9	RA
Repair Lane 0	0000	XX	R1	R2	R3	R0	R4	R5	R6	R7	R8	R9
Repair Lane 1	0001	R1	XX	R2	R3	R0	R4	R5	R6	R7	R8	R9
Repair Lane 2	0010	R1	R2	XX	R3	R0	R4	R5	R6	R7	R8	R9
Repair Lane 3	0011	R1	R2	R3	XX	R0	R4	R5	R6	R7	R8	R9
Repair Lane 4	0100	R1	R2	R3	R0	XX	R4	R5	R6	R7	R8	R9
Repair Lane 5	0101	R1	R2	R3	R0	R4	XX	R5	R6	R7	R8	R9
Repair Lane 6	0110	R1	R2	R3	R0	R4	R5	XX	R6	R7	R8	R9
Repair Lane 7	0111	R1	R2	R3	R0	R4	R5	R6	XX	R7	R8	R9
Repair Lane 8	1000	R1	R2	R3	R0	R4	R5	R6	R7	XX	R8	R9
Repair Lane 9	1001	R1	R2	R3	R0	R4	R5	R6	R7	R8	XX	R9
Reserved	1010 to 1110	R1	R2	R3	R0	R4	R5	R6	R7	R8	R9	RA
Default – No Repair	1111	R1	R2	R3	R0	R4	R5	R6	R7	R8	R9	RA

NOTE 1 XX = Lane is remapped

6.7.1.2 Column Command Bus – Remapping Table

Table 50 — AWORD - Column Command Bus Remapping

Description	Register Encoding	C0	C1	C2	C3	C4	C5	C6	C7	APAR	ARFU	RA
Repair Lane 0	0000	XX	C0	C1	C2	C3	C4	C5	C6	C7	APAR	ARFU
Repair Lane 1	0001	C0	XX	C1	C2	C3	C4	C5	C6	C7	APAR	ARFU
Repair Lane 2	0010	C0	C1	XX	C2	C3	C4	C5	C6	C7	APAR	ARFU
Repair Lane 3	0011	C0	C1	C2	XX	C3	C4	C5	C6	C7	APAR	ARFU
Repair Lane 4	0100	C0	C1	C2	C3	XX	C4	C5	C6	C7	APAR	ARFU
Repair Lane 5	0101	C0	C1	C2	C3	C4	XX	C5	C6	C7	APAR	ARFU
Repair Lane 6	0110	C0	C1	C2	C3	C4	C5	XX	C6	C7	APAR	ARFU
Repair Lane 7	0111	C0	C1	C2	C3	C4	C5	C6	XX	C7	APAR	ARFU
Repair Lane 8	1000	C0	C1	C2	C3	C4	C5	C6	C7	XX	APAR	ARFU
Repair Lane 9	1001	C0	C1	C2	C3	C4	C5	C6	C7	APAR	XX	ARFU
Reserved	1010 to 1110	C0	C1	C2	C3	C4	C5	C6	C7	APAR	ARFU	RA
Default – No Repair	1111	C0	C1	C2	C3	C4	C5	C6	C7	APAR	ARFU	RA
NOTE 1 XX = Lane is remapped												

6.7.1.3 AWORD Remapping Examples

As an example, C0_4 is the broken lane in the Column Command bus with no broken lanes on the Row Command bus. The lane is remapped by programming Channel 0's LANE REPAIR WDR bits AWORD_CA[3:0] to 4h and AWORD_RA[3:0] to Fh.

Table 51 — Original Lane Assignment - Channel 0 - AWORD Column Repair

	ARFU0		C0_7		C0_5		C0_4		C0_2		C0_0
RA0		APAR0		C0_6		CK0_t		C0_3		C0_1	
	R0_9		R0_7		CK0_c		R0_4		R0_3		R0_1
AERR0		R0_8		R0_6		R0_5		R0_0		R0_2	

Table 52 — Remapped Lane Assignment - Channel 0 - AWORD Column Repair

	APAR0		C0_6		C0_4		XX		C0_2		C0_0
ARFU0		C0_7		C0_5		CK0_t		C0_3		C0_1	
	R0_9		R0_7		CK0_c		R0_4		R0_3		R0_1
AERR0		R0_8		R0_6		R0_5		R0_0		R0_2	

6.7.1.3 AWORD Remapping Examples (cont'd)

In a second example, R0_0 is the broken lane in the Row Command bus with no broken lanes on the Column Command bus. The lane is remapped by programming Channel 0's LANE REPAIR WDR bits AWORD_RA[3:0] to 0h and AWORD_CA[3:0] to Fh.

Table 53 — Original Lane Assignment - Channel 0 - AWORD Row Repair

	ARFU0		C0_7		C0_5		C0_4		C0_2		C0_0
RA0		APAR0		C0_6		CK0_t		C0_3		C0_1	
	R0_9		R0_7		CK0_c		R0_4		R0_3		R0_1
AERR0		R0_8		R0_6		R0_5		R0_0		R0_2	

Table 54 — Remapped Lane Assignment - Channel 0 - AWORD Row Repair

	ARFU0		C0_7		C0_5		C0_4		C0_2		C0_0
R0_9		APAR0		C0_6		CK0_t		C0_3		C0_1	
	R0_8		R0_6		CK0_c		R0_0		R0_3		R0_1
AERR0		R0_7		R0_5		R0_4		XX		R0_2	

6.7.2 DWORD Remapping

HBM4 supports remapping of one broken data bus lane per double byte. Two adjacent bytes (e.g. DQ[15:0], DQ[31:16], DQ[47:32], DQ[63:48]) are treated as a pair (double byte), but each double byte is treated independently.

After a lane is remapped, the input buffer associated with the broken lane is turned off and the output driver is tri-stated; the input buffer associated with the redundant lane (RD) is additionally turned on and the output driver is activated.

It is required to program "1111b" for the intact byte within the double byte while the remapping for the broken lane in the other byte is encoded according to the table.

DBI functionality is preserved as long as the Mode Register setting for DBI function is enabled. There is no impact on the Data Parity function. WDQS_c, WDQS_t, RDQS_c, RDQS_t, PAR and DERR signals cannot be remapped.

During Reads, the RD output drivers are enabled along with the DQ, DBI and ECC/SEV lanes of the physical byte the lane is located in: RD0 and RD2 are located within even bytes and thus enabled one clock cycle prior to the first valid data bit, and RD1 and RD3 are located within odd bytes and thus enabled two clock cycles prior to the first valid data bit.

6.7.2.1 DWORD Remapping Table

Table 55 — DWORD Remapping (1 Byte)

Description	Register Encoding	ECC0 (ECC1/ SEV0/ SEV1)	DQ0 (DQ8/ DQ16/ DQ24)	DQ1 (DQ9/ DQ17/ DQ25)	DQ2 (DQ10/ DQ18/ DQ26)	DQ3 (DQ11/ DQ19/ DQ27)	DQ4 (DQ12/ DQ20/ DQ28)	DQ5 (DQ13/ DQ21/ DQ29)	DQ6 (DQ14/ DQ22/ DQ30)	DQ7 (DQ15/ DQ23/ DQ31)	DBI0 (DBI1/ DBI2/ DBI3)	RD0 (RD0/ RD1/ RD1)
Repair Lane 0	0000	XX	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0
Repair Lane 1	0001	ECC0	XX	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0
Repair Lane 2	0010	ECC0	DQ0	XX	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0
Repair Lane 3	0011	ECC0	DQ0	DQ1	XX	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0
Repair Lane 4	0100	ECC0	DQ0	DQ1	DQ2	XX	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0
Repair Lane 5	0101	ECC0	DQ0	DQ1	DQ2	DQ3	XX	DQ4	DQ5	DQ6	DQ7	DBI0
Repair Lane 6	0110	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	XX	DQ5	DQ6	DQ7	DBI0
Repair Lane 7	0111	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	XX	DQ6	DQ7	DBI0
Repair Lane 8	1000	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	XX	DQ7	DBI0
Repair Lane 9	1001	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	XX	DBI0
Reserved	1010 to 1110	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0	RD0
Default – No Repair	1111	ECC0	DQ0	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DBI0	RD0

NOTE 1 XX = Lane is remapped
NOTE 2 DWORD0 and DWORD0_BYTE1 are shown as an example
NOTE 3 ECC is associated with DWORD0_BYTE0, DWORD0_BYTE1, DWORD1_BYTE0 and DWORD1_BYTE1
NOTE 4 SEV is associated with DWORD0_BYTE2, DWORD0_BYTE3, DWORD1_BYTE2 and DWORD1_BYTE3

6.7.2.2 DWORD Remapping Example

As an example, ECC0_0 is a broken lane for byte 0 while all lanes for byte 1 are intact. The lane is remapped as illustrated in Table 57 by programming channel 0's LANE REPAIR WDR bits DWORD0_BYTE0[3:0] to 0h and bits DWORD0_BYTE1[3:0] to Fh.

Table 56 — Original DWORD Lane Assignment - Channel 0 – Byte [1:0]

	DQ0_7		DQ0_5		RD0_0		DQa3		DQ0_1		ECC0_0
DBI0_0		DQ0_6		DQ0_4		PAR0_0		DQ0_2		DQ0_0	
	VDDQL		VDDQL		VDDQL		VDDQL		VDDQL		VDDQL
DBI0_1		DQ0_14		DQ0_12		WDQS0_0_t		DQ0_10		DQ0_8	
	DQ0_15		DQ0_13		WDQS0_0_c		DQ0_11		DQ0_9		ECC0_1

Table 57 — Remapped DWORD Lane Assignment - Channel 0 – Byte [1:0]

	DQ0_6		DQ0_4		DBI0_0		DQ0_2		DQ0_0		XX
DQ0_7		DQ0_5		DQ0_3		PAR0_0		DQ0_1		ECC0_0	
	VDDQL		VDDQL		VDDQL		VDDQL		VDDQL		VDDQL
DBI0_1		DQ0_14		DQ0_12		WDQS0_0_t		DQ0_10		DQ0_8	
	DQ0_15		DQ0_13		WDQS0_0_c		DQ0_11		DQ0_9		ECC0_1

6.7.2.2 DWORD Remapping Example (cont'd)

The circuit diagram in Figure 69 illustrates the DQ lane remapping in more detail. Physical micro-bump DQ3 will be connected to internal logical DQ3 input and output paths when the DQ3 lane is not remapped; with remapping the internal DQ3 input and output paths would be routed to the physical DQ4 micro-bump.

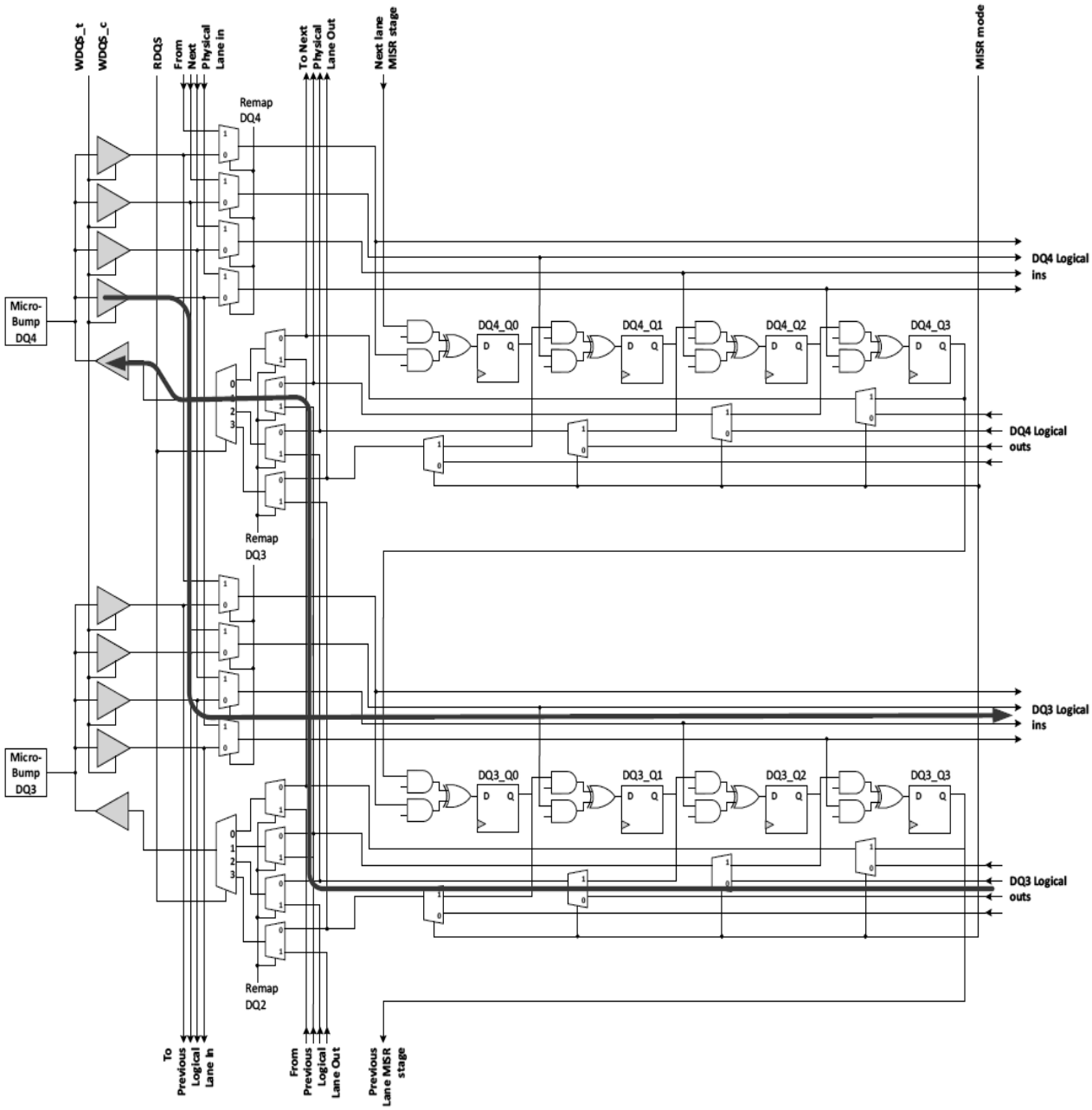


Figure 69 — Example Signal Paths with Lane Repair

6.7.2.3 WSO Remapping Example (cont'd)

As an example, WSO29 is the broken lane in the midstack. The lane is remapped by programming Channel 16's LANE REPAIR WDR bits WSO[4:0] to Dh. All other channels program bypass.

Table 58 — Original Lane Assignment – WSO Repair

RM1		WSO31		WSO30		WSO29		WSO28		WSO27		WSO26

Table 59 — Remapped Lane Assignment – WSO Repair

RM1		WSO31		WSO30		XX		WSO28		WSO27		WSO26

In the second example, WSO29 and WSO15 are the broken lanes in the midstack. The lanes are remapped by programming Channel 16's LANE REPAIR WDR bits WSO[4:0] to Dh and Channel 0's LANE REPAIR WDR bits WSO[4:0] to Fh. All other channels program bypass.

Table 60 — Original Lane Assignment – WSO Repair (2 WSOs Broken)

RM1		WSO31		WSO30		WSO29		WSO28		WSO27		WSO26
RM0		WSO15		WSO14		WSO13		WSO12		WSO11		WSO10

Table 61 — Remapped Lane Assignment – WSO Repair (2 WSOs Broken)

WSO31		WSO30		WSO29		XX		WSO28		WSO27		WSO26
WSO15		XX		WSO14		WSO13		WSO12		WSO11		WSO10

6.8 HBM4 Loopback Test Modes

A Multiple-input Shift Register (MISR) / Linear Feedback Shift Register (LFSR) circuit is defined within the HBM4 AWORD and DWORD I/O blocks. These circuits are intended for testing and training the link between the Host and the HBM4 device. Referring to Figure 70, each byte within a DWORD implements a 40-bit MISR/LFSR circuit, comprised of WDQS 2-cycles Rise and Fall 4-bits for each of the eight DQs plus DBI and ECC/SEV signals. Respective Q0, Q1, Q2, Q3 indicate half WDQS cycle for each one signal within each byte of a DWORD implement. The BL0 to BL3 of HBM4 are matched with the Q0 to Q3 in the front two WDQS cycles and the BL4 to BL7 of HBM4 are matched with the Q0 to Q3 in the next two WDQS cycles. In operation, the MISR/LFSR circuits operate independently across the bytes. The AWORD implements a 38-bit MISR/LFSR circuit comprised of CK DDR Rise and Fall bits for the 18 row and column command bits, plus ARFU. When the MISR registers are read via the IEEE 1500 port DWORD_MISR instruction, the four bytes per DWORD (160-bits) for the two DWORDs within a channel are serially shifted out, for a total of 320-bits. The 38-bit AWORD MISR content is read via the AWORD_MISR instruction. See
Table 122 and Table 123 for the bit-orders for these MISR registers.

The term MISR modes collectively refers to all of the modes - LFSR mode, Register mode, MISR mode, and LFSR Compare mode. AWORD MISR modes and DWORD MISR modes refer to all of the modes defined for the specific bus.

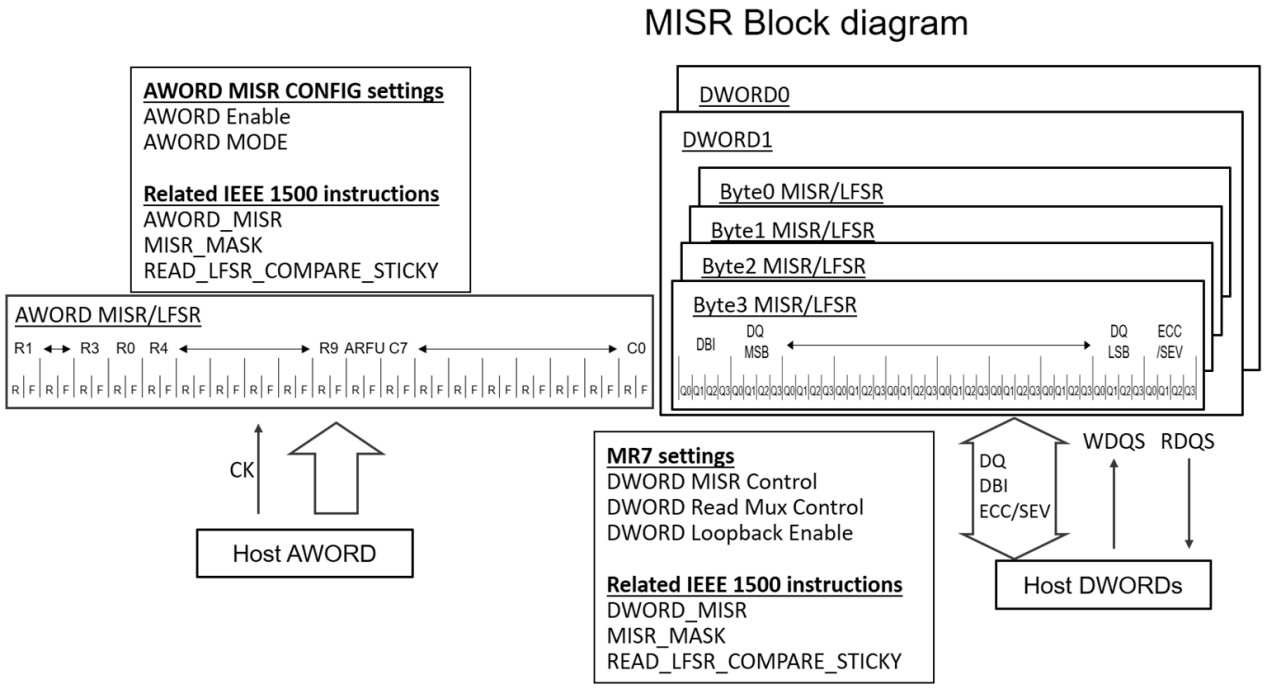


Figure 70 — MISR Features Block Diagram of HBM4

6.8.1 HBM4 Polynomial Structure

Figure 71 provides an example of a 4-bit Galois type MISR/LFSR structure that implements the following polynomial:

$$f(x) = X^4 + X^3 + 1$$

The example circuit and function table are for illustration only, and this circuit's modes are not fully representative of the actual DWORD and AWORD MISR definitions as outlined below. For example, the circuit shown in Figure 71 implements a reset function, while the AWORD and DWORD MISRs instead implement a preset function, where specific bits are set to logic 1.

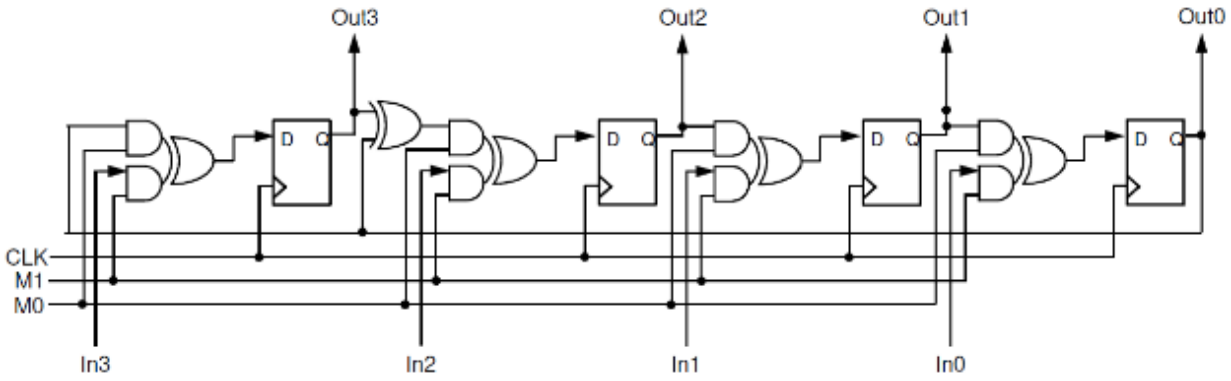


Figure 71 — Example of 4 bit MISR-LFSR implementing $f(x) = X^4 + X^3 + 1$

Table 62 — MISR Function Table

M1	M0	Function
0	0	Reset
0	1	LFSR
1	0	Register
1	1	MISR

6.8.1.1 AWORD MISR Polynomial

The HBM4 AWORD MISR structure is a 38-bit MISR/LFSR with the following polynomial:

$$f(x) = X^{38} + X^6 + X^5 + X + 1$$

The AWORD MISR may be serially accessed via the AWORD_MISR IEEE 1500 port instruction. See Table 123 for the AWORD MISR wrapper data register bit order.

6.8.1.2 DWORD MISR Polynomial

The DWORD MISR structure is a 40-bit MISR/LFSR per byte with the following polynomial:

$$f(x) = X^{40} + X^{38} + X^{21} + X^{19} + 1$$

Note that when the DWORD MISRs are accessed via the DWORD_MISR IEEE 1500 port instructions that all of the individual byte MISRs within a channel are concatenated into a 320-bit wrapper data register. See
Table **122** for the DWORD MISR bit order.

6.8.2 General Loopback Modes Features and Behavior

This section addresses features and behaviors that generally apply to all of the MISR modes.

a) **Entering the MISR modes** – MISR modes may be enabled after tINIT3 within the initialization sequence; they may also be entered any time after completing the initialization (see 4 Initialization). DWORD MISR modes are controlled via MR7 OP[5:3] (see Table 18), while AWORD MISR modes are controlled via the IEEE 1500 port AWORD MISR CONFIG instruction. AWORD and DWORD MISR modes cannot be used simultaneously since the DWORD MISR modes are driven via READ and WRITE commands on the AWORD bus.

b) **Entering and exiting AWORD MISR modes** – HBM4 allows the AWORD MISR modes to be utilized on one or more channels while the other channels continue to operate normally. After normal initialization, to enter the AWORD MISR modes on a given channel the host must put the HBM4 channel into either precharge power-down or self refresh modes. Self refresh mode may be used in order to retain memory content while using the AWORD MISR modes, as needed. AWORD MISR modes may also be enabled after tINIT4 within the initialization sequence. Enabling the AWORD MISR modes re-enables the AWORD I/O buffers that are normally disabled in power-down and self refresh modes, which may result in increased current draw over the IDD2P, IDD2P0 and IDD6x specifications. If returning to normal operation is not required, the host may assert an initialization sequence per ~~clause~~ section Initialization after operating the AWORD MISR modes. The sequence for entering AWORD MISR modes, and then exiting back to normal operation is as follows:

   1) At any time after initializing the HBM4 enter the all banks idle state.
   2) Enter either the precharge power-down state or the self refresh state. R0 = LOW while in these states.
   3) Stop toggling CK (CK_t = LOW, CK_c = HIGH).
   4) Enable/enter and operate the AWORD MISR modes (AWORD_MISR_CONFIG Enable = 1 - On). Finish these operations with CK stopped (CK_t = LOW, CK_c = HIGH) and R0 = LOW.
   5) Disable the AWORD MISR modes and follow the Power-Down (PDE, PDX) or Self Refresh (SRE, SRX) exit procedures.
   6) When using the AWORD MISR modes after t_{INIT3} within the initialization sequence, power-down or self refresh entry and exit does not apply.

   If the DRAM is not required to continue with mission mode operation after AWORD MISR test, there is no requirement on row/column command bus and the precharge power-down state or the self refresh state after loopback test. The AWORD MISR modes (AWORD_MISR_CONFIG Enable bit) can be reset by WRST_n during a subsequent initialization sequence.

6.8.2 General Loopback Modes Features and Behavior (cont'd)

c) **Entering and exiting DWORD MISR modes** – HBM4 allows the DWORD MISR modes to be utilized on one or more channels while the other channels continue to operate normally. After normal initialization (see Initialization), to enter the DWORD MISR modes on a given channel the host must put the HBM4 channel into the all banks idle, enable the DWORD MISR modes (MR7 Loopback Enable = 1 - Enable; see Table 18), and then enter precharge power-down or self refresh. Self refresh may be used in order to retain memory content while using the DWORD MISR modes, as needed. Enabling the DWORD MISR modes before entering precharge power-down or self refresh keeps the AWORD and DWORD I/O buffers enabled, and may result in increased current draw over the IDD2P, IDD2P0 and IDD6x specifications. DWORD MISR modes may also be enabled after t_{INIT3} within the initialization sequence. Also see items f) and h) for related DWORD MISR modes configuration setting. On the column command bus only READ (RD), WRITE (WR), and Column No Operation (CNOP) commands may be issued which operate the DWORD MISR modes, MR14 MRS commands may be issued to set the DWORD VREF (VREFD), and MR7 MRS commands may be issued to select the DWORD MISR modes. On the row command bus only R0 = static LOW may be issued. The sequence for entering DWORD MISR modes, and then exiting back to normal operation is as follows:

1) At any time after initializing the HBM4, enter the all banks idle state.
2) Set all configuration mode registers as needed for use in the DWORD MISR modes (see items h), k), and n)).
3) Set MR7 DWORD Loopback Enable = 1 - Enable, and then wait t_{MOD}.
4) Enter either precharge power-down or self refresh. R0 = LOW while in these states.
5) Select and operate the DWORD MISR modes via MR7 settings (Row command input requires RNOP with R0=L to keep power-down or self refresh status during MR7 setting) and sending RD, WR, and CNOP commands. After completing DWORD MISR operations, send CNOP commands.
6) Follow the power-down exit (PDX) or self refresh exit (SRX) procedures.
7) Set MR7 DWORD Loopback Enable = 0 - Disable, and then wait t_{MOD} before continuing normal operation.

MRS commands are not supported until after t_{INIT5} in the initialization sequences; therefore, to configure and control the mode registers for DWORD MISR modes usage after t_{INIT3} the MODE_REGISTER_DUMP_SET instruction must be used. The sequence for entering and operating the DWORD MISR modes after t_{INIT3} in the initialization sequence is as follows:

1) Start CK with PD and CNOP on the command busses.
2) Using MODE_REGISTER_DUMP_SET sets all configuration mode registers as needed for use in the DWORD MISR modes (see items f) and h)), set MR7 DWORD Loopback Enable = 1 - Enable, and then wait t_{MOD}.
3) Select and operate the DWORD MISR modes via MR7 settings (using MODE_REGISTER_-DUMP_SET) and sending RD, WR, and CNOP commands.
4) After completing DWORD MISR operations, send CNOP commands, set MR7 DWORD Loopback Enable = 0 - Disable using MODE_REGISTER_DUMP_SET, then wait t_{MOD}.
5) CK clocking may be stopped if desired.

6.8.2 General Loopback Modes Features and Behavior (cont'd)

6) Proceed to other IEEE 1500 instructions, or proceed with the initialization sequence from Figure 6 time Td.

If the DRAM is not required to continue with mission mode operation after DWORD MISR test, there is no requirement to follow the power-down or self refresh procedures and set MR7 DWORD Loopback Enable = 0 - Disable. The Loopback Enable bit can be reset by a subsequent initialization sequence with RESET_n = LOW.

d) **Command decode is disabled in AWORD MISR modes** - When AWORD MISR modes are enabled the traffic sent on the AWORD bus is not limited to valid commands. To prevent undefined states and operations, when AWORD MISR modes are enabled (AWORD MISR CONFIG Enable = 1 - On), command decoding is disabled.

e) **With lane repairs the MISR bit positions remain with their logical signals** - The MISR bits are associated with their logic signals, not the physical microbumps (see Figure 69). For example, if DQ3 has been repaired (which routes the DQ3 data to the DQ4 microbump) the data received on the DQ4 microbump is routed to the DQ3 MISR bits. Effectively, the behaviors for all MISR modes are unchanged - all 10 bits of the byte are captured in the MISR in the same bit locations, as if no lane repair were active.

f) **HBM4 DBI, and ECC/SEV logic circuits are not functional in the DWORD MISR modes** - The DBI and ECC/SEV signals are treated as pure data signals. Their raw values are captured, compared, or sent without regard to their normal bus inversion or ECC functional meaning.

- It is required to enable Write DBIac and Read DBIac in MR0 in order to enable the I/O buffers on the DBI signals. A value of 0 is internally assumed for all DBI write data in case WDBI is disabled.

- Regardless whether meta data and severity reporting are is enabled in MR9 or not, setting MR7 DWORD Loopback Enable = 1 will enable the ECC/SEV signal's I/O buffers. Note that the SEV signals are bidirectional I/Os in loopback test mode only.

- The host may write DBI encoded or non-encoded data to the HBM4. In MISR mode or Register mode, the raw data received from the host well be directly captured (not DBI decoded) to the MISR register.

- For LFSR Compare mode to match, the host must send the LFSR generated raw data on all 10 signals of the byte without write DBI encoding. The HBM4 will not DBI decode the received data, and thus the host must send the raw LFSR data in order for LFSR Compare to match.

- For LFSR mode, the HBM4 will generate non-DBI encoded read data.

6.8.2 General Loopback Modes Features and Behavior (cont'd)

g) **DWORD read path parity traffic generation** - In DWORD LFSR mode (Read direction) and Read Register mode, the HBM4 parity logic is not active and the MR0 DQ Bus Read Parity settings has no effect. To generate traffic on the DWORD parity signal a copy of a nearby DQ signal is produced on the Parity signal. Logical signals DQ2, DQ34 are sent on the respective DWORD block parity DPAR0, DPAR1 signals, irrespective of any lane repairs. The parity signals are driven with the DQ data without any additional cycle delay - effectively with Parity Latency = 0. A suggested host-side implementation is to use signature register circuits for checking the validity of the received parity signal. When reading back the LFSR_COMPARE_STICKY error bits, the parity signal output is unspecified.

h) **AWORD and DWORD write parity checking** - In AWORD and DWORD Register mode, MISR mode, and LFSR Compare mode the HBM4 parity evaluation logic is active and outputs results on AERR after t_{PARAC} and DERR after t_{PARDQ}, respectively (if enabled in MR0, see Table 11). The MR1 Parity Latency setting (see Table 12) must be set to a vendor implementation-specific supported PL value, which may be interface speed specific. The HBM4 device will process write parity per the PL setting and protocol, including any required additional WDQS cycles. A suggested host-side implementation is to use signature register circuits for checking the correctness of the AERR and DERR signals. It is also suggested that the host generate data on the DWORD Parity signals in order to exercise these signal paths and logic.

i) **Preset state AAAAAAAAAAh and 2AAAAAAAAAh** - The Preset state for the DWORD MISR registers is AAAAAAAAAAh, which initializes the Rise bit for each signal to 1'b1 and the Fall bit to 1'b0. This is a useful state for producing an alternating 0/1/0/1/0/1/0/1 pattern on all 10 bits associated with a DWORD byte when put into DWORD read Register mode (burst length 8). This basic pattern may be used by the host for RDQS eye centering. READ commands from the DWORD_MISR are supported in Preset state. WRITE and READ commands to and from the DWORD_MISR do not change the DWORD_MISR content in this mode. The AWORD MISR register is also preset to the same 0/1 pattern (0x2AAAAAAAAAh for the 38-bit polynomial) for implementation consistency; although the AWORD cannot be enabled to drive this data pattern back to the host. Any non-zero initialization pattern is sufficient for all of the MISR modes; however, an initial pattern of all zeroes is a stuck-at-zero state for the DWORD LFSR mode. The Preset state may be overridden using the Write Register modes (see AWORD and DWORD Write Register Modes).

j) **DWORD MISR registers are writeable via IEEE 1500** - The normal intended method for writing the DWORD MISR registers are through the functional interface (see Test Method for DWORD Write MISR mode). The values of the DWORD MISR registers may also be written using the DWORD_MISR IEEE 1500 port instruction. This feature enables setting alternate seed values.

k) **DWORD read and write latencies must be set properly** - READ and WRITE commands are used to generate DWORD MISR modes traffic. Normal mode DWORD read and write protocol is followed using the latency settings, as supported by the operating frequency being used.

l) **DWORD Write preamble and post-amble clocks adhere to the normal protocol** - For DWORD write MISR modes (Register mode, MISR mode, and LFSR Compare mode), the host is expected to send WDQS preamble and postamble clocks, and the HBM4 samples the DWORD data, consistent with the write protocols defined in the section entitled Write Command (WR, WRA).

6.8.2 General Loopback Modes Features and Behavior (cont'd)

m) **DWORD Read preamble and post-amble clocks adhere to the normal protocol** - For DWORD Read Register mode, LFSR mode (Read direction), and when returning the LFSR_COMPARE_ STICKY bits, the HBM4 will produce RDQS preamble and postamble clocks, and send DWORD data, consistent with the read protocols defined in the section entitled Read Command (RD, RDA).

n) **AWORD MISR modes preamble clock filter** - In the AWORD MISR modes, the host is expected to stop CK toggling, enable the desired AWORD MISR mode, and then start sending CK toggles and AWORD data. To avoid timing impairment on the CK startup cycle, the HBM4 will treat the first received CK cycle as a preamble clock cycle and not process the data on the AWORD signals in MISR or Register mode, nor compare them in LFSR Compare mode. The MISR block will keep its state unchanged during filter cycle. The first clock cycle filter circuit is enabled by setting AWORD_MISR_CONFIG MODE = 2'b00 - Preset. The first data sampled by the HBM4 is on the second CK clock cycle. Only the very first CK clock cycle will be filtered - if the host were to stop and restart CK clocking while remaining in an AWORD MISR mode (without applying another Preset), the AWORD data will be sampled on the startup clock cycle, with possible CK edge timing impairment.

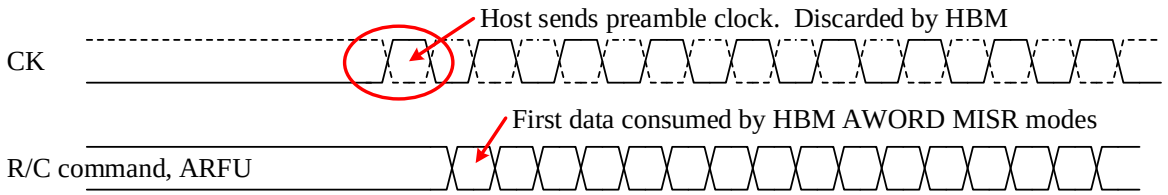


Figure 72 — AWORD MISR Modes Preamble Clock Filter Behavior

o) **Cycles processed in the MISR modes** - AWORD MISR modes rely on stopping CK clocks before and after the test sequence. All AWORD cycles sent to the HBM4 after the filtered preamble clock cycle are processed into the MISR (MISR mode and Register mode) or compared (LFSR Compare mode), including the last cycle before CK is stopped. An even number of AWORD cycles – not counting the filtered preamble cycle – provides that the test results is the same regardless of the specific AWORD MISR implementation. For DWORD MISR modes, all valid data cycles written to the HBM4 are processed into the MISR (MISR mode and Register mode) or compared (LFSR Compare mode) while the DWORD MISR modes are enabled, consistent with the DWORD write protocol and write latency setting. Data pin signal states during preamble and post-amble cycles are not processed into the MISR. For example, if 10 non-seamless Burst Length = 8 write operations are sent to the HBM4 in DWORD MISR mode a total of 80 data bit times (UI) will be processed into the MISR.

6.8.3 AWORD and DWORD Write MISR Modes

When the AWORD or DWORD MISR modes are active, the data on the AWORD or DWORD data signals is received based on the CK or WDQS clocks respectively, and compressed in the MISR circuits. The host is in complete control of the number of data cycles that are sent, and if successfully received by the HBM4 the values captured in the respective MISRs will be repeatable and deterministic. Figure 73 illustrates the behavior for DWORD MISR mode (Write direction).

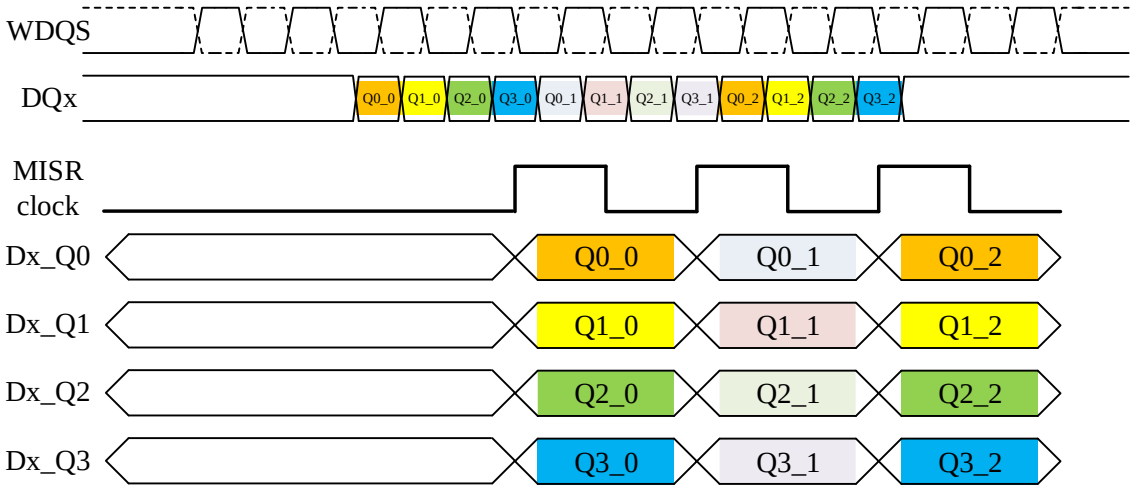


Figure 73 — DWORD Write MISR Modes Behavior

6.8.3.1 Test Method for AWORD (Write) MISR Mode

a) After the required HBM4 initialization, the host issues either precharge power-down or self refresh mode (R0 = LOW) and stops sending CK clocks to the HBM4 (CK_t = LOW, CK_c = HIGH).

b) Initialize the AWORD MISR by setting the AWORD_MISR_CONFIG Enable = 1'b1 - On and AWORD_MISR_CONFIG MODE = 2’b00 - Preset. The Preset operation also enables the preamble clock filter circuit.

c) Enable the AWORD MISR mode by setting AWORD_MISR_CONFIG Mode = 2'b11 - MISR mode.

d) The host sends three or more CK clock cycles and data on the AWORD signals. The first received CK clock cycle is discarded as a preamble clock by the HBM4 and shall be followed by an even number of clock cycles. The HBM4 clocks the received data into the AWORD MISR and evaluates parity, if enabled. The ending clock state applied by the host is CK_t = LOW, CK_c = HIGH.

e) The host reads the MISR content via the IEEE 1500 AWORD_MISR instruction.

6.8.3.2 Test Method for DWORD Write MISR Mode

a) Initialize the test sequence by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and presetting the MISR registers by setting the DWORD MISR Control = 3'b000 - Preset. The controller can load the DWORD MISR registers with an alternate seed value via the functional interface or IEEE 1500 (see AWORD and DWORD Write Register Modes and DWORD_MISR IEEE1500 port instruction).

b) Enable DWORD MISR mode by setting MR7 DWORD MISR Control = 3'b011 - MISR mode.

c) The host sends one or more DWORD write cycles following the write latency and burst length setting and following the normal write protocol. The HBM4 clocks the received data into the DWORD MISRs and evaluates parity, if enabled.

d) The host reads the MISR content via the IEEE 1500 DWORD_MISR instruction. The MISR content is also readable via the functional interface (see DWORD Read Register Mode).

6.8.4 AWORD and DWORD Write Register Modes

When the AWORD or DWORD Register modes are active, the data on the AWORD or DWORD data signals are received based on the CK or WDQS clocks respectively, and stored directly into the respective MISR registers without compression. Effectively the MISR register operates as a 2-bit storage register for AWORD and as a 4-bit storage register for DWORD. On rising CK or WDQS edges the signal states on the AWORD or DWORD bus respectively are stored in the Rising bits within the MISR registers, and on falling CK or WDQS edges the bus signal states are stored in the Falling bits within the MISR registers. If the host sends multiple DDR cycles to the HBM4, the MISRs will contain the last 2-bit per AWORD MISR cycle and 4-bit per DWORD MISR cycle, if successfully received by the HBM4.

The Register modes are intended for basic, quick link testing and training, and for initializing the DWORD MISR seed values.

6.8.4.1 Test Method for AWORD (Write) Register Mode

a) After the required HBM4 initialization, the host issues either precharge power-down or self refresh mode (R0 = LOW) and stops sending CK clocks to the HBM4 (CK_t = LOW, CK_c = HIGH).
b) Initialize the AWORD MISR by setting the AWORD_MISR_CONFIG Enable = 1'b1 - On and AWORD_MISR_CONFIG MODE = 2’b00 - Preset. The Preset operation enables the preamble clock filter circuit.
c) Enable the AWORD Register mode by setting AWORD_MISR_CONFIG MODE = 2’b10 – Register mode.
d) The host sends two or more CK clock cycles and data on the AWORD signals. The first received CK clock cycle is discarded as a preamble clock by the HBM4. The HBM4 clocks the raw received data into the AWORD MISR register without MISR compression and evaluates parity, if enabled. The ending clock state applied by the host is CK_t = LOW, CK_c = HIGH. The last clocked DDR cycle data is retained in the AWORD MISR register.
e) The host reads the MISR content via the IEEE 1500 AWORD_MISR instruction.

6.8.4.1 Test Method for AWORD (Write) Register Mode (cont'd)

Note that the AWORD write register mode cannot practically be used to apply an alternate seed value into the AWORD MISR register. In section 6.8.4.1 Test Method for AWORD (Write) Register Mode step d, the preamble clock filter circuit is exercised and cleared. At this point while it is allowed for the host to then stop sending AWORD cycles, set the AWORD_MISR_CONFIG MODE to MISR mode or LFSR Compare mode, and then send additional AWORD cycles, there may be timing impairment for the beginning of the second set of AWORD cycles.

The preamble clock filter circuit cannot be re-enabled for these additional AWORD cycles without applying the AWORD MISR Preset function, which would also overwrite the alternate seed value applied by the AWORD write register operation. There is no expected application value for using an alternate MISR seed value for the AWORD MISR functions since the AWORD bus is receive-only.

6.8.4.2 Test Method for DWORD Write Register Mode

a) Enable DWORD Register mode by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and DWORD MISR Control = 3'b010 - Register mode. A Preset is not required prior to using Register mode.
b) The host sends one or more DWORD write cycles following the write latency and burst length setting and following the normal write protocol. The HBM4 clocks the raw received data into the DWORD MISR registers without MISR compression and evaluates parity, if enabled. The last clocked DDR cycle data is retained in the DWORD MISR registers.
c) The host reads the MISR content via the IEEE 1500 DWORD_MISR instruction. The MISR content is also readable via the functional interface (see DWORD Read Register Mode).

6.8.5 DWORD Read Register Mode

The content of various DWORD MISR mode related registers may be read over the functional interface, assuming that the read path with the host is properly trained (or used for read path training). The MR7 DWORD Read Mux Control bit field is used to select the data source. The host issues read commands and the HBM4 responds following the read command protocol (such as read latency and burst length) and timing (such as pre and post-amble clocks) per Read Command (RD, RDA).

Intended uses for the various read data sources include the following:

- Reading the sticky error bits after an LFSR Compare mode test sequence (DWORD Read Mux Control = 1'b1 - Return LFSR_COMPARE_STICKY) - Sticky error data is a single data bit per signal and is output as static values on the interface for the full read burst length.

- NOTE: When using the LFSR mode (see DWORD Read LFSR Mode) set the DWORD Read Mux Control = 1'b0 - Return data from DWORD MISR registers.

- Reading a basic clock pattern on all or select signals for DWORD read link training (DWORD Read Mux Control = 1'b0 - Return data from DWORD MISR registers) - Which signals toggle may be set with the Preset mode or a DWORD Register write (see AWORD and DWORD Write Register Modes).

- Reading the MISR registers final values at the end of a MISR mode test sequence (DWORD Read Mux Control = 1'b0 - Return data from DWORD MISR registers) - The results of a MISR mode test sequence may be read back on the functional interface, or via the IEEE 1500 port DWORD_MISR instruction. The MISR content is sent on UI 0 - 3 and then repeated on UI 4 – 7, or all data(UI 0 to 7) of the most recent Write depending on vendor’s implementation (see Table 19 DWORD MISR Read and Write Operations in Loopback Test Mode).

6.8.5.1 Test Method for DWORD Read Register Mode

a) Enable the test mode and select the desired read-back register by setting MR7 DWORD Loopback Enable = 1'b1 - Enable, DWORD MISR Control = 3'b010 - Register mode, and DWORD Read Mux Control = 0.
b) The host sends one or more DWORD read commands. The HBM4 responds following the read latency and burst length setting and following the normal read protocol.

6.8.6 DWORD LFSR Mode (Read direction)

When in DWORD LFSR mode (Read direction), the HBM4 generates DWORD data from the LFSR in response to read commands issued by the host. LFSR data is generated consistent with only the valid UIs of the read protocol. Read Preamble and post-amble RDQS clocks are generated consistent with the read protocol. The first data cycle generated will be the LFSR initial state, based on Preset or an alternate seed value if loaded. Figure 74 illustrates the behavior for DWORD LFSR mode (Read direction).

NOTE: There is no AWORD LFSR mode since the AWORD bus cannot source data to the host.

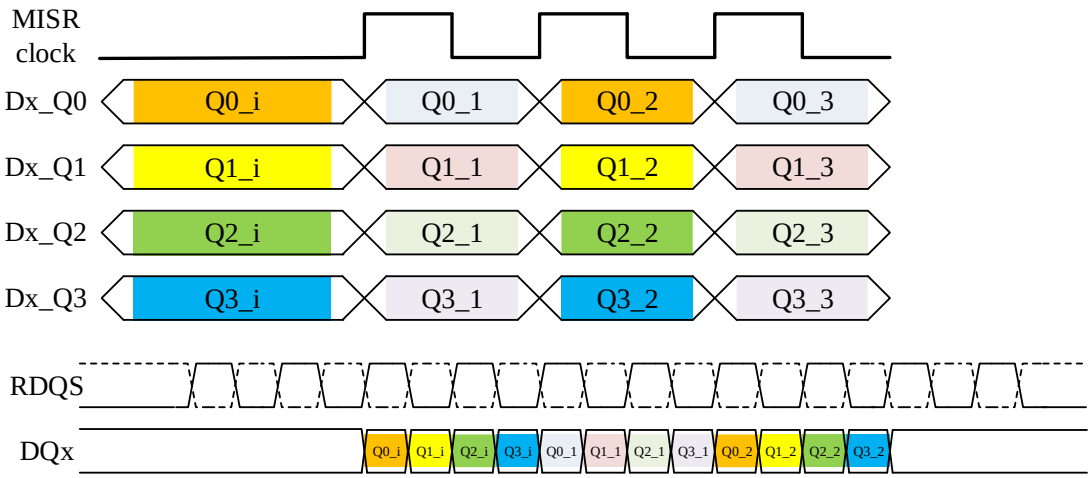


Figure 74 — DWORD Read LFSR Modes Behavior

6.8.6.1 Test Method for DWORD LFSR Mode (Read direction)

a) Initialize the test sequence by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and presetting the MISR registers by setting the DWORD MISR Control = 3'b000 - Preset. The controller can load the DWORD MISR registers with an alternate seed value via the functional interface or IEEE 1500 (see AWORD and DWORD Write Register Modes and DWORD_MISR IEEE 1500 port instruction).

b) Enable DWORD LFSR mode by setting MR7 DWORD MISR Control = 3'b001 - LFSR mode and DWORD Read Mux Control = 1'b0 - Return data from DWORD MISR registers.

c) The host sends one or more DWORD read commands. The HBM4 responds following the read latency and burst length setting and following the normal read protocol, with data produced from the LFSR. A suggested host-side implementation is to use signature register circuits for checking the validity of the received data.

6.8.7 AWORD and DWORD Write LFSR Compare Modes

The LFSR Compare modes enable direct identification of failing signal connections between the Host and HBM4. It is assumed that the Host implements LFSR data generators that match the lengths and polynomials of the HBM4 LFSRs, and that the Host and HBM4 LFSRs start and run in synch. The LFSRs generate data on each signal, and the compare circuitry checks for matching data for each data unit interval (UI). Any mismatch between the data received at the HBM4 inputs (based on the respective CK or WDQS clocking) and the data predicted by the HBM4 LFSR will set the sticky error bit for the respective signals. The first data cycle expected from the host and compared by the HBM4 will be the LFSR initial state, based on Preset or an alternate seed value if loaded.

Once a miscompare is found on a signal, its sticky error bit is set (1'b1) for the remainder of the test sequence. The sticky error bits may be read via the IEEE 1500 port READ_LFSR_COMPARE_STICKY instruction or via the functional interface (see DWORD Read Register Mode). AWORD sticky error bits are only readable via the IEEE 1500 port. The sticky error bits are reset (1'b0) via the MR7 DWORD MISR Control = 3'b000 - Preset, or IEEE 1500 AWORD_MISR_CONFIG MODE = 2'b00 - Preset.

Figure 75 illustrates the system-level configuration for LFSR Compare mode.

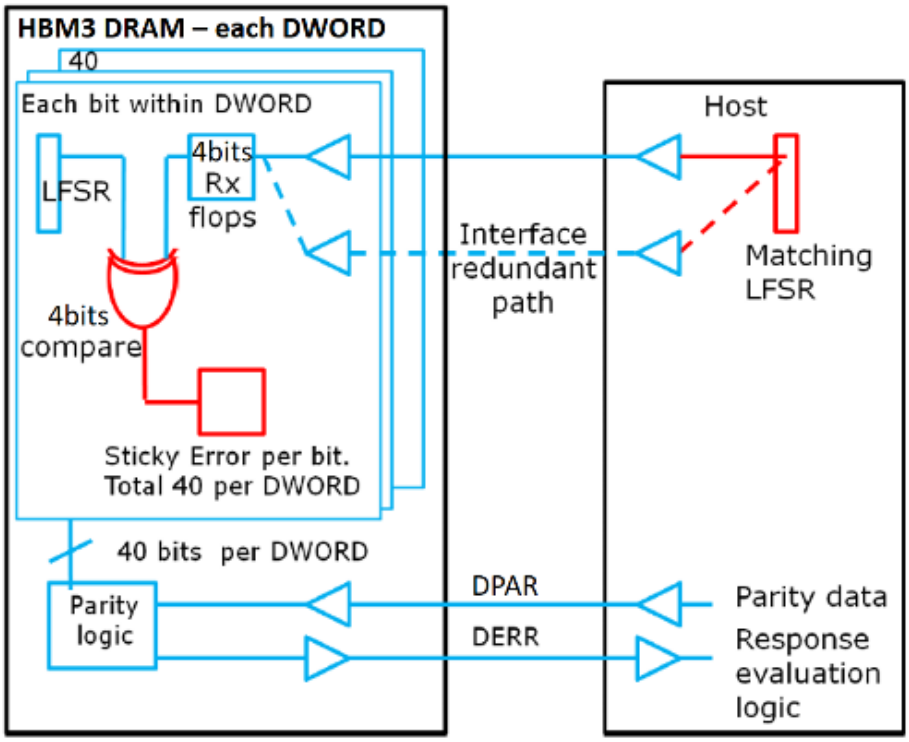


Figure 75 — LFSR Compare Mode Block Diagram

Note that data produced on the DWORD Parity signals from the host to the HBM4 is an implementation suggestion for exercising the parity signal paths and HBM4 input timing and logic. The host-side implementation for parity signal generation is not specified. This figure also illustrates that a host-driven logical signal is compared with the matching logical signal data by the HBM4 compare circuit, regardless of any active lane repairs which may shift the physical signal routing. The AWORD LFSR Compare circuit matches the DWORD circuit except for the non-existent Parity signals.

6.8.7.1 Test method for AWORD (Write) LFSR Compare Mode

a) After the required HBM4 initialization, the host issues either precharge power-down or self refresh mode (R0 = LOW) and stops sending CK clocks to the HBM4 (CK_t = LOW, CK_c = HIGH).

b) Initialize the AWORD MISR (LFSR) register by setting the AWORD_MISR_CONFIG Enable = 1'b1 - On and AWORD_MISR_CONFIG MODE = 2'b00 - Preset. The Preset operation also clears the AWORD per-signal sticky error bits and enables the preamble clock filter circuit. The host-side LFSR data generator should also be initialized to the same value.

c) Enable the AWORD LFSR Compare mode by setting AWORD_MISR_CONFIG MODE = 2'b01 - LFSR Compare mode.

d) The host sends two or more CK clock cycles with LFSR-generated data on the AWORD signals. The first received CK clock cycle is discarded as a preamble clock by the HBM4. The HBM4 LFSR predicts expected AWORD data per cycle from the host, based on matching LFSR polynomials and starting seeds in the host and HBM4. Any mismatches set sticky error for the respective signal. Parity is evaluated, if enabled. The ending clock state applied by the host is CK_t = LOW, CK_c = HIGH.

e) The host reads the Sticky error bits to determine which signals failed. The bits are readable via the IEEE 1500 port READ_LFSR_COMPARE_STICKY instruction.

6.8.7.2 Test Method for DWORD Write LFSR Compare mode

a) Initialize the DWORD LFSR (MISR) registers by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and DWORD MISR Control = 3'b000 - Preset. The Preset operation also clears the DWORD per-signal sticky error bits. The controller can load the DWORD MISR registers with an alternate seed value via the functional interface or IEEE 1500 (see AWORD and DWORD Write Register Modes and DWORD_MISR IEEE1500 port instruction). The host-side LFSR data generator should also be preset/initialized to the same value.

b) Enable DWORD LFSR Compare mode by setting MR7 DWORD MISR Control = 3'b100 – LFSR Compare mode.

c) The host sends one or more DWORD write cycles with LFSR-generated data on the DWORD signals following the write latency and burst length setting and following the normal write protocol. The HBM4 LFSRs predict expected DWORD data per cycle from the host, based on matching LFSR polynomials and starting seeds in the host and HBM4. Any mismatches set sticky error for the respective signal. Parity is evaluated, if enabled.

d) The host reads the sticky error bits to determine which signals failed. The bits are readable via the IEEE 1500 port READ_LFSR_COMPARE_STICKY instruction. The sticky error bits are also readable via the functional interface (see DWORD Read Register Mode).

6.9 On-die DRAM ECC

6.9.1 ECC Overview

The HBM4 device uses a symbol-based on-die ECC, read/write meta-data (MD) bits, an error scrubbing mechanism, an error transparency protocol, interface transmission parity, and fault isolation limits to achieve a high level of system RAS.

HBM4 ECC features:

- Minimum 304b codeword
  - 256b+16b user data access size
  - Symbol-based on-die ECC
  - Symbol size is implementation specific

- On-die ECC real-time transparency
  - Two pins per PC transmit error severity
  - SBE signaled only after SBE threshold exceeded

- Automated on-die error scrubbing mechanism
  - Auto-ECS during REFab operation has MR for enable/disable
  - Auto-ECS during SRF has MR for enable/disable
  - MR bit to enable correction of CEm during ECS
  - Errors are only logged during ECS

- Single bit READ and WRITE data interface parity
  - DQ, DBI, and ECC bits included in parity calculation
  - SEV transparency bits not included in parity calculation

An overview of an example HBM4 on-die ECC engine is shown in Figure 76.

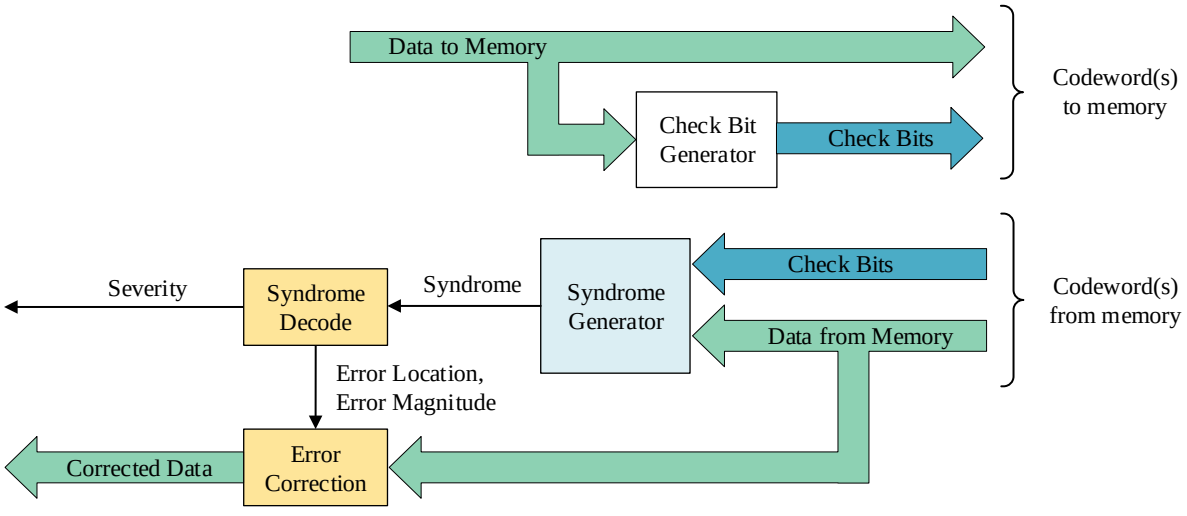


Figure 76 — On-die ECC Overview Diagram Example

6.9.2 HBM4 On-die ECC Requirements

On-die ECC Engine:

HBM4 devices shall implement on-die symbol-based ECC.

HBM4 on-die ECC has a codeword size dependent on symbol size error correction capability. The data-word and example check-bits of the codeword are as follows:

- Data-word: 272b (256b data per PC + 16b meta data per PC)
- On-die ECC check-bits: Implementation specific (e.g. 32b assuming 16b single symbol correction)

The 272b user data consists of 256b transmitted over 32 DQ pins x BL8 and 16b transmitted over 2 ECC pins x BL8.

On reads the DRAM corrects all errors that are less than or equal to a single symbol size and within the symbol boundary before returning the 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 check bits and writes the data and check bits to the array.

In the case of interface MD bits being disabled via MR9 OP0, the DRAM may assume any value for the 16b of the ECC data-word corresponding to the MD bits. The DRAM can only guarantee valid array MD bits if written while interface MD function is enabled. The ECC engine treatment of the MD bits is not affected by the disabling of the interface MD setting.

The specific ECC H-matrix used, the symbol size, and the number of codewords is implementation specific.

6.9.3 DRAM Fault Isolation Requirements

Fault isolation is the management of errors caused by various faults to be isolated within certain boundaries regardless of the od-ECC operation.

The fault isolation boundaries will be chosen in accordance with the ECC symbol size to maximize the correction capability of multi-bit faults. The design must guarantee that the most common multi-bit fault modes will create errors constrained to a correctable symbol-size or fewer bits.

6.9.4 Error Check and Scrub (ECS)

The HBM4 device will implement an Auto ECS function. Auto ECS will use on-die ECC and operate in the background during REFab and SRE periods. The ECS mode allows the DRAM to internally read, detect errors, correct errors, and write back corrected data bits to the array (scrub errors) while providing transparency. 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.

6.9.4 Error Check and Scrub (ECS) (cont'd)

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 single-bit error, the error will be corrected, and code-word(s) will be written back to DRAM.
3. If the ECC engine detects a correctable multi-bit error, the error will be corrected, and code-word(s) will be written back to DRAM. CEm during ECS can be enabled/disabled by MR9 OP6.
4. 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.
5. If the ECC engine detects no error, the DRAM may choose to write the resultant code-word(s) back to DRAM or not.

ECS related MR control:

Table 63 — ECS Modes

ECS Mode	Mode Register Value (Default All Disabled)
Auto ECS via REFab	MR9 OP4, 1 = Enabled, 0 = Disabled
Auto ECS during Self Refresh	MR9 OP5, 1 = Enabled, 0 = Disabled
CEm during ECS	MR9 OP6, 1 = Enabled, 0 = Disabled
ECS Error Type and Address Reset	MR9 OP7, 1 = Reset (Self Clearing), 0 = Maintain
ECS Error Log Reset with Log Read-out	MR8 OP2 1 = Enabled, 0 = Disabled
NOTE 1 REFab used for ECS will count toward refresh credit. NOTE 2 When ECS during REFab is enabled, the host must issue REFab commands at an average rate of t_{ECSint}.	

The internal Error Check and Scrub Log status for error type and address is initialized either by a device RESET or by manually writing a "1" to MR9 OP7.

ECS modes MR9 OP[6:4] defined in Table 63 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.

The DRAM can only guarantee valid ECS operations if array bits are written to prior to executing ECS operations, thus enabling DRAM to calculate the proper parity bits.

6.9.4 Error Check and Scrub (ECS) (cont'd)

ECS related timing parameters:

The ECS operation timing is shown in Figure 77.

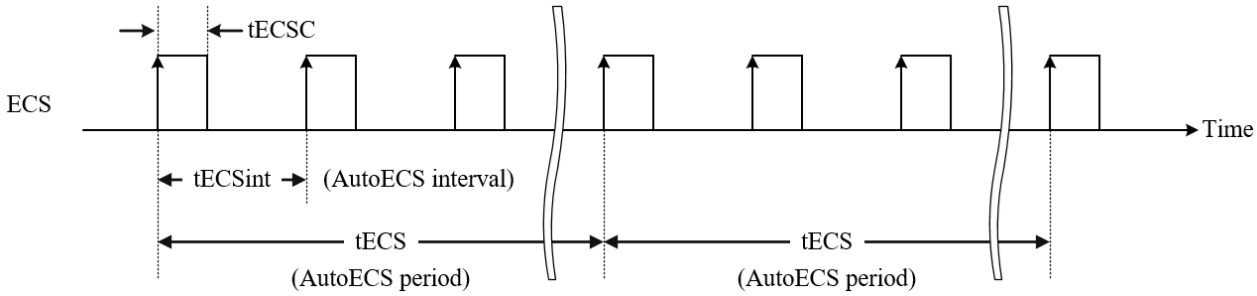


Figure 77 — ECS operation timing

- t_{ECSC}: Max time for HBM4 to complete ECS operation
- t_{ECSint}: Average ECS interval to cover all codewords in a specified period of t_{ECS} (e.g. 24h)
- t_{ECS}: Period of time to complete ECS on all codeword
- ERRTH: Vendor specific filter threshold of ERRCNT used for transparency. No CEs will be logged or transmitted on SEV pins until ERRCNT > ERRTH

In order to complete a full Error Check and Scrub within the recommended t_{ECS} (e.g. 24 hours), the average periodic interval of ECS operations (t_{ECSint}) is 86,400 seconds divided by the total number of codewords as described in Table 64. The number if ECS operations is configuration dependent.

Table 64 — t_{ECSint} per Stack (ECS independent of SID)

Configuration	24 Gb x 4/ 8/ 12/ 16	32 Gb x 4/ 8/ 12/ 16
GB per device	12/ 24/ 36/ 48 GB	16/ 32/ 48/ 64 GB
Gb per PC	3 Gb	4 Gb
304b code-words per PC per SID	2^23*1.5	2^24
t_{ECSint} [ms] per PC	6.866	5.150

In order for the HBM4 to perform ECS operations when in ECS Mode, the host needs to issue periodic REFab or SRE commands. The maximum average spacing between REFab or SRE commands for the DRAM to complete the automatic scrub within the recommended t_{ECS}(e.g. 24 hours) is t_{ECSint}. Meeting this REFab requirement allows the DRAM to perform the ECS operations without placing additional restrictions on refresh mode usage, i.e., all bank/per-bank refresh or normal mode refresh, while in ECS mode. REFab commands issued in excess of required by the DRAM for ECS operations (one per t_{ECSint}) may be used by the DRAM for normal refresh operation.

6.9.4 Error Check and Scrub (ECS) (cont'd)

ECS related logging: The registers are allocated per PC and SID accessible via the IEEE 1500 interface.

1. When the on-die ECC detects an error, the DRAM address of the error must be logged in the form of Bank, Row, Column, Error Type Severity
2. The error is logged within t_{ECSC} and accessible via IEEE 1500

The priority of error logging is defined in Table 65.

Table 65 — Error Overwrite Priority Rules to Handle Multiple Error Logging

Previous Error	Current Error			
	NE	**CEs**	**CEm**	**UE**
NE (No error)	None	Update	Update	Update
CEs (Corrected single-bit error)	Maintain	Update	Update	Update
CEm (Corrected multi-bit error)	Maintain	Maintain	Update	Update
UE (Uncorrectable error)	Maintain	Maintain	Maintain	Update
NOTE 1 Logging of newest error may be lost in case of a simultaneous reset and new ECS error NOTE 2 In the case of MR8 OP2 = 1, reset of ECS error log can only be guaranteed when captured WDR of ECS error log is valid				

Reset of ECS error log:

A reset of the ECS error log clears all VALID bits of the ECS_ERROR_LOG WDR to 0b and the error priority log to "NE" (no error). There are three independent methods for clearing the error log:

- The host may issue device RESET
- The host may issue a log reset according to MR9 OP7
- The host may configure MR8 OP2 ECS error log auto-reset, in which case the error log will be reset upon read of the error log

Error counting:

The number of CEs are counted during ECS in order to control whether the severity information of CEs is conveyed on the SEV pins during READ operations. Error counting assumes one codeword covering each access.

- ERRCNT == number of error events accumulated during ECS
- ERRCNT is independently maintained per PC and SID
- CEs count as one event toward ERRCNT
- CEm and UE do not count toward ERRCNT
- If more than one codeword is used, a CEs in both codewords counts as a CEm
- ERRCNT will be incremented a maximum of one for any codeword size

Reset of ERRCNT:

Automatic reset internally by HBM4 after each t_{ECS}

6.9.5 On-die ECC Transparency Protocol

An HBM4 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 66.

Table 66 — Transparency Attributes and Their Access/Control Mechanism

Attribute	Operation	Transparency Mechanism
Real-time severity metadata	RD/RDA	Two SEV pins per PC
Logging address and severity of an error	ECS	IEEE1500 register

Severity Metadata: The severity of an error denotes the outcome of the on-die ECC processing over a codeword(s) during a READ operation. The severity information is conveyed on the SEV pins together with the data transfer on the DQ pins. Severity transmission will use the encoding shown in the Table 67 for each BL8 transaction.

Table 67 — Severity Encodings on the SEV pins

Severity	Pin	Burst Position							
		0	1	2	3	4	5	6	7
NE	SEV[0]	0	0	0	0	0	0	0	0
	SEV[1]	0	0	0	0	0	0	0	0
CEs	SEV[0]	0	0	0	0	1	1	1	1
	SEV[1]	0	0	0	0	0	0	0	0
CEm	SEV[0]	0	0	0	0	1	1	1	1
	SEV[1]	0	0	0	0	1	1	1	1
UE	SEV[0]	0	0	0	0	0	0	0	0
	SEV[1]	0	0	0	0	1	1	1	1

Severity Metadata Signaling Control: The HBM4 device includes a mode register MR9 OP1 to enable or disable the severity metadata signaling by the HBM4 device.

Table 68 — Severity Transmission on READ

On-die ECC Severity	NE	CEs	CEm	UE
Severity on SEV[1:0]	NE	NE if ERRCNT <= ERRTH CEs if previous or current ERRCNT > ERRTH	CEm	UE

6.9.5 On-die ECC Transparency Protocol (cont'd)

The CEs output enable timing for SEV is shown in Figure 78.

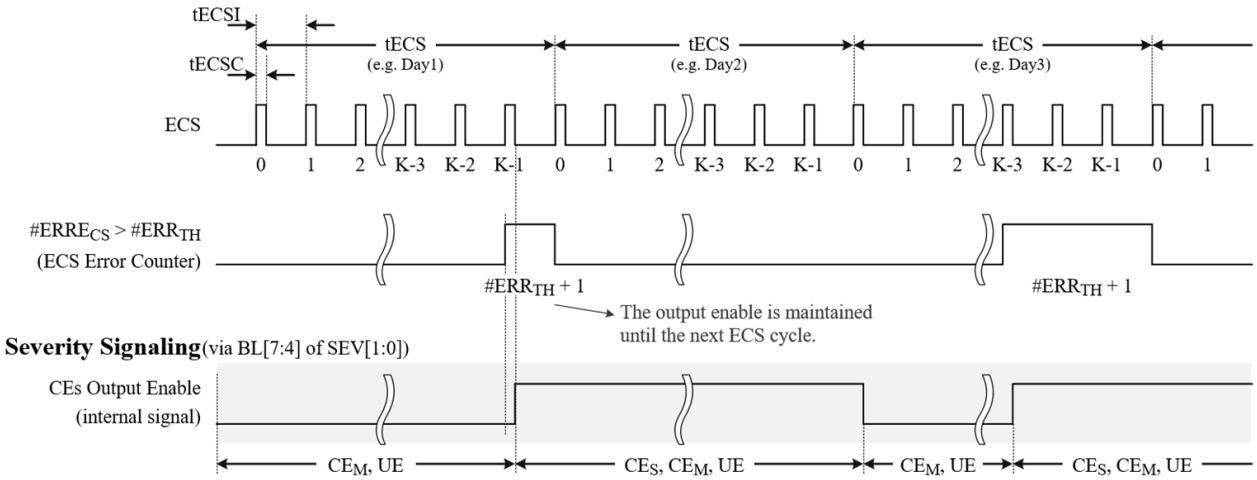


Figure 78 — ECS CEs Output Enable Timing for SEV Signaling

6.9.6 ECC Engine Test Mode

HBM4 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.

Table 69 — ECC Engine Test Modes

Selection by MRS	ECC Engine Test Mode
ECC Engine Test Mode (MR9 OP2)	0 – Normal Operation (Default) 1 – ECC Engine Test Mode
Error Vector Patterns (MR9 OP3)	0 – CW0 (Codeword0) Data '1' means error bit and Data '0' means non-error bit 1 – CW1 (Codeword1) Data '0' means error bit and Data '1' means non-error bit

While in the ECC engine test mode in Table 69,

1. WR will function as an error injection command, Write DQ data is error injection pattern (CW0 or CW1 by MR9 OP3)
2. RD will function as an outcome output command, Read DQ/ECC/SEV data is the outcome of ECC engine test

6.9.6 ECC Engine Test Mode (cont'd)

The following sequence must be satisfied to perform a functional On-die ECC engine test mode of HBM4 DRAM. See Figure 79 and Table 70.

1. The HBM4 device registers Mode Register Set command (MRS) by MR9 OP[3:2] for the entry of On-die ECC engine test mode in Table 69. Error severity reporting must be enabled via the SEVR bit in MR9 OP1. The MD bit in MR9 OP0 must be set according to the user's desire to include the ECC signals in this test or not.

2. As an example in Table 70 — Example of Error Vectors 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.

3. To check the result of engine test, read the output after t_{WTR}.
   A. The DQs will show the correction data as ALL "0" when the DATA is CEs or CEm in the case of CW0. Also, the output will show the values as written data when the DATA is UE case.
   B. The BL[7:4] of SEV[1:0] pins will indicate NE, CEs, CEm and UE. CEs severity information can be real-time signaling via SEV[1:0]. During ECC engine test, #ERRTH value is ignored.

4. Repeat the 2, 3, 4 sequence and the operation for the engine test after t_{RTW}.
   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/ECC and DATA[271:0] 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 HBM4 is in the ECC Engine Test Mode, it does not guarantee data retention and the only allowed commands are CNOP, WR, RD and MRS to disable this test mode.

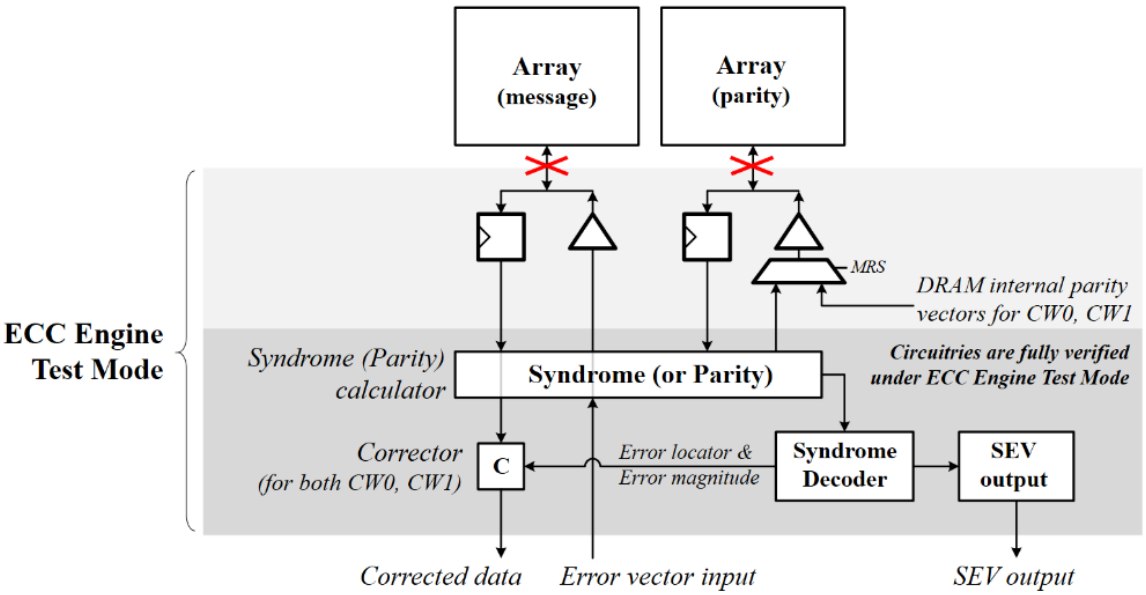


Figure 79 — The Block Diagram of On-die ECC Engine and Path for ECC Engine Test Mode

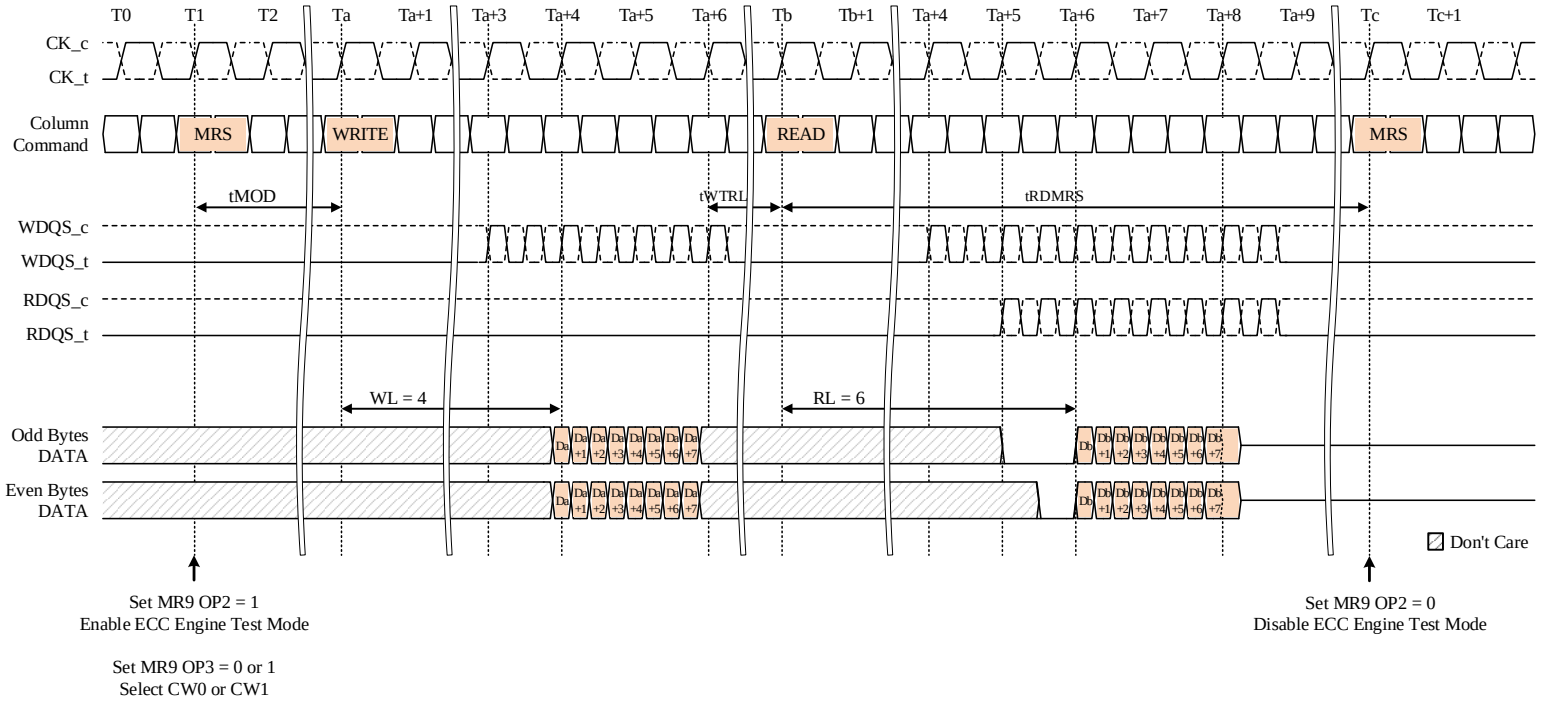
6.9.6 ECC Engine Test Mode (cont'd)

Table 70 — Example of Error Vectors and Corresponding Severity

Severity	Error Vector Pattern (MR9 OP3)	Error Vector Input[271:0] (Write data)	Error Vector Output[271:0] (Read data)	Severity (SEV[1:0])	Note
NE	CW0	0000…00000000	0000…00000000	NE	1
	CW1	1111…11111111	1111…11111111		2
CEs	CW0	1000…00000000 … 0000…00000001	0000…00000000 … 0000…00000000	CEs	1
	CW1	0111…11111111 … 1111…11111110	1111…11111111 … 1111…11111111		2
CEm	CW0	0000…00001111 … 1111…00000000	0000…00000000 … 0000…00000000	CEm	1, 3
	CW1	1111…11110000 … 0000…11111111	1111…11111111 … 1111…11111111		2, 3
UE	CW0 or 1	None of the above	Not specified	UE	

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.
NOTE 3 CEm is limited to a symbol.

6.9.6 ECC Engine Test Mode (cont'd)



NOTE 1 WRITE and READ address must be the same for ECC Engine Test Mode.
NOTE 2 WRITE and READ commands don't require a preceding ACT command for ECC Engine Test Mode.
NOTE 3 No other commands are allowed except CNOP and MRS to disable ECC Engine Test mode.
NOTE 4 WL = 4 and RL = 6 are shown as an example.
NOTE 5 DATA = DQ[31:0], DBI[3:0], ECC[1:0], SEV[1:0] for PC0 and DATA = DQ[63:32], DBI[7:4], ECC[3:2], SEV[3:2] for PC1.
WDQS_t/_c = WDQS0_t/_c for PC0 and WDQS1_t/_c for PC1.
RDQS_t/_c = RDQS0_t/_c for PC0 and RDQS1_t/_c for PC1.
NOTE 6 Da, ..., Da+7 = data-in for WRITE command. Db, ..., Db+7 = data-out for READ command.
NOTE 7 $t_{WDQS2DQ\_O}$, t_{DQSS} = 0 and nominal t_{QW} are shown for illustration purposes.
NOTE 8 t_{WTR} should be t_{WTRL} by both WRITE and READ access banks in the same bank group for ECC Engine Test Mode.
NOTE 9 WDBI and RDBI could be on or off. WDBI is controlled with MR0 OP1 and RDBI is controlled with MR0 OP0.
NOTE 10 WDBI and RDBI off are recommend for output of ECC Engine Test Mode. (see Table 70 — Example of Error Vectors and Corresponding Severity)
NOTE 11 SEVR on is mandatory to verify on-die ECC transparency.
NOTE 12 It is recommended that the MD bit is evaluated and the ECC signals are not included with MD on.
NOTE 13 The WRITE and READ commands do not require a preceding ACT command in this test mode.

Figure 80 — Timing Diagram of ECC Engine Test Mode

6.10 WOSC

6.10.1 WDQS Interval Oscillator

As voltage and temperature change on the HBM4 DRAM, the WDQS clock tree delay will shift and may require re-training. The HBM4 DRAM includes an internal WDQS clock-tree oscillators 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 WDQS Interval Oscillator ("oscillator") will provide the controller with important information regarding the need to re-train, and the magnitude of potential error. The oscillator is not associated with any channel and operates fully independent of any channel's operating frequency or state (e.g. bank active, bank idle, power-down or self refresh). Also, no CK, WDQS or WRCK clock is required while the oscillator is counting. The oscillator is disabled by default upon power-up.

The IEEE1500 instructions WOSC_RUN and WOSC_COUNT are associated with the oscillator. Setting the WOSC_START_STOP bit in the WOSC_RUN Wrapper Data Register to 1 will start an internal ring oscillator that counts the number of times a signal propagates through a copy of the WDQS clock tree. The oscillator is stopped by setting the WOSC_START_STOP bit back to 0. The maximum count is 2^{24} - 1, and the longest run time for the oscillator to not overflow the counter can be calculated as follows:

$$\textbf{Longest Run Time Interval} = 2^{24} * t_{RX\_DQS2DQ}\textbf{(min)}$$

The validity of the clock count is indicated by the WOSC_COUNT_VALID bit in the WOSC_COUNT Wrapper Data 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 WOSC_COUNT_VALID bit is set to 1 to indicate a valid count, and the result of the counter is stored in the WOSC_COUNT_VALUE field of the WOSC_COUNT WDR. The WOSC_COUNT_VALID bit will remain 0 (invalid) if the counter overflows (2^{24} or more cycles) or if the oscillator is interrupted by pulling RESET_n to LOW. On the other hand, pulling WRST_n to LOW does not impact the oscillator's operation. After the oscillator stops the host may issue the WOSC_COUNT instruction to read out the count.

The controller may adjust the accuracy of the result by running the oscillator for shorter (less accurate) or longer (more accurate) duration. The accuracy of the result for a given temperature and voltage is determined by the following equation:

$$\textbf{WDQS Oscillator Granularity Error} = \frac{\textbf{2 * (WDQS delay)}}{\textbf{Run Time}}$$

Where:

- Run Time = total time between the oscillator starting and automatically stopping
- WDQS delay = the value of the WDQS clock tree delay [t_{RX_DQS2DQ}(min/max)]

Additional matching error must be included, which is the difference between WDQS training circuit and the actual WDQS clock tree across voltage and temperature. The matching error is vendor specific.

Therefore, the total accuracy of the WDQS Oscillator counter is given by:

$$\textbf{WDQS Oscillator Accuracy} = 1 - \textbf{Granularity Error} - \textbf{Matching Error}$$

6.10.1 WDQS Interval Oscillator (cont'd)

Example: If the total time between start and stop is 100 ns, and the maximum WDQS clock tree delay is 400 ps [t_{RX_DQS2DQ}(max)], then the WDQS Oscillator Granularity Error is:

$$\textbf{WDQS Oscillator Granularity Error} = \frac{\textbf{2 * (0.4 ns)}}{\textbf{100 ns}} = \textbf{0.8\%}$$

This equates to a granularity timing error of 3.2ps.

Assuming a circuit Matching Error of 5.5ps across voltage and temperature, then the accuracy is:

$$\textbf{WDQS Oscillator Accuracy} = \textbf{1 -} \frac{\textbf{3.2 + 5.5}}{\textbf{400}} = \textbf{97.8\%}$$

Example: Running the WDQS Oscillator for a longer period improves the accuracy. If the total time between start and stop is 250ns, and the maximum WDQS clock tree delay is 400ps [t_{RX_DQS2DQ}(max)], then the WDQS Oscillator Granularity Error is:

$$\textbf{WDQS Oscillator Granularity Error} = \frac{\textbf{2 * (0.4 ns)}}{\textbf{250 ns}} = \textbf{0.32\%}$$

This equates to a granularity timing error or 1.28ps.

Assuming a circuit Matching Error of 5.5ps across voltage and temperature, then the accuracy is:

$$\textbf{WDQS Oscillator Accuracy} = \textbf{1 -} \frac{\textbf{1.28 + 5.5}}{\textbf{400}} = \textbf{98.3\%}$$

6.10.1 WDQS Interval Oscillator (cont'd)

The WDQS Interval Oscillator matching error is defined as the difference between the WDQS training circuit (interval oscillator) and the actual WDQS clock tree across voltage and temperature.

Parameters:

- t_{RX_DQS2DQ}: Actual WDQS clock tree delay
- $t_{WDQSosc}$: Training circuit (interval oscillator) delay
- $WOSC_{Offset(V)}$: Average delay difference over voltage
- $WOSC_{Offset(T)}$: Average delay difference over temp
- $WOSC_{Match(V)}$: WDQS oscillator matching error over voltage
- $WOSC_{Match(T)}$: WDQS oscillator matching error over temp

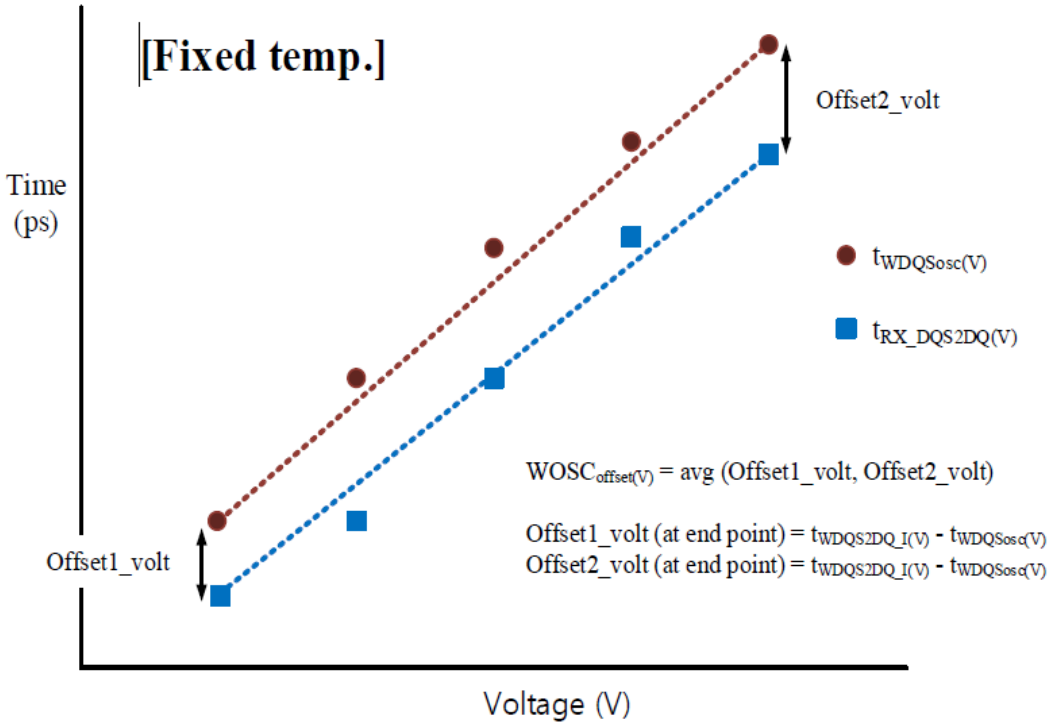


Figure 81 — Oscillator offset ($WOSC_{offset(V)}$)

$WOSC_{Match(V)}$:

$$\mathbf{WOSC_{Match(V)} = [t_{RX\_DQS2DQ(V)} - t_{WDQSosc(V)} - WOSC_{offset(V)}]}$$

$t_{DQSosc(V)}$:

$$\mathbf{t_{WDQSosc(V)} = \frac{Runtime}{2 * Count}}$$

6.10.1 WDQS Interval Oscillator (cont'd)

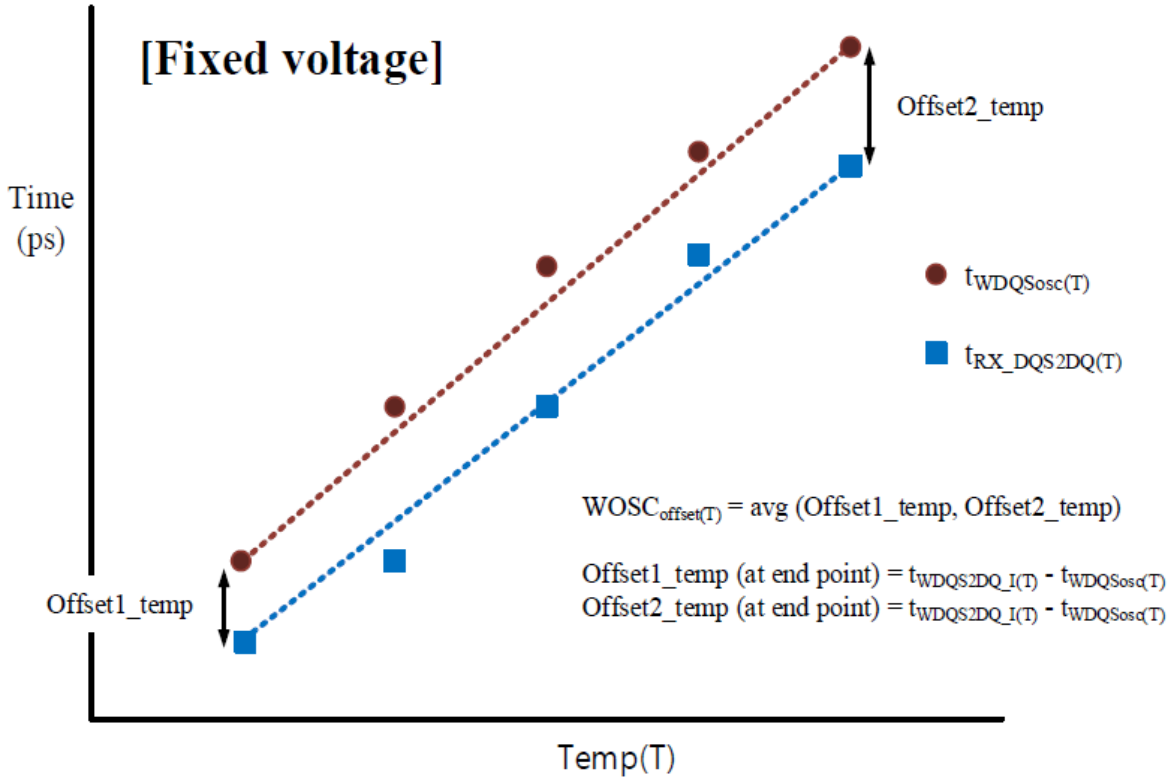


Figure 82 — Oscillator offset ($WOSC_{offset(T)}$)

$WOSC_{Match(T)}$:

$$\mathbf{WOSC_{Match(T)} = [t_{RX\_DQS2DQ(T)} - t_{WDQSosc(T)} - WOSC_{offset(T)}]}$$

$t_{WDQSosc(T)}$:

$$\mathbf{t_{WDQSosc(T)} = \frac{Runtime}{2 * Count}}$$

6.10.1 WDQS Interval Oscillator (cont'd)

Table 71 — WDQS Oscillator Matching Error Specification

Parameter	Symbol	Min	Max	Unit	Notes
WDQS Oscillator Matching Error: voltage variation	$WOSC_{Match(V)}$			ps	1, 2, 3, 5
WDQS Oscillator Matching Error: temperature variation	$WOSC_{Match(T)}$			ps	1, 2, 3, 5
WDQS Oscillator Offset for voltage variation	$WOSC_{offset(V)}$			ps	2, 5
WDQS Oscillator Offset for temperature variation	$WOSC_{offset(T)}$			ps	2, 5

NOTE 1 The $WOSC_{Match}$ is the matching error per between the actual WDQS and WDQS interval oscillator over voltage or temp.
NOTE 2 This parameter will be characterized or guaranteed by design.
NOTE 3 The input stimulus for t_{RX_DQS2DQ} will be consistent over voltage and temp conditions.
NOTE 4 $t_{RX\_DQS2DQ(V,\ or\ T)}$ delay will be the average of WDQS to DQ delay over the runtime period.
NOTE 5 The matching error and offset of the oscillator came from WDQS Interval oscillator.
NOTE 6 These parameters are defined per device.

6.10.2 tWDQS2DQ_I Offset due to Temperature and Voltage Variation

As temperature and voltage change on the HBM4 DRAM, the WDQS 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 $t_{WDQS2DQ\_I}$ offset due to temperature and voltage variation specification can be used for instances when the oscillator cannot be used to control the $t_{WDQS2DQ\_I}$.

6.11 DCA and DCM

6.11.1 Duty Cycle Adjuster (DCA)

HBM4 DRAMs support a Duty Cycle Adjuster (DCA) that allows the memory controller to adjust the DRAM internally generated WDQS to compensate for a systemic duty cycle error on WDQS. The DCA is located before the WDQS divider or equivalent (see High Level Block Diagram Example of Clocking Scheme Figure 10). The DCA will affect the WDQS duty cycle for both Write and Read operations.

A separate DCA is provided for each WDQS (See Table 23):

- the DCA for WDQS0 (PC0) is controlled via MR11 OP[3:0];
- the DCA for WDQS1 (PC1) is controlled via MR11 OP[7:4];

A range of -7 steps to +7 steps is supported as shown in Figure 83 and changes the effective internal WDQS duty cycle as follows:

- a positive value increases the effective t_{WQSH} time and decreases the effective t_{WQSL} time.
- a negative value decreases the effective t_{WQSH} time and increases the effective t_{WQSL} time.

The use of the DCA is optional for the memory controller and is not supported at CK clock frequencies lower than f_{CKDCA}; at those frequencies it is required to disable the DCA by setting the DCA code to the default value (0000).

A duty cycle adjustment, with or without a duty cycle monitor sequence, shall be performed prior to WDQS-to-CK Alignment Training.

An example of the effect of a DCA code change to the WDQS duty cycle is shown in Figure 84. The maximum offset and step are given in Table 72.

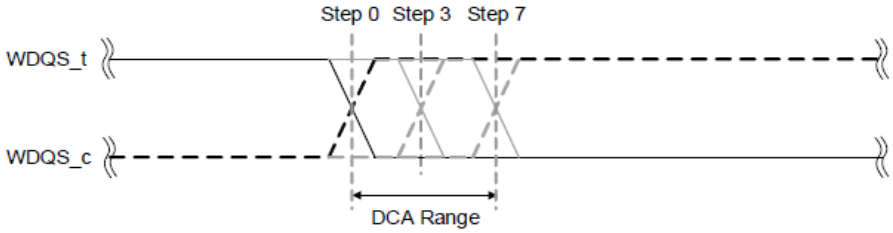


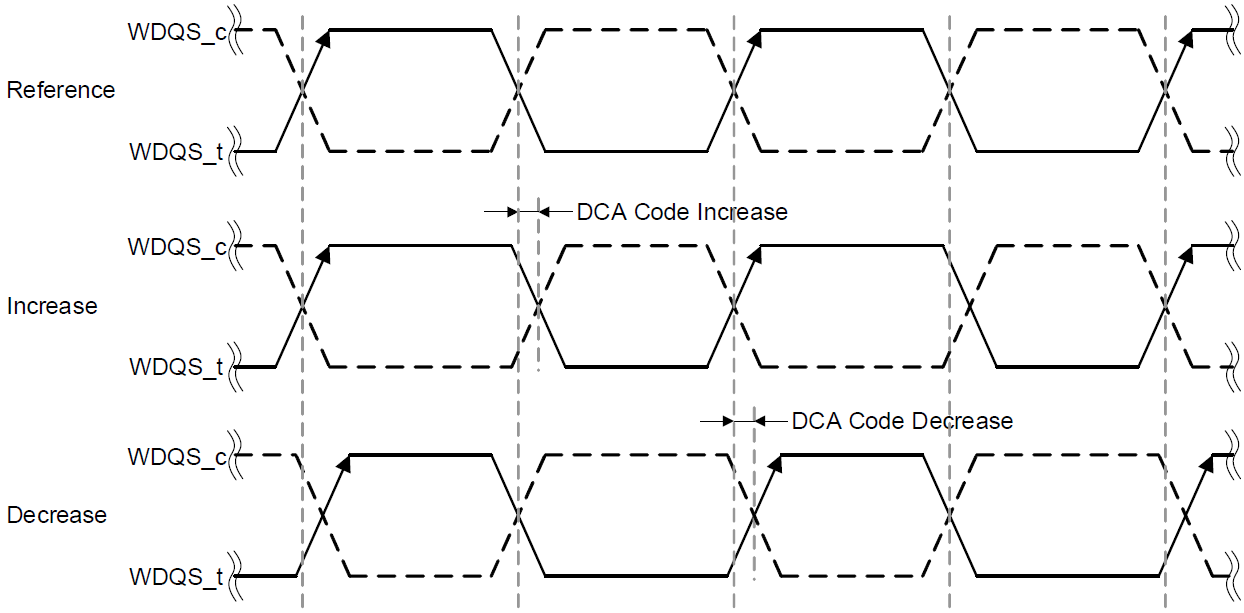
Figure 83 — Duty Cycle Adjuster Range

Table 72 — DCA Maximum Offset and Step Size

Parameter	Min	Max	Unit	Notes
Duty cycle adjuster maximum offset	15	35	ps	1, 2
Duty cycle adjuster single step size	2	5	ps	1, 3

Note 1 The values are guaranteed by design.
Note 2 The parameter describes the absolute maximum offset from step 0 to step +7 or from step 0 to step -7.
Note 3 The single step size reflects the non-linearity of each step.

6.11.1 Duty Cycle Adjuster (DCA) (cont'd)



NOTE 1 Refer to the AC Timings section for the definition of t_{WQSH}, t_{WQSL} and t_{WDQS}.

Figure 84 — Relationship Between WDQS Waveform and DCA Code Change (Example)

6.11.2 Duty Cycle Monitor (DCM)

The HBM4 DRAM includes a Duty Cycle Monitor (DCM) that allows the memory controller to observe the DRAM internal WDQS clock tree duty cycle distortion.

The DCM is controlled via MR6 OP[7:6] (see Table 17). Once DCM is enabled by setting MR6 OP6 to 1, the DCM will start the WDQS duty cycle distortion measurement and provide the result on DERR0 for DWORD0 (PC0) and on DERR1 for DWORD1 (PC1) after waiting at least t_{DCMM} time. An even number of continuous WDQS pulses will be required for the complete duration of the measurement cycle, from the MRS command that initiates the measurement until the t_{DCMM} timing has been met. The result will remain valid until the DCM is disabled by setting MR6 OP6 back to 0. The DERR outputs will then return to their default state latest after t_{MOD} has elapsed.

Commands allowed while in this mode are REFab, REFpb, RFMab, RFMpb, RNOP, CNOP and MRS to disable the duty cycle monitor. Internal current spikes generated by the use of REFab, REFpb, RFMab and RFMpb commands in this mode may negatively impact the training result. Controllers that cannot account for this impact should avoid use of REFab, REFpb, RFMab and RFMpb commands in this mode.

The DCM is not supported at CK clock frequencies lower than fCKDCA, as in the case of DCA.

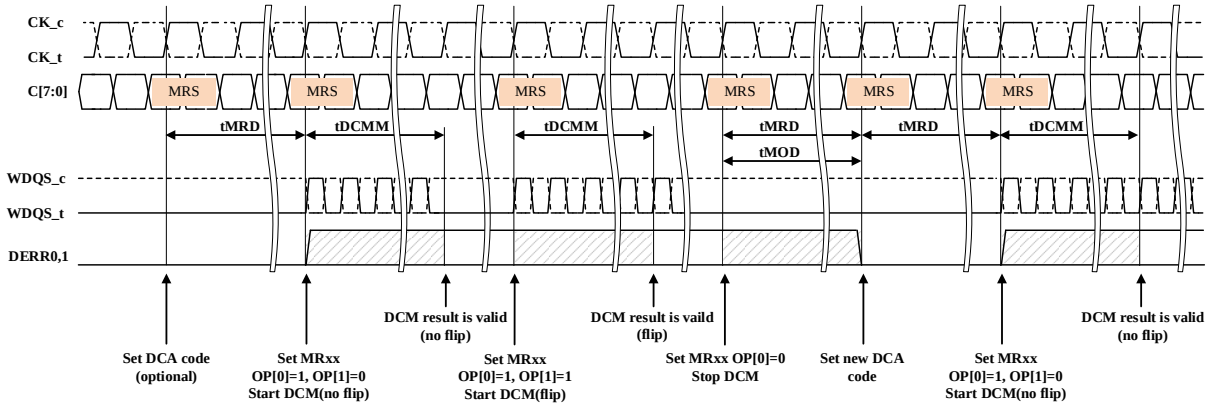
Table 73 — DCM Measurement Result

WDQS Duty Cycle	Result (DERR0, DERR1)	Notes
< 50%	LOW	1
≥ 50%	HIGH	
NOTE 1 The result is valid a time t_{DCMM} after enabling DCM		

The following example command sequence may be used for WDQS duty cycle adjustment (see also Figure 85):

1. Enable both WDQS strobes;
2. Enable DCM and wait for t_{DCMM};
3. Observe the measurement result via DERR0 and DERR1 outputs;
4. Disable DCM and wait for t_{MOD}; DERR0 and DERR1 outputs return to their default state;
5. Issue an MRS command to set an appropriate DCA codes for both WDQS strobes and wait for t_{MOD};
6. Repeat steps 2 to 5 as needed;
7. Perform WDQS-to-CK alignment training.

6.11.2 Duty Cycle Monitor (DCM) (cont'd)



NOTE 1 The host may send continuous WDQS pulses throughout the whole duty cycle adjustment procedure, in addition to the required WDQS pulses as shown in the figure.

Figure 85 — Example Sequence for WDQS Duty Cycle Correction

6.12 Rx Offset Calibration Training

6.12.1 Rx Offset Calibration Training Description

HBM4 DRAM supports Rx offset calibration (RXoffC) training for adjusting DQ Rx as an optional feature. Support for the feature is indicated in the Device ID RXoffC bit field, which is started/stopped using MR8 OP1.

If supported, it is recommended to perform Rx offset calibration whenever training is performed whether as part of the power-up initialization process or during normal operation to operate every powerup or initialization training sequence to cope with operating condition change. Since Rx offset calibration affects DRAM write calibration training, Rx offset calibration training must be performed before VREFD training and WDQS-to-CK alignment.

Before the DRAM starts the training, DQ channel should be floated by the host. DQ ODTs shall be automatically enabled after MRS is issued for starting an offset calibration training.

6.12.2 Rx Offset Calibration Training Sequence

The following sequence must be satisfied for offset calibration training.

1. Issue MRS command for starting an offset calibration training. At this time, DQ channel should be floated by the Host.
2. Wait tOSCAL until the HBM4 DRAM completes the offset calibration.
3. Issue MRS command for exiting the offset calibration training.

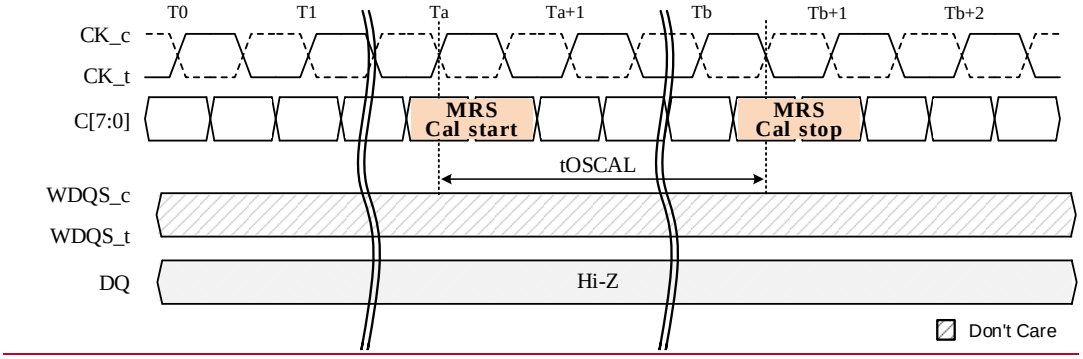


Figure 86 — Rx Offset Calibration Training Timing

Table 74 — Rx Offset Calibration Training Time Parameter

Item	Symbol	Min.	Max.	Unit
Rx Offset Calibration Training Time	tOSCAL	-	6	µs

Table 75 shows Mode Register for the Rx offset calibration.

Table 75 — MR8 Rx Offset Calibration Control

Field	Bits	Description	Note
Rx Offset Calibration Start/Stop	OP[1]	0 - Stop (default) 1 – Start	1
NOTE 1 Rx Offset Calibration is optional features and must set to 0 if it is unavailable.			

Table 76 shows DEVICE_ID for the RX offset calibration.

Table 76— Rx Offset Calibration Device ID

Bit Position	Bit Field	Type	Description	Note
TBD	RXoffC	R	0 – Rx offset calibration is not supported 1 – Rx offset calibration is supported	

6.13 Self Repair

The HBM4 DRAM supports self repair to help improve SIP assembly yield or to achieve a high level of system reliability by scanning for and repairing failures in the DRAM during the initialization process.

The IEEE1500 instructions SELF_REP and SELF_REP_RESULTS are associated with the HBM4 self repair functionality. Self repair is initiated by setting WIR [7:0] to '1Ah' which loads the SELF_REP instruction. Since the instruction works on 8 or 16 channels at a time, WIR [13:8] must be set to '38h' or '39h' to select one half of the channels to run on, or must be set to '3Ah', '3Bh', '3Ch', or '3Dh' to select one quarter of the channels to run on. A parallel operation of Self Repair on all of groups of 8 or 16 channels are not supported. SELF_REP clock source can be WRCK as a direct clock source or reference clock source or an internal clocked mode independent of WRCK and independent of any I/O functional clocks.

Setting REP_TYPE field, bits[3:2], of the SELF_REP instruction to '11b' will instruct the DRAM to start the first phase of the self repair process which is 'self-test' to identify any hard failures. The SELF_REP instruction works on one SID at a time and must be run on each SID separately by using the SID_SELECT field, bits [5:4]. The number of SELF_REP instructions required to check all channels and SIDs is listed in Table 77. The SELFR_REF_RATE field, bits [7:6], must be set by the host to control temperature compensated refresh rate.

Table 77 — SELF_REP Instruction vs Stack Height

Stack Height	SID	Min # SELF_REP on 8 Channel Groups to cover all 32 Channels	Min # SELF_REP on 16 Channel Groups to cover all 32 Channels
4H	SID0	4	2
8H	SID0, SID1	8	4
12H	SID0, SID1, SID2	12	6
16H	SID0, SID1, SID2, SID3	16	12
NOTE 1 This table only shows minimum number of channels covered by SELF_REP. All stack height configuration supports. 4/8/12/16Hi configutaion supports both 8 and 16 channel grouping.			

The 'self-test' will use vendor specific pattern(s) that detect hard failures in the HBM4 DRAM. Once the 'self-test' phase is complete the DRAM will proceed to the 'auto-repair' phase. The 'auto-repair' automatically repairs failed address(es) from the 'self-test' phase with the number of failed addresses repaired vendor specific.

SELF_REP may be issued any time after the device has been properly initialized, specifically t_{INIT3} has been met and the DRAM is in the all banks idle state. Since the SELF_REP instruction operates on 8 channels at a time as selected by WIR[13:8], the 8 channels identified in 3Ah, 3Bh, 3Ch or 3Dh or the 16 channels identified in 38h or 39h must be in the all banks idle state. See the vendor specification for the mapping of channels for 38h and 3Dh.

During the self repair process the host can poll the DRAM for status using the SR_PROGRESS field of the SELF_REP instruction. The DRAM will report whether the "self-test" is in progress, the "auto-repair" is in progress or the self repair process has completed or not running. The SELF_REP instruction must be kept in the WIR during this time.

Once the self repair process is complete, the SELF_REP_RESULTS instruction can be issued to read out the results. The DRAM will report the results for each SID and will indicate whether; i) fails remain, ii) unrepairable fails remain; iii) SELF_REP should be run again; or iv) Self Repair has not run since INIT or no fails remain after most recent run.

6.12 Self Repair (cont'd)

If after running both the 'self-test' and 'auto-repair' phases the results indicate that fails remain, the SELF REP instruction can be issued to run only the 'auto-repair' phase to repair additional fails by setting the REP_TYPE to '10b'. With REP_TYPE set to '10b' the DRAM will repair additional row addresses from the previous 'self-test'. If additional fails remain, the host can continue to issue SELF_REP instructions with REP_TYPE= '10b', followed by SELF_REP_RESULTS, until the DRAM reports '00b' to indicate that there are no fails remaining. If the DRAM reports '11b', this is an indication to the host to run SELF_REP again with either REP_TYPE set to '01b' ('self-test only) or '11b' (self-test and auto-repair) to load internal fail addresses from the 'self -test' phase.

With REP_TYPE set to '01b', the SELF_REPAIR instruction will only run the 'self-test' phase and the host can check the results after completion to decide next steps.

If the host runs the 'auto-repair' only without previously having run the 'self-test' phase, then the SELF_REP_RESULTS instruction will report '00b'as there are no failing address(es).

If the DRAM reports "Unrepairable fails remain" on a channel, this indicates that there are not enough repair elements remaining to repair failed addresses latched during the 'self-test' phase. The host can decide whether to run self repair again to repair other channels or complete the repair process.

Once the self repair process is complete, the host must issue a reset of the DRAM by driving RESET_n to low and then following the Initialization Sequence with Stable power.

The host is able to cancel the self repair in progress by using the SELF_REP instruction with REP_TYPE set to '00b', however only the "self-test" phase can be cancelled. The SR_PROGRESS field of the SELF_REP will be set to '00b' and the host must wait t_{SELF_CANCEL} before any additional SELF repair. If no further repair is needed, the host must reset the DRAM.

If the host does not use the polling to determine completion then the following timing parameters will indicate the completion of the REP_TYPE.

Table 78 — SELF_REPAIR Timings

Parameter	REP_TYPE	Phase	Min/Max	Unit
t_{SELF_HEAL}	11b	Self-test and Auto-repair	Max	s
t_{SELF_REP}	10b	Auto-repair	Max	s
t_{SELF_NR}	01b	Self-test	Max	s
t_{SELF_CANCEL}	00b	Self-test cancel time	Max	µs

Self repair resources are vendor specific. The Self repair resources can be shared with the hard/soft repair resources if the DRAM supports two or more resources per bank. The SHARED_REP_RES field of the DEVICE_ID indicates whether the DRAM supports separate or shared resources. If the DRAM shares the resources the host can use the HS_REP_CAP instruction to tell how many resources are available for self repair. When resources shared, any repairs done by self repair will update the resources per bank. The number of repair done by the DRAM per SELF_REP instruction is vendor specific.

6.12 Self Repair (cont'd)

If the DRAM shares resources with self repair, the DRAM must not use all the available resources in a bank. One resource per bank must be left for the host to perform soft repair. If the host desires to allow self repair to use all of the shared resources then the SHARED_OVERRIDE field can be set to '1b'. If the resources are not shared the SHARED_OVERRIDE field will be ignored. Before any self repair, the host is recommended to clear any soft repairs using the undo function on the channels the host plans to run self repair, or a chip reset. Failure to do so could result in the DRAM using the shared resources and operation is then not guaranteed, including loss of data. After the self repair is complete the host can perform soft repairs after checking if resources are available. Table 79 illustrates the expected behavior of the DRAM and the expected SEL_REP_RESULTS when SHARED_OVERRIDE is set to '0' and '1'.

Table 79 — SELF_REP – Expected DRAM Behavior When Resources Shared

Case	Resource vs Fails	Examples						
		Resource(s) before SELF_REP	Fails	Override (0 = default)	DRAM behavior	Results	Resource(s) after SELF_REP	Subsequent Hard/ Soft repair
1	Resources < Fails	1	2	0 (No)	No repair	Unrepairable fails remain	1	Yes
				1 (Yes)	Auto-repair	Unrepairable fails remain	0	No
2	Resources = Fails	2	2	0 (No)	No repair	Unrepairable fails remain	2	Yes
				1 (Yes)	Auto-repair(s)[1]	No fails remain (Fails remain)[2]	0 (1)[3]	No (Yes)[3]
3	Resources > Fails	3	2	0 (No)	Auto-repair(s)[1]	No fails remain (Fails remain)[2]	1 (2)[4]	Yes
				1 (Yes)	Auto-repair(s)[1]	No fails remain (Fails remain)[2]	1 (2)[4]	Yes

NOTE 1 The number of repairs per SELF_REP instruction is vendor specific.
NOTE 2 If the DRAM does 1 repair per SELF_REP then after the initial SELF_REP the DRAM will report 'fails remain' and the host will need to issue a second SELF_REP to repair the other fail so that the results are 'no fails remain'. If the DRAM repairs both fails in one SELF_REP instruction then the results will be 'no fails remain.
NOTE 3 If the DRAM only does 1 repair per SELF_REP, the host has the option to do no further repair and leave the remaining resource for soft repair or use up the one remaining resource with an additional SELF_REP. If the DRAM does more than 1 repair per SELF_REP then the host cannot do subsequent soft repair.
NOTE 4 For Case 3 the number of resources repaired depends on whether the initial SELF_REP repairs 1 fail or both.

6.12 Self Repair (cont'd)

Figure 87 provides a flow chart showing the 3 flows for REP_TYPE field of the SELF_REP instruction.

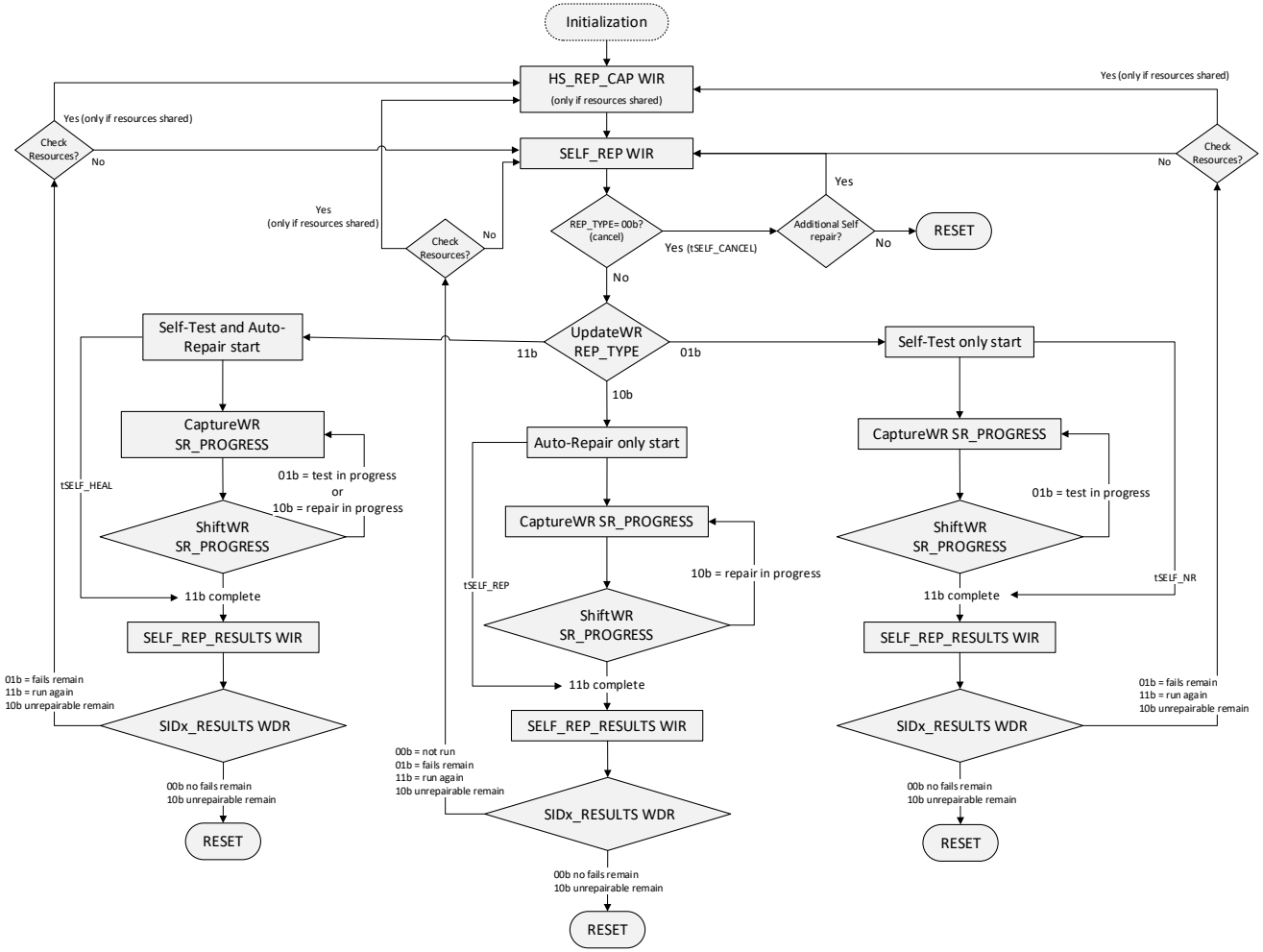


Figure 87 — Self Repair Flowchart

7 Operating Conditions

7.1 Absolute Maximum DC Rating

Table 80 — Absolute Maximum DC Ratings

Parameter	Symbol	Rating	Unit	Notes
Voltage on V_{DDC} relative to V_{SS}	V_{DDC}	-0.3 to 1.4	V	1, 2
Voltage on V_{DDQ} relative to V_{SS}	V_{DDQ}		V	1, 2, 4
Voltage on V_{DDQL} relative to V_{SS}	V_{DDQL}	-0.3 to 0.8	V	1, 2
Voltage on V_{PP} relative to V_{SS}	V_{PP}	-0.3 to 2.1	V	1, 2
Voltage on any signal pin relative to V_{SS}	V_{IN}, V_{OUT}		V	1, 2, 4
Storage Temperature	$T_{STORAGE}$		°C	1, 2

NOTE 1 Stresses greater than those listed under "Absolute Maximum Ratings" 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 standard is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
NOTE 2 See HBM4 Power-up and Initialization Sequence for the relationship between the power supplies.
NOTE 3 Storage temperature is the case surface temperature on the center/top side of the HBM4 device. For the Measurement conditions, please refer to JESD51-2 standard.
NOTE 4 The vendor's datasheet shall be consulted for the rating values.

7.2 Recommended DC Operating Condition

For HBM4, the need arose to define more than one VDDQ range due to the requirements of the HBM4 specification and device designs envisioned in the lifetime of HBM4. The typical levels listed in Table 81 represent the known values at the time of publication.

Vendor datasheets should be consulted for actual VDDQ voltage(s) supported, as factors such as process technology and supported system voltage(s) may require typical operating voltages to be added, dropped or maintained over time.

Table 81 — Recommended DC Operating Condition

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Notes
Core Supply Voltage	V_{DDC}	1.018	1.05	1.124	V	1
I/O Supply Voltage	V_{DDQ}	0.873	0.9	0.963	V	1
		And / or				2
		0.776	0.8	0.856		1
Supply Voltage for TX Driver Output Stage	V_{DDQL}	0.38	0.4	0.44	V	1
Pump Voltage	V_{PP}	1.746	1.8	1.95	V	1

NOTE 1 The voltage ranges are defined at the HBM4 DRAM micropillars. DC bandwidth is limited to 20MHz.
NOTE 2 HBM4 must support at least one typical VDDQ voltage and the following tolerances must be supported. The minimum value of VDDQ = 0.97 * typical VDDQ and the maximum value of VDDQ = 1.07 * typical VDDQ.

7.3 Operating Temperature

Table 82 — Operating Temperature

Parameter		Symbol	JESD402-1 Operating Juction Temp Range	Minimum	Maximum	Unit	Notes
Operating Temperature	Standard	T_N				°C	1
Operating Temperature (Optional)	Extended	T_E				°C	1, 2, 3

NOTE 1 The operating temperature refers to the junction temperature of all memory die(s) and the optional logic die of the HBM4 DRAM. The host is required to monitor the operating temperature via the IEEE1500 test port instructions TEMPERATURE and CHANNEL_TEMPERATURE. The host is also required to monitor the CATTRIP output that signals if the junction temperature of any die in the HBM4 DRAM exceeds a catastrophic trip-point level that could result in permanent damage of the device.

NOTE 2 HBM DRAM may require additional Refresh cycle. Refer to vendor datasheet.

NOTE 3 HBM4 operating temperatures are vendor specific. Please see JESD402-1B or later for the TJopr ranges that can be supported and vendor specifications for the specific ranges supported.

8 Electrical Characteristics and DQ/CA Rx

8.1 Leakage Current

Table 83 — Input Leakage Current

Parameter	Symbol	Minimum	Max	Unit	Notes
Input leakage current for AWORD and DWORD inputs	I_L	-50	50	µA	1
NOTE 1 Any input $0V \leq V_{IN} \leq V_{DDQL}$. (All inputs pins including IEEE1500 not under test = 0V)					

8.2 Capacitance

Table 84 — Input/Output Capacitance

Parameter	Symbol	Min	Max	Unit	Notes
Input/Output Capacitance – DQs, DBI, DPAR, ECC, SEV	C_{IO}		0.4	pF	1
Input Capacitance – Row and pF Column Address	C_{ADDR}		0.4	pF	1
Input/Output Capacitance – Read Strobe	C_{RDQS}		0.4	pF	1
Input Capacitance – Write Strobe	C_{WDQS}		0.4	pF	1
Input Capacitance – Clock	C_{CK}		0.4	pF	1
Input/Output Capacitance – DERR, AERR	C_{ERROR}		0.4	pF	1
NOTE 1 This parameter is not subject to production test.					

8.3 DQ Rx Voltage and Timing

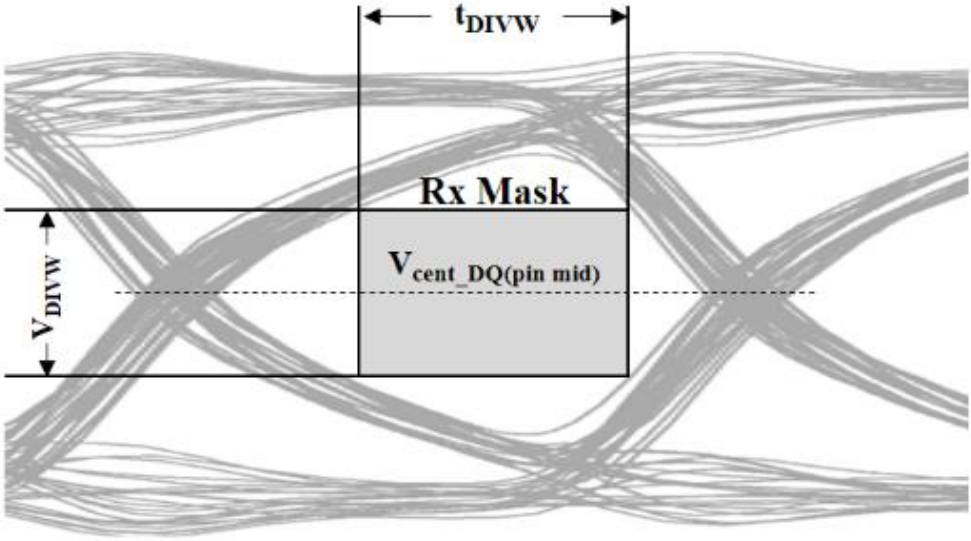


Figure 88 — DQ receiver mask

The DQ input receiver mask for voltage and timing is shown in Figure 88 is applied per pin. The DQ Rx mask (V_{DIVW}, t_{DIVW}) defines the area the input signal must not encroach in order for the DQ input receiver to successfully capture an input signal with a BER of lower than TBD. The mask is a receiver property.

$V_{cent\_DQ(pin\_mid)}$ is defined as the midpoint between the largest V_{cent_DQ} voltage level and the smallest V_{cent_DQ} voltage level across all DQ pins for a given DRAM TBD (Determined by DRAM V_{REFD} training granularity, i.e., component/channel/DWORD) level. Each DQ V_{cent} is defined by the center, i.e., widest opening, of the cumulative data input eye as depicted in Figure 89. This clarifies that any DRAM TBD level variation must be accounted for within the DQ Rx mask.

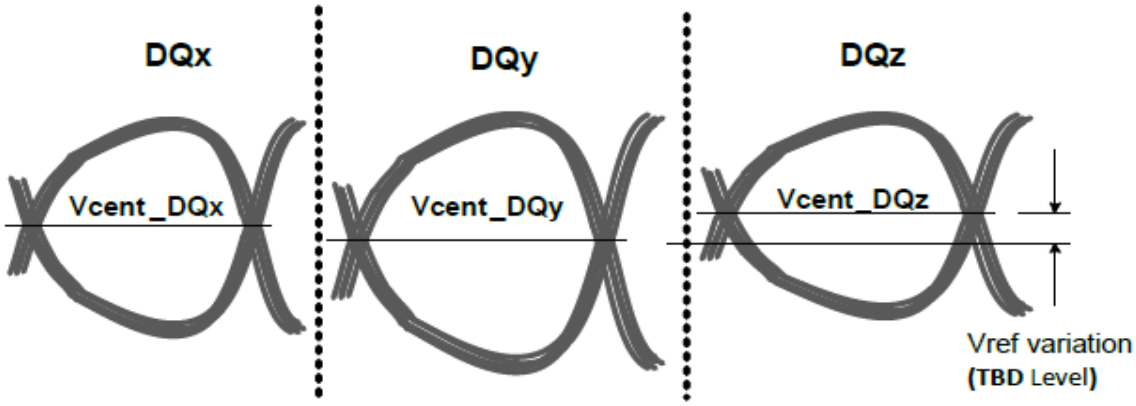


Figure 89 — Across Pin V_{REFD} Voltage Variation

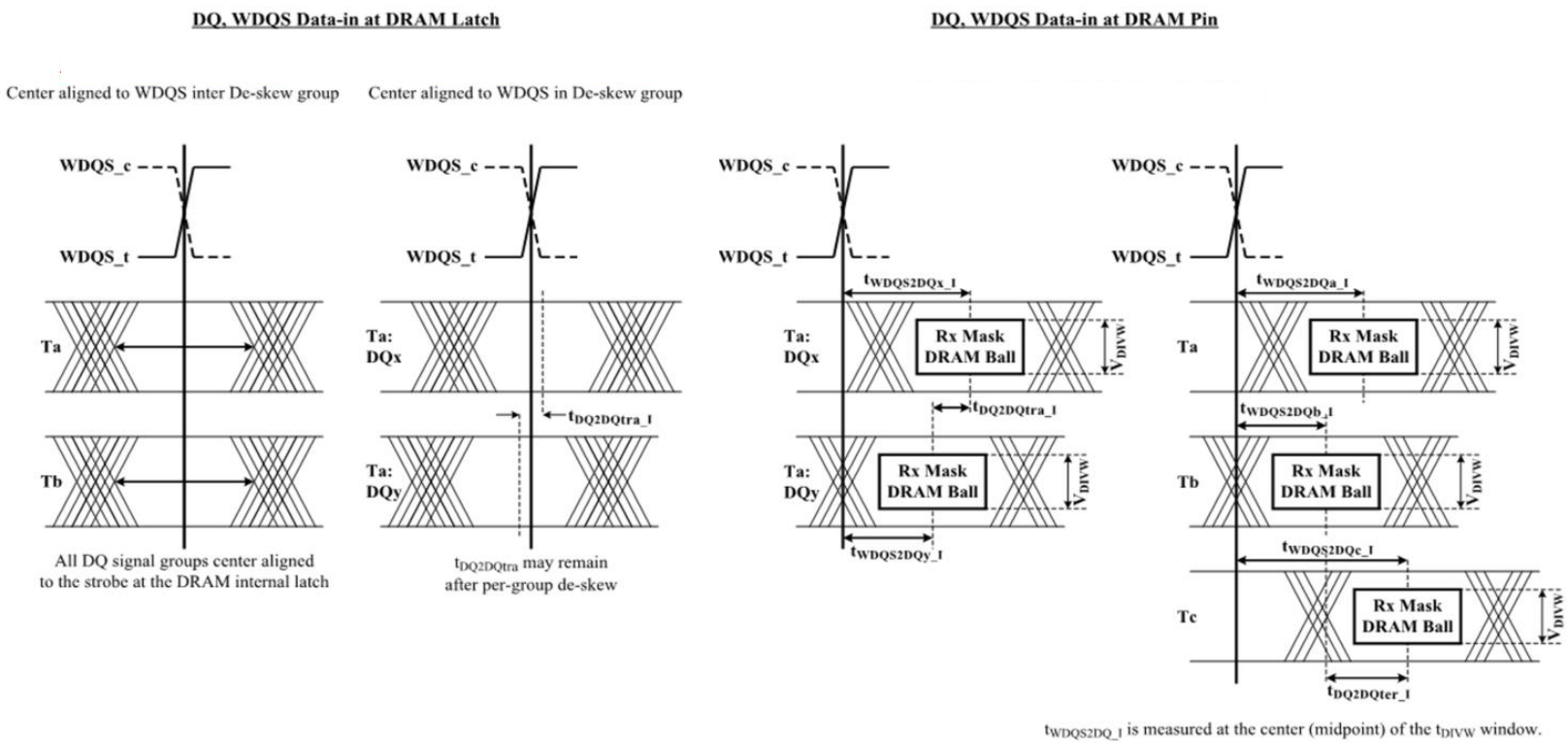
8.3 DQ Rx Voltage and Timing (cont'd)

Table 85 — Input Receiver Voltage Level Specification

Parameter	Symbol	Speed Bin										Unit	Notes
		4.8 Gbps		5.2 Gbps		5.6 Gbps		6.0 Gbps		6.4 Gbps			
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max		
DQ Rx Mask voltage p-p	V_{DIVW}		120		120		120		120		120	mV	1, 2
Input Slew Rate over V_{DIVW}	SR_{IN_DIVW}	1	7	1	7	1	7	1	7	1	7	V/ns	3
Rx single pulse amplitude	V_{IHLDQ_AC}	190		190		190		190		190		mV	4

NOTE 1 DQ Rx mask voltage and timing parameters are applied per pin and includes DRAM DQ to WDQS voltage AC noise impact for frequencies > 20MHz at a fixed temperature on a die.
NOTE 2 DQ Rx mask voltage V_{DIVW} has to be centered around $V_{cent\_DQ(pin\_mid)}$.
NOTE 3 Input slew rate over V_{DIVW} mask centered at $V_{cent\_DQ(pin\_mid)}$.
NOTE 4 DQ single input pulse amplitude into the receiver has to meet or exceed $V_{IHL\mathbf{DQ}\_AC}$ at any point over the total UI. No timing requirement above level. $V_{IHL\mathbf{DQ}\_AC}$ is the peak to peak voltage centered around $V_{cent\_DQ(pin\_mid)}$ such that $V_{IHL\_AC}/2$ min has to be met both above and below $V_{cent\_DQ(pin\_mid)}$.

8.3 DQ Rx Voltage and Timing (cont'd)

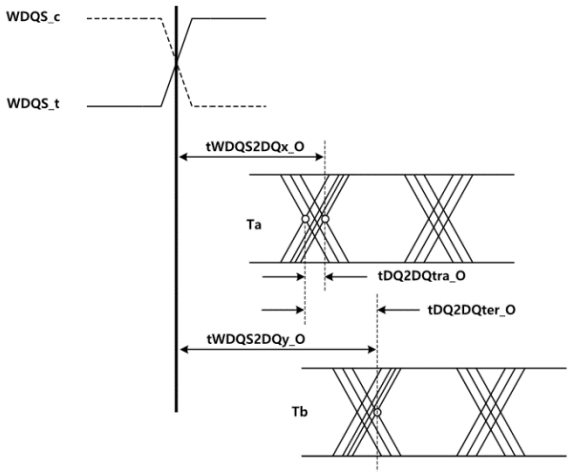


NOTE 1 DQx and DQy are in the same data de-skewing group (Ta) in a DWORD.
NOTE 2 DQx represents the max $t_{WDQS2DQx\_I}$ in group Ta, and DQy represents the min $t_{WDQS2DQy\_I}$ in group Ta, in this example.
NOTE 3 $t_{WDQS2DQc\_I}$ represents the max $t_{WDQS2DQ\_I}$ and $t_{WDQS2DQb\_I}$ represents the min $t_{WDQS2DQ\_I}$ in a DWORD, where Ta, Tb, and Tc are signals in each de-skewing group in this example. $t_{WDQS2DQa\_I}$ represents the reference of $t_{WDQS2DQ\_I}$ for comparison in this example.
NOTE 4 Timing different between $t_{WDQS2DQx\_I}$ and $t_{WDQS2DQy\_I}$ represents the max $t_{DQ2DQtra\_I}$, in this example.
NOTE 5 Timing different between $t_{WDQS2DQb\_I}$ and $t_{WDQS2DQc\_I}$ represents the max $t_{DQ2DQter\_I}$, in this example.
NOTE 6 Refer to Table 44 for the signals that belong to each signal group.
NOTE 7 The skew between the earliest and latest data input within a DWORD is defined by tDQ2DQter_I(max). The skew among data inputs within a byte group is limited to tDQ2DQtra_I(max), and tDQ2DQtra_I is included in tDQ2DQter_I.

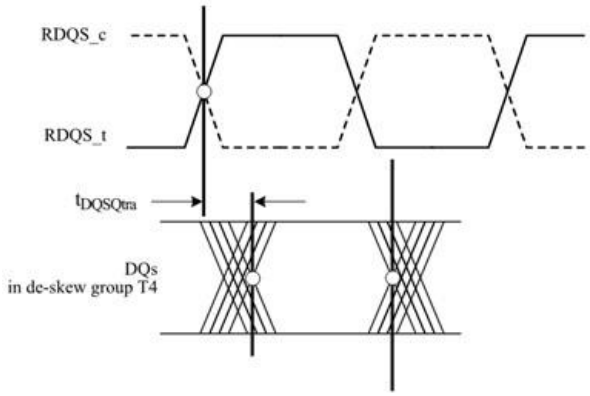
Figure 90 — DQ to WDQS Timings ($t_{WDQS2DQ\_I}$, $t_{DQ2DQtra\_I}$ and $t_{DQ2DQter\_I}$) at DRAM pins referenced from the internal latch

8.3 DQ Rx Voltage and Timing (cont'd)

All of the timing terms in DQ to WDQS_t are measured from the WDQS_t/WDQS_c to the center midpoint of the t_{DIVW} window taken at the V_{DIVW} voltage levels centered around $V_{cent\_DQ(pin\_mid)}$. In Figure 90 the timings at the pins are referenced with respect to all DQ signal groups center aligned to the DRAM internal latch. The data to data offset in write de-skew group, $t_{DQ2DQtra\_I}$, is defined as the difference between the min and max $t_{WDQS2DQ\_I}$ for a given de-skew group. The data to data offset in different write de-skew group, $t_{DQ2DQter\_I}$, is defined as the difference between the min and max $t_{WDQS2DQ\_I}$ for a given DWORD. $t_{WDQS2DQ\_O}$ is defined as the WDQS to read data and RDQS offset.



$t_{DQ2DQtra\_O}$ and $t_{DQ2DQter\_O}$



$t_{DQSQtra\_O}$

NOTE 1 $t_{DQ2DQtra\_O}$ is defined at the same input pattern for all DQ in the same de-skew group signals.
NOTE 2 $t_{DQ2DQter\_O}$ is defined at the skew between de-skew group signals (T_a, T_b in this example) at the latest valid transition of the associated DQ pins. $t_{WDQS2DQx\_O}$ represents the min $t_{WDQS2DQ\_O}$ and $t_{WDQS2Dqy\_O}$ represents the max $t_{WDQS2DQ\_O}$, in this example.
NOTE 3 $t_{DQSQtra}$ is defined at the skew between RDQS to the last valid transition of the DQ pins in de-skew group T4.
NOTE 4 The skew between the earliest and latest data output within a DWORD including RDQS_t/_c is defined by tDQ2DQter_O(max). The skew among data outputs within a byte group is limited to tDQ2DQtra_O(max), and tDQ2DQtra_O is included in tDQ2DQter_O.

Figure 91 — Read Data Timing Definitions of $t_{DQ2DQtra\_O}$, $t_{DQ2DQter\_O}$, and $t_{DQSQtra\_O}$

8.4 AWORD Signaling

Table 86 — AWORD Receiver Voltage Level Specification

Parameter	Symbol	Min	Max	Unit	Notes
Input HIGH Voltage	V_{IHCA}	V_{REFCA} + 0.1		V	1, 3
Input LOW Voltage	V_{ILCA}		V_{REFCA} – 0.1	V	1, 3
Command/Address Rx single pulse amplitude	V_{IHLCA_AC}	240		mV	2, 3

NOTE 1 V_{REFCA} based input receiver enabled (see MR13, Table 25). For C, R, ARFU and APAR inputs.
NOTE 2 CA single input pulse amplitude into the receiver has to meet or exceed V_{IHL_AC} at any point over the total UI. No timing requirement above level. V_{IHLCA_AC} is the peak to peak voltage centered around $V_{DDQL}/2$ such that $V_{IHLCA\_AC}/2$ min has to be met both above and below $V_{DDQL}/2$.
NOTE 3 Parameter is applied to all speed bins.

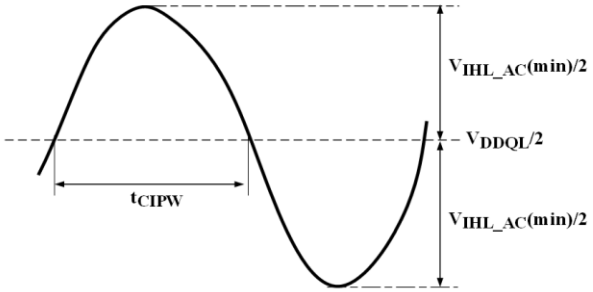


Figure 92 — CA Single Pulse Amplitude and Pulse Width

8.5 CK and WDQS Input Signaling

Table 87 — CK and WDQS Input Voltage Level Specification

Parameter	Symbol	Min	Max	Unit	Notes
CK clock Input Differential Input Voltage	V_{IDCK}	160		mV	1, 7
CK clock Differential Input Cross-point Voltage	V_{IXCK}	$V_{DDQL}/2$ – 40mV	$V_{DDQL}/2$ + 40mV	V	2, 7
WDQS Differential Input Voltage	V_{IDWDQS}	150		mV	3, 7
WDQS Differential Input Cross-point Voltage	V_{IXWDQS}	$V_{DDQL}/2$ – 30mV	$V_{DDQL}/2$ + 30mV	V	4, 7
RDQS Differential Output Cross-point Voltage	V_{OXRDQS}	$V_{DDQL}/2$ – TBD	$V_{DDQL}/2$ + TBD	V	5, 6, 7
WDQS Differential Input Slew Rate	SR_WDQS	TBD	TBD	V/ns	7

NOTE 1 V_{IDCK} is the magnitude of the difference between the input level on CK_t and the input level on CK_c.
NOTE 2 The input reference level for timings referenced to CK is the point at which CK_t and CK_c cross.
NOTE 3 V_{IDWDQS} is the magnitude of the difference between the input level on WDQS_t and the input level on WDQS_c.
NOTE 4 The input reference level for timings referenced to WDQS is the point at which WDQS_t and WDQS_c cross.
NOTE 5 Includes VDDQL, VDDQ AC noise impact of TBD mV (pk-pk) at the DRAM supply microbumps; AC noise includes system PDN impact.
NOTE 6 This parameter is guaranteed by design at the DRAM micropillars with Output Timing reference load and Read DBI enabled.
NOTE 7 Parameter is applied to all speed bins.

8.5 CK and WDQS Input Signaling (cont'd)

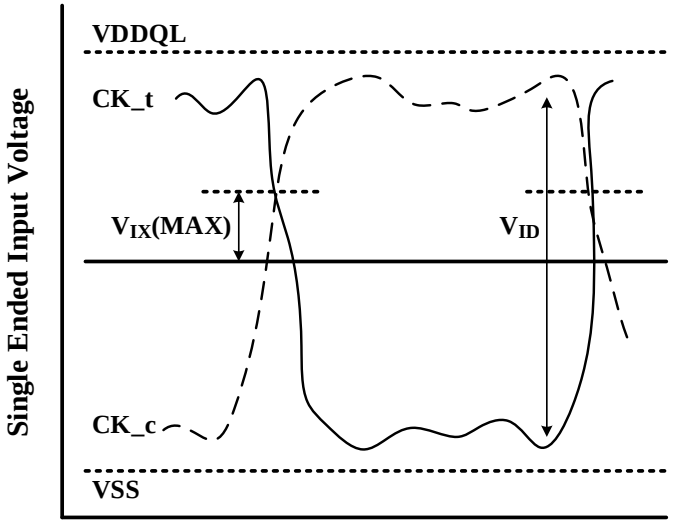


Figure 93 — CK Single Pulse

Table 88 — Differential Input Level for WDQS_t, WDQS_c

Parameter	Symbol	Min	Max.	Unit	Notes
WDQS Differential Input High	V_{IHdiff_WDQS}	TBD		mV	
WDQS Differential Input Low	V_{ILdiff_WDQS}		TBD	mV	

Table 89 — Differential Input Slew Rate Definition for WDQS_t, WDQS_c

Description	From	To	Defined by
WDQS Differential Input Slew Rate for Rising Edge (WDQS_t – WDQS_c)	V_{ILdiff_WDQS}	V_{IHdiff_WDQS}	$\lvert V_{ILdiff\_WDQS} - V_{IHdiff\_WDQS} \rvert / T_{Rdiff}$
WDQS Differential Input Slew Rate for Falling Edge (WDQS_t – WDQS_c)	V_{IHdiff_WDQS}	V_{ILdiff_WDQS}	$\lvert V_{ILdiff\_WDQS} - V_{IHdiff\_WDQS} \rvert / T_{Fdiff}$

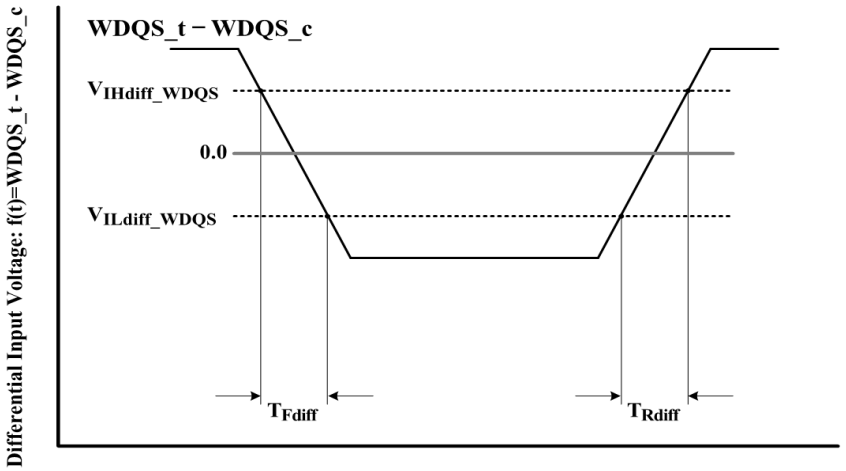


Figure 94 — Differential Input Slew Rate Definition for WDQS_t, WDQS_c

8.6 Midstack Signaling

Table 90 — Midstack Parameter Specification

Parameter	Symbol	Min	Max	Unit	Notes
Input HIGH Voltage for RESET_n and WRST_n, WRCK, SELECTWIR, SHIFTWR, CAPTUREWR, UPDATEWR and WSI inputs	V_{IHR}	0.7 x V_{DDQ}		V	1
Input LOW Voltage for RESET_n and WRST_n, WRCK, SELECTWIR, SHIFTWR, CAPTUREWR, UPDATEWR and WSI inputs	V_{ILR}		0.2 x V_{DDQ}	V	1
Output HIGH Voltage for CATTRIP and WSO outputs	V_{OHR}	0.7 x V_{DDQ}		V	
Output LOW Voltage for CATTRIP and WSO outputs	V_{OLR}		0.3 x V_{DDQ}	V	
NOTE 1 CMOS input receivers. For RESET_n, WRST_n, WRCK, SELECTWIR, SHIFTWR, CAPTUREWR, UPDATEWR and WSI inputs.					

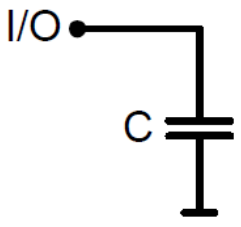
8.7 Transmit Driver Resistance

HBM4 drivers have programmable resistance settings with 20% accuracy. Driver targets, in Ohm are shown in Table 91 below.

Table 91 — Transmit Driver Resistance Specification

Nominal (Ohm)	Notes
25	1,2
20	1,2
16.7	1,2
14.3	1,2
NOTE 1 Transmit driver resistance have -20% of min. lower and +20% of upper tolerances, respect to each corresponding nominal values NOTE 2 Nominal values in this table follow corresponding Vddql x 0.5 level. See — Recommended DC Operating **Condition** sections.	

8.8 Output Timing Reference Load



NOTE $C = C_{TOTAL} - C_{IO}$; where C_{TOTAL} = 2.50 pF.

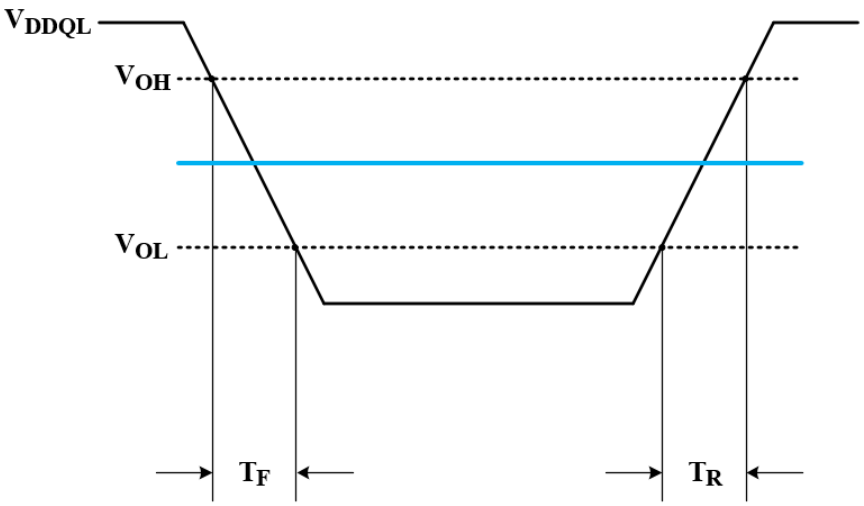
Figure 95 — Timing Reference Load

8.9 Output Voltage Level

Table 92 — Output Voltage Level

Parameter	Symbol	Min	Max	Unit	Notes
Output HIGH Voltage	V_{OH}	V_{DDQL} / 2 + 60mV		V	
Output LOW Voltage	V_{OL}		V_{DDQL} / 2 – 60mV	V	

8.10 Output Rise and Fall Time



NOTE T_R = { C_{TOTAL} x (V_{OH} – V_{OL}) } / I; T_F = { C_{TOTAL} x (V_{OH} – V_{OL}) } / I;
Where I = Transmit Drive Current in mA.

Figure 96 — Output Rise and Fall Definition

8.11 Overshoot/Undershoot

Table 93 — Overshoot/Undershoot Specification for AWORD and DWORD Signals

Parameter	4.8 Gbps	5.2 Gbps	5.6 Gbps	6.0 Gbps	6.4 Gbps	Unit	Notes
Maximum peak amplitude allowed for overshoot area	0.12	0.12	0.12	0.12	0.12	V	
Maximum peak amplitude allowed for undershoot area	0.12	0.12	0.12	0.12	0.12	V	
Maximum overshoot area above V_{DDQL}	13	12	11	10	9	mV-ns	
Maximum undershoot area below V_{SS}	13	12	11	10	9	mV-ns	

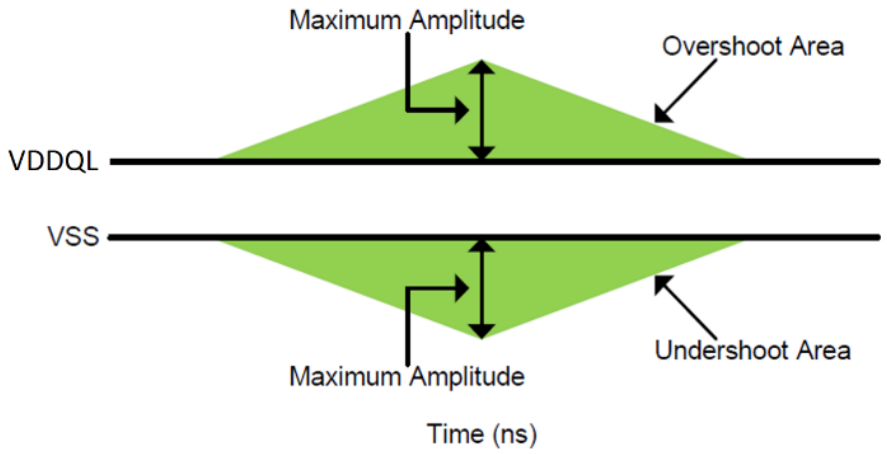


Figure 97 — Overshoot, Undershoot Definition

9 IDD Specification

9.1 IDD and IPP Specification Parameters and Test Conditions

This section defines operating current measurement conditions and loop pattern.

- I_{DD} currents are measured as time-averaged currents with all V_{DDC} microbumps of the HBM4 device under test tied together.
- I_{PP} currents use the same definitions as I_{DD} except that the current on the V_{PP} supply is measured. All V_{PP} microbumps of the HBM4 device under test are tied together for I_{PP} current measurements.
- I_{DDQ} currents are measured as time-averaged currents with all V_{DDQ} microbumps of the HBM4 device under test tied together are not included in the measurements. Instead, DRAM vendors shall provide simulated values using the I_{DD4R} measurement-loop pattern as defined in Table 97.
- I_{DDQL} currents are measured as time-averaged currents with all V_{DDQL} microbumps of the HBM4 device under test tied together. Output reference load C_{TOTAL} is 2.5 pF, and the simulated load driving current can be added to the IDDQL if vendor uses no load for this measurement.
- I_{DD}, I_{DDQ}, I_{DDQL} and I_{PP} measurements are taken with all channels of the HBM4 device simultaneously executing the same pattern. However, values in the vendor's datasheet shall be given per channel.

For IDD measurements, the following definitions apply:

- "0" and "LOW" are defined as $V_{IN} \leq V_{IL}$(max);
- "1" and "HIGH" are defined as $V_{IN} \geq V_{IH}$(min);
- WL, RL, RAS and RTP are programmed to appropriate values;
- DBIac is enabled for Reads and Writes;
- SEV and parity are disabled;
- MD is enabled in MR9;
- CNOP/RNOP commands and all address inputs are stable during idle command cycles;
- Some I_{DD} Measurement-Loop pattern use high order address bits RA13 which are not defined for all densities. In those cases the respective undefined address bit(s) shall be kept LOW.
- Basic I_{DD} Measurement Conditions are described in Table 94.
- I_{DD} Measurements are done after properly initializing the HBM4 device. This includes the pre-load of the memory array with data pattern used with I_{DD4R} measurements.
- The I_{DD} Measurement-Loop patterns shall be executed at least once before actual I_{DD} measurement is started.
- For timing parameters used with I_{DD} Measurement-Loop pattern: nRC = tRC/t_{CK}; nRAS = t_{RAS}/t_{CK}, nRP = t_{RP}/t_{CK}, and nRFC = t_{RFC}/t_{CK}. If not already an integer, round up to the next integer.

9.1 IDD and IPP Specification Parameters and Test Conditions (cont'd)

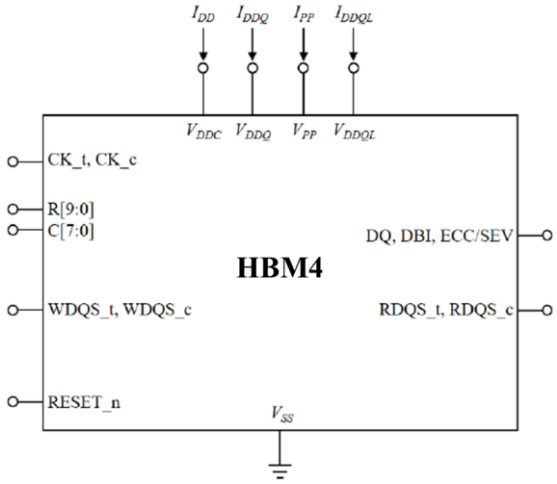


Figure 98 — Measurement Setup for IDD and IPP Measurements

Table 94 — Basic IDD/IDDQ/IPP/IDDQL Measurement Conditions

Parameter/Condition	Symbol
One Bank Activate Precharge Current: $t_{CK} = t_{CK}(min)$; t_{RC}, t_{RAS} and t_{RP} as defined in Table 95; R and C inputs are HIGH between valid commands; DQ, ECC and DBI inputs are LOW; bank and row addresses with ACT and PRE commands as defined in Table 96.	I_{DD0}, I_{DDQ0}, I_{PP0}, I_{DDQL0}
Precharge Power-down Current: Device in Precharge Power-Down is issued; $t_{CK} = t_{CK}(min)$; all banks are idle; R0 input is LOW; R[9:1] and C inputs are HIGH; DQ, ECC and DBI inputs are LOW	I_{DD2P}, I_{DDQ2P}, I_{PP2P}, I_{DDQL2P}
Precharge Power-down Current with clock stop: Device in Precharge Power-Down is issued; CK_t is LOW; CK_c is HIGH; all banks are idle; R0 input is LOW; R[9:1] and C inputs are HIGH; DQ, ECC and DBI inputs are LOW	I_{DD2P0}, I_{DDQ2P0}, I_{PP2P0}, $I_{DDQL2P0}$
Precharge Standby Current: $t_{CK} = t_{CK}(min)$; all banks are idle; R and C inputs are HIGH; DQ, ECC and DBI inputs are LOW	I_{DD2N}, I_{DDQ2N}, I_{PP2N}, I_{DDQL2N}
Active Power-down Current: Device in Active Power-Down is issued; $t_{CK} = t_{CK}(min)$; one bank is active; R0 input is LOW; R[9:1] and C inputs are HIGH; DQ, ECC and DBI inputs are LOW	I_{DD3P}, I_{DDQ3P}, I_{PP3P}, I_{DDQL3P}
Active Power-down Current with clock stop: Device in Active Power-Down is issued; CK_t is LOW; CK_c is HIGH; one bank is active; R0 input is LOW; R[9:1] and C inputs are HIGH; DQ, ECC and DBI inputs are LOW	I_{DD3P0}, I_{DDQ3P0}, I_{PP3P0}, $I_{DDQL3P0}$
Active Standby Current: $t_{CK} = t_{CK}(min)$; one bank is active; R and C inputs are HIGH; DQ, ECC and DBI inputs are LOW	I_{DD3N}, I_{DDQ3N}, I_{PP3N}, I_{DDQL3N}

Parameter/Condition	Symbol
Read Burst Current: $t_{CK} = t_{CK}$(min); all banks activated; continuous read burst across bank groups as defined in Table 97; IOUT = 0mA; Ctotal = 2.5 pF	I_{DD4R}, I_{DDQ4R}, I_{PP4R}, I_{DDQL4R}
Write Burst Current: $t_{CK} = t_{CK}$(min); all banks activated; continuous write burst across bank groups as defined in Table 98.	I_{DD4W}, I_{DDQ4W}, I_{PP4W}, I_{DDQL4W}
All-bank Refresh Burst Current: $t_{CK} = t_{CK}$(min); t_{RFCab} as defined Table 95; R and C inputs are HIGH between valid commands; DQ, ECC and DBI inputs are LOW	I_{DD5B}, I_{DDQ5B}, I_{PP5B}, I_{DDQL5B}
Per-bank Refresh Burst Current: $t_{CK} = t_{CK}$(min); Use tRFCpb and tRREFD as defined in Table 95; R and C inputs are HIGH between valid commands; DQ, ECC and DBI inputs are LOW; The order of bank is sequential from BK0 to BK15 with sequential SID.	I_{DD5P}, I_{DDQ5P}, I_{PP5P}, I_{DDQL5P}
Self Refresh Current: R0 input is LOW; R[9:1] and C inputs are LOW; DQ, ECC and DBI inputs are LOW	I_{DD6}, I_{DDQ6}, I_{PP6}, I_{DDQL6}
All-Bank Interleave Read Current: One bank in each of the 4 bank groups activated and precharged at t_{RC}(min) as defined in Table 95; continuous read burst across bank groups; Ctotal = 2.5 pF	I_{DD7}, I_{DDQ7}, I_{PP7}, I_{DDQL7}
Reset Low Current: RESET_n is LOW; CK_t, CK_c, WDQS_t, WDQS_c are LOW; R and C inputs are LOW; DQ, ECC and DBI inputs are LOW; Note: Reset low current reading is valid once power is stable and RESET_n has been LOW for at least 1ms	I_{DD8}, I_{DDQ8}, I_{PP8}, I_{DDQL8}

9.1 IDD and IPP Specification Parameters and Test Conditions (cont'd)

Table 95 — Example of Timings used for IDD Measurement-Loop Pattern

Parameter				Value	Unit
t_{RC}				48	ns
t_{RAS}				33	ns
t_{RP}				15	ns
t_{RREFD}				8	ns
t_{RFCpb}				200	ns
t_{RFCab}	24 Gb/die	4-High	6 Gb / channel	360	ns
		8-High	12 Gb / channel	410	ns
		12-High	18 Gb / channel	450	ns
		16-High	24 Gb / channel	490	ns
	32 Gb/die	4-High	4 Gb / channel	400	ns
		8-High	8 Gb / channel	450	ns
		12-High	12 Gb / channel	490	ns
		16-High	16 Gb / channel	530	ns

NOTE 1 DRAM vendors may decide to use different values for t_{RAS} and t_{RP}; however, nRAS + nRP = nRC must be achieved. nRAS = RU(tRAS/tCK), nRP = RU(tRP/tCK), nRFCpb = RU(tRFCpb/tCK), nRREFD = RU(tRREFD/tCK). If not already an integer, round up to the next integer.

9.1 IDD and IPP Specification Parameters and Test Conditions (cont'd)

Table 96 — IDD0 Measurement-Loop Pattern

Sub-Loop	Cycle Number	Row Command	Column Command	Bank Address (BA[3:0])	Row Address (RA[13:0])	Col. Address (CA[4:0])
0	0	ACT – PC0	CNOP	00h	05555h	0Ah
	1		CNOP			0Ah
	2	ACT – PC1	CNOP	00h	05555h	0Ah
	3		CNOP			0Ah
	4	RNOP	CNOP	00h	05555h	0Ah
	…	Repeat pattern until cycle (nRAS)				
	nRAS + 1	PRE – PC0	CNOP	00h	05555h	0Ah
	nRAS + 2	RNOP	CNOP	00h	05555h	0Ah
	nRAS + 3	PRE – PC1	CNOP	00h	05555h	0Ah
	nRAS + 4	RNOP	CNOP	00h	05555h	0Ah
	…	Repeat pattern until cycle (nRC – 1)				
1	nRC	repeat sub-loop 0 pattern until cycle (2 × nRC - 1); use BA = 05h and RA = 02AAAh instead				
2	2 x nRC	repeat sub-loop 0 pattern until cycle (3 × nRC - 1); use BA = 02h and RA = 05555h instead				
3	3 x nRC	repeat sub-loop 0 pattern until cycle (4 × nRC - 1); use BA = 07h and RA = 02AAAh instead				
4	4 x nRC	repeat sub-loop 0 pattern until cycle (5 × nRC - 1); use BA = 01h and RA = 05555h instead				
5	5 x nRC	repeat sub-loop 0 pattern until cycle (6 × nRC - 1); use BA = 06h and RA = 02AAAh instead				
6	6 x nRC	repeat sub-loop 0 pattern until cycle (7 × nRC - 1); use BA = 03h and RA = 05555h instead				
7	7 x nRC	repeat sub-loop 0 pattern until cycle (8 × nRC - 1); use BA = 04h and RA = 02AAAh instead				
8 to 15	for 16-bank devices: repeat sub-loops 0 to 7 pattern; use BA3 = 1 instead, maintain SID[1:0] = 00					
16 to 31	for 8-High,12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 01 instead					
32 to 47	for 12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 10 instead					
48 to 63	for 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 11 instead					
Note 1	ACT is a 1.5 cycle command. The falling edge of the second cycle is a RNOP					

9.1 IDD and IPP Specification Parameters and Test Conditions (cont'd)

Table 97 — IDD4R Measurement-Loop Pattern

Sub-Loop	Cycle Number	Row Command	Column Command	Row Bank Address (BA[3:0])	Col. Bank Address (BA[3:0])	Row Address (RA[13:0])	Col. Address (CA[4:0])	Data Pattern (1 Byte)
0	0	RNOP	READ – PC0	02h	00h	05555h	0Ah	Pattern A
	1	RNOP	READ – PC1	02h	00h	05555h	0Ah	Pattern A
	2	RNOP	READ – PC0	02h	04h	05555h	15h	Pattern B
	3	RNOP	READ – PC1	02h	04h	05555h	15h	Pattern B
	4	RNOP	READ – PC0	02h	08h	05555h	1Ah	Pattern A
	5	RNOP	READ – PC1	02h	08h	05555h	1Ah	Pattern A
	6	RNOP	READ – PC0	02h	0Ch	05555h	05h	Pattern B
	7	RNOP	READ – PC1	02h	0Ch	05555h	05h	Pattern B
	8	RNOP	READ – PC0	06h	04h	02AAAh	0Ah	Pattern A
	9	RNOP	READ – PC1	06h	04h	02AAAh	0Ah	Pattern A
	10	RNOP	READ – PC0	06h	08h	02AAAh	15h	Pattern B
	11	RNOP	READ – PC1	06h	08h	02AAAh	15h	Pattern B
	12	RNOP	READ – PC0	06h	0Ch	02AAAh	1Ah	Pattern A
	13	RNOP	READ – PC1	06h	0Ch	02AAAh	1Ah	Pattern A
	14	RNOP	READ – PC0	06h	00h	02AAAh	05h	Pattern B
	15	RNOP	READ – PC1	06h	00h	02AAAh	05h	Pattern B
	16	RNOP	READ – PC0	0Ah	08h	05555h	0Ah	Pattern A
	17	RNOP	READ – PC1	0Ah	08h	05555h	0Ah	Pattern A
	18	RNOP	READ – PC0	0Ah	0Ch	05555h	15h	Pattern B
	19	RNOP	READ – PC1	0Ah	0Ch	05555h	15h	Pattern B
	20	RNOP	READ – PC0	0Ah	00h	05555h	1Ah	Pattern A
	21	RNOP	READ – PC1	0Ah	00h	05555h	1Ah	Pattern A
	22	RNOP	READ – PC0	0Ah	04h	05555h	05h	Pattern B
	23	RNOP	READ – PC1	0Ah	04h	05555h	05h	Pattern B
	24	RNOP	READ – PC0	0Eh	0Ch	02AAAh	0Ah	Pattern A
	25	RNOP	READ – PC1	0Eh	0Ch	02AAAh	0Ah	Pattern A
	26	RNOP	READ – PC0	0Eh	00h	02AAAh	15h	Pattern B
	27	RNOP	READ – PC1	0Eh	00h	02AAAh	15h	Pattern B
	28	RNOP	READ – PC0	0Eh	04h	02AAAh	1Ah	Pattern A
	29	RNOP	READ – PC1	0Eh	04h	02AAAh	1Ah	Pattern A
	30	RNOP	READ – PC0	0Eh	08h	02AAAh	05h	Pattern B
	31	RNOP	READ – PC1	0Eh	08h	02AAAh	05h	Pattern B
1		repeat sub-loop 0 pattern; use RBA0 = 1 and CBA0 = 1 instead, maintain SID[1:0]=00						
2		repeat sub-loop 0 and 1 pattern; use RBA1 = 0 and CBA1 = 1 instead, maintain SID[1:0]=00 RNOP, CNOP 2-cycle before SID change to meet tCCDR						
3		for 8-High,12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 01 instead						
4		for 12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 10 instead						
5		for 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 11 instead						

Note 1 Pattern A for a Byte (BL0 to BL7) : 00h, 3Ch, F0h, 69h, AAh, FFh, C3h, 55h
Note 2 Pattern B for a Byte (BL0 to BL7) : F0h, A5h, 96h, 0Fh, C3h, 55h, CCh, 5Ah
Note 3 Vendors may provide IDD4/7 values based on worse toggle patterns reflecting internal bus architecture

Table 98 — IDD4W Measurement-Loop Pattern

Sub-Loop	Cycle Number	Row Command	Column Command	Row Bank Address (BA[3:0])	Col. Bank Address (BA[3:0])	Row Address (RA[13:0])	Col. Address (CA[4:0])	Data Pattern (1 Byte)
0	0	RNOP	WRITE – PC0	02h	00h	05555h	0Ah	Pattern A
	1	RNOP	WRITE – PC1	02h	00h	05555h	0Ah	Pattern A
	2	RNOP	WRITE – PC0	02h	04h	05555h	15h	Pattern B
	3	RNOP	WRITE – PC1	02h	04h	05555h	15h	Pattern B
	4	RNOP	WRITE – PC0	02h	08h	05555h	1Ah	Pattern A
	5	RNOP	WRITE – PC1	02h	08h	05555h	1Ah	Pattern A
	6	RNOP	WRITE – PC0	02h	0Ch	05555h	05h	Pattern B
	7	RNOP	WRITE – PC1	02h	0Ch	05555h	05h	Pattern B
	8	RNOP	WRITE – PC0	06h	04h	02AAAh	0Ah	Pattern A
	9	RNOP	WRITE – PC1	06h	04h	02AAAh	0Ah	Pattern A
	10	RNOP	WRITE – PC0	06h	08h	02AAAh	15h	Pattern B
	11	RNOP	WRITE – PC1	06h	08h	02AAAh	15h	Pattern B
	12	RNOP	WRITE – PC0	06h	0Ch	02AAAh	1Ah	Pattern A
	13	RNOP	WRITE – PC1	06h	0Ch	02AAAh	1Ah	Pattern A
	14	RNOP	WRITE – PC0	06h	00h	02AAAh	05h	Pattern B
	15	RNOP	WRITE – PC1	06h	00h	02AAAh	05h	Pattern B
	16	RNOP	WRITE – PC0	0Ah	08h	05555h	0Ah	Pattern A
	17	RNOP	WRITE – PC1	0Ah	08h	05555h	0Ah	Pattern A
	18	RNOP	WRITE – PC0	0Ah	0Ch	05555h	15h	Pattern B
	19	RNOP	WRITE – PC1	0Ah	0Ch	05555h	15h	Pattern B
	20	RNOP	WRITE – PC0	0Ah	00h	05555h	1Ah	Pattern A
	21	RNOP	WRITE – PC1	0Ah	00h	05555h	1Ah	Pattern A
	22	RNOP	WRITE – PC0	0Ah	04h	05555h	05h	Pattern B
	23	RNOP	WRITE – PC1	0Ah	04h	05555h	05h	Pattern B
	24	RNOP	WRITE – PC0	0Eh	0Ch	02AAAh	0Ah	Pattern A
	25	RNOP	WRITE – PC1	0Eh	0Ch	02AAAh	0Ah	Pattern A
	26	RNOP	WRITE – PC0	0Eh	00h	02AAAh	15h	Pattern B
	27	RNOP	WRITE – PC1	0Eh	00h	02AAAh	15h	Pattern B
	28	RNOP	WRITE – PC0	0Eh	04h	02AAAh	1Ah	Pattern A
	29	RNOP	WRITE – PC1	0Eh	04h	02AAAh	1Ah	Pattern A
	30	RNOP	WRITE – PC0	0Eh	08h	02AAAh	05h	Pattern B
	31	RNOP	WRITE – PC1	0Eh	08h	02AAAh	05h	Pattern B
1		repeat sub-loop 0 pattern; use RBA0 = 1 and CBA0 = 1 instead, maintain SID[1:0]=00						
2		repeat sub-loop 0 and 1 pattern; use RBA1 = 0 and CBA1 = 1 instead, maintain SID[1:0]=00						
3		for 8-High,12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 01 instead						
4		for 12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 10 instead						
5		for 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 11 instead						

Note 1 Pattern A for a Byte (BL0 to BL7) : 00h, 3Ch, F0h, 69h, AAh, FFh, C3h, 55h
Note 2 Pattern B for a Byte (BL0 to BL7) : F0h, A5h, 96h, 0Fh, C3h, 55h, CCh, 5Ah
Note 3 Vendors may provide IDD4/7 values based on worse toggle patterns reflecting internal bus architecture

Table 99 — IDD5P Measurement-Loop Pattern

Sub-Loop	Cycle Number	Row Command	Column Command	Row Bank Address (SID[1:0], BA[3:0])
0	0	PER-BANK REFRESH – PC0	CNOP	00h, 00h
	1	RNOP	CNOP	N/A
	2	PER-BANK REFRESH – PC1	CNOP	00h, 00h
	3	RNOP	CNOP	N/A
	…	repeat RNOP until cycle (nRREFD-1)		
1	nRREFD	repeat sub-loop 0 pattern until cycle (2 x nRREFD-1); use BA=05h instead		
2	2 x nRREFD	repeat sub-loop 0 pattern until cycle (3 x nRREFD-1); use BA=02h instead		
3	3 x nRREFD	repeat sub-loop 0 pattern until cycle (4 x nRREFD-1); use BA=07h instead		
4	4 x nRREFD	repeat sub-loop 0 pattern until cycle (5 x nRREFD-1); use BA=01h instead		
5	5 x nRREFD	repeat sub-loop 0 pattern until cycle (6 x nRREFD-1); use BA=06h instead		
6	6 x nRREFD	repeat sub-loop 0 pattern until cycle (7 x nRREFD-1); use BA=03h instead		
7	7 x nRREFD	repeat sub-loop 0 pattern until cycle (8 x nRREFD-1); use BA=04h instead		
	repeat sub-loops 0 to 7 pattern; use BA3=1 instead			
	for 4-High devices: wait for (nRFCpb - 15 x nRREFD) for all other devices: repeat sub-loops 0 to 15 pattern; use SID[1:0] = 01h instead			
	for 12-High and 16-High devices: repeat sub-loops 0 to 15 pattern; use SID[1:0] = 02h instead			
	for 16-High devices: repeat sub-loops 0 to 15 pattern; use SID[1:0] = 03h instead			
Note 1 nRFCpb = RU(tRFCpb/tCK), nRREFD = RU(tRREFD/tCK). If not already an integer, round up to the next integer.				

Table 100 — IDD7 Measurement-Loop Pattern

Sub-Loop	Cycle Number	Row Command	Column Command	Row Bank Address (BA[3:0])	Col. Bank Address (BA[3:0])	Row Address (RA[13:0])	Col. Address (CA[4:0])	Data Pattern (1 Byte)
0	0	ACT – PC0	READ – PC0	02h	00h	05555h	0Ah	Pattern A
	1		READ – PC1		00h		0Ah	Pattern A
	2	ACT – PC1	READ – PC0	02h	04h	05555h	15h	Pattern B
	3		READ – PC1		04h		15h	Pattern B
	4	RNOP	READ – PC0	02h	08h	05555h	1Ah	Pattern A
	5	RNOP	READ – PC1	02h	08h	05555h	1Ah	Pattern A
	6	RNOP	READ – PC0	02h	0Ch	05555h	05h	Pattern B
	7	RNOP	READ – PC1	02h	0Ch	05555h	05h	Pattern B
	8	ACT – PC0	READ – PC0	06h	04h	02AAAh	0Ah	Pattern A

	9		READ – PC1		04h		0Ah	Pattern A
	10	ACT – PC1	READ – PC0	06h	08h	02AAAh	15h	Pattern B
	11		READ – PC1		08h		15h	Pattern B
	12	RNOP	READ – PC0	06h	0Ch	02AAAh	1Ah	Pattern A
	13	RNOP	READ – PC1	06h	0Ch	02AAAh	1Ah	Pattern A
	14	RNOP	READ – PC0	06h	00h	02AAAh	05h	Pattern B
	15	RNOP	READ – PC1	06h	00h	02AAAh	05h	Pattern B
	16	ACT – PC0	READ – PC0	0Ah	08h	05555h	0Ah	Pattern A
	17		READ – PC1		08h		0Ah	Pattern A
	18	ACT – PC1	READ – PC0	0Ah	0Ch	05555h	15h	Pattern B
	19		READ – PC1		0Ch		15h	Pattern B
	20	RNOP	READ – PC0	0Ah	00h	05555h	1Ah	Pattern A
	21	RNOP	READ – PC1	0Ah	00h	05555h	1Ah	Pattern A
	22	RNOP	READ – PC0	0Ah	04h	05555h	05h	Pattern B
	23	RNOP	READ – PC1	0Ah	04h	05555h	05h	Pattern B
	24	ACT – PC0	READA – PC0	0Eh	0Ch	02AAAh	0Ah	Pattern A
	25		READA – PC1		0Ch		0Ah	Pattern A
	26	ACT – PC1	READA – PC0	0Eh	00h	02AAAh	15h	Pattern B
	27		READA – PC1		00h		15h	Pattern B
	28	RNOP	READA – PC0	0Eh	04h	02AAAh	1Ah	Pattern A
	29	RNOP	READA – PC1	0Eh	04h	02AAAh	1Ah	Pattern A
	30	RNOP	READA – PC0	0Eh	08h	02AAAh	05h	Pattern B
	31	RNOP	READA – PC1	0Eh	08h	02AAAh	05h	Pattern B
1		repeat sub-loop 0 pattern; use RBA0 = 1 and CBA0 = 1 instead, maintain SID[1:0]=00						
2		repeat sub-loop 0 and 1 pattern; use RBA1 = 0 and CBA1 = 1 instead, maintain SID[1:0]=00 RNOP, CNOP 2-cycle before SID change to meet tCCDR						
3		for 8-High,12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 01 instead						
4		for 12-High and 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 10 instead						
5		for 16-High devices: repeat sub-loops 0,1 and 2 pattern; use SID[1:0] = 11 instead						

Note 1 Pattern A for a Byte (BL0 to BL7) : 00h, 3Ch, F0h, 69h, AAh, FFh, C3h, 55h
Note 2 Pattern B for a Byte (BL0 to BL7) : F0h, A5h, 96h, 0Fh, C3h, 55h, CCh, 5Ah
Note 3 Vendors may provide IDD4/7 values based on worse toggle patterns reflecting internal bus architecture
Note 4 ACT is a 1.5 cycle command. The falling edge of the second cycle is a RNOP

9.2 IDD and IPP Specifications

IDD and IPP values are valid for the full operating range of voltage and temperature unless otherwise noted.

Table 101 — IDD and IPP Specification Example

Symbol	Speed Bin		Unit	Notes
	IDD (Max)	IPP (Max)		
IDD0			mA	
IDD2P			mA	
IDD2P0			mA	
IDD2N			mA	
IDD3P			mA	

IDD3P0			mA	
IDD3N			mA	
IDD4R			mA	
IDD4W			mA	
IDD5B			mA	
IDD5P			mA	
IDD6x	See Separate Table		mA	
IDD7			mA	
IDD8			mA	

9.3 IDD6 Specification

Table 102 — IDD6 Specification

Symbol	Temperature Range	Value	Unit	Notes
IDD6N	0ºC - T_N		mA	2, 3, 7
IDD6E (Optional)	0ºC - T_E		mA	1, 3, 4, 7
IDD6R (Optional)	0ºC - T_R		mA	3, 5, 7
IDD6A (Optional)	0ºC - T_a		mA	3, 5, 5, 6
	T_b - T_y (optional)		mA	3, 5, 5, 6
	T_z - $T_{OPERmax}$		mA	3, 5, 5, 6, 8

NOTE 1 Max. values for IDD currents considering worst case conditions of process, temperature and voltage.
NOTE 2 Applicable for MR0 settings OP2=0.
NOTE 3 Supplier data sheets include a max value.
NOTE 4 IDD6E is only specified for devices which support the Extended Temperature Range feature.
NOTE 5 IDD6A is only specified for devices which support the Temperature Controlled Self Refresh feature enabled by MR0 with OP2=1.
NOTE 6 The number of discrete temperature ranges supported and the associated T_a - T_z, and $T_{OPERmax}$ 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 7 T_R represents the temperature used to reflect the current consumed in a typical room temperature environment.
NOTE 8 $T_{OPERmax}$ represents the max temperature supported by the DRAM when TCSR is enabled.

10 AC Timings

Table 103 — Timings Parameters

Parameter	Symbol	Speed Bin[2]										Unit	Notes
		4.8 Gbps/pin		5.2 Gbps/pin		5.6 Gbps/pin		6.0 Gbps/pin		6.4 Gbps/pin			
		min	Max	min	Max	min	Max	min	Max	min	Max		
CK Timings													
CK clock frequency	f_{CK}	50	1200	50	1300	50	1400	50	1500	50	1600	MHz	
CK clock period	t_{CK}	0.833	20	0.769	20	0.714	20	0.667	20	0.625	20	ns	4
Absolute CK clock differential HIGH-level width	t_{CH}	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	t_{CK}	40
Absolute CK clock differential LOW-level width	t_{CL}	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	t_{CK}	40
Command and Address Input Timings													
Command and address input setup time based on VIH/VIL	t_{IS}	92		85		79		73		69		ps	5
Command and address input hold time based on VIH/VIL	t_{IH}	92		85		79		73		69		ps	5
Command and address single pulse width	t_{CIPW}	292		269		250		233		219		ps	27
Data Input Timings													
WDQS clock period	t_{WDQS}	0.416	10	0.385	10	0.357	10	0.333	10	0.312	10	ns	41
Average WDQS differential input HIGH pulse width	$t_{WQSH(avg)}$	0.47	0.53	0.47	0.53	0.47	0.53	0.47	0.53	0.47	0.53	t_{WDQS}	42
Average WDQS differential input LOW pulse width	$t_{WQSL(avg)}$	0.47	0.53	0.47	0.53	0.47	0.53	0.47	0.53	0.47	0.53	t_{WDQS}	42
Absolute WDQS differential input HIGH pulse width	$t_{WQSH(abs)}$	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	t_{WDQS}	43
Absolute WDQS differential input LOW pulse width	$t_{WQSL(abs)}$	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	0.45	0.55	t_{WDQS}	43
Rx single pulse width	t_{DIPW}	0.55		0.55		0.55		0.55		0.55		UI	27
Rx Timing Window with PSIJ	t_{DIVW}	0.30		0.30		0.30		0.30		0.30		UI	
WDQS to write data offset	$t_{WDQS2DQ\_I}$	150	900	150	900	150	900	150	900	150	900	ps	33,34,45
WDQS to write data offset voltage variation for write	$t_{WDQS2DQ\_I\_VOLT}$		0.8		0.8		0.8		0.8		0.8	ps/mV	34,46
WDQS to write data offset temperature variation for write	$t_{WDQS2DQ\_I\_TEMP}$		0.4		0.4		0.4		0.4		0.4	ps/C	34,46,47

Table 98 — Timings Parameters (cont'd)

Parameter	Symbol	Speed Bin[2]										Unit	Notes
		4.8 Gbps/pin		5.2 Gbps/pin		5.6 Gbps/pin		6.0 Gbps/pin		6.4 Gbps/pin			
		min	Max	min	Max	min	Max	min	Max	min	Max		
Data Input Timings (cont'd)													
DQ to DQ skew (intra-byte) for write	$t_{DQ2DQtra\_I}$		10		10		10		10		10	ps	
DQ to DQ skew (inter-byte) for write	$t_{DQ2DQter\_I}$		40		40		40		40		40	ps	
Data Output Timings													
RDQS differential output HIGH	t_{QSH}	$t_{WQSH(abs)}$ - 0.08		$t_{WQSH(abs)}$ - 0.08		$t_{WQSH(abs)}$ - 0.08		$t_{WQSH(abs)}$ - 0.08		$t_{WQSH(abs)}$ - 0.08		t_{WDQS}	6,38
RDQS differential output LOW	t_{QSL}	$t_{WQSL(abs)}$ - 0.08		$t_{WQSL(abs)}$ - 0.08		$t_{WQSL(abs)}$ - 0.08		$t_{WQSL(abs)}$ - 0.08		$t_{WQSL(abs)}$ - 0.08		t_{WDQS}	6,39
DQ output hold time from DQS	t_{QH}	Min(t_{QSH}, t_{QSL})		Min(t_{QSH}, t_{QSL})		Min(t_{QSH}, t_{QSL})		Min(t_{QSH}, t_{QSL})		Min(t_{QSH}, t_{QSL})		t_{WDQS}	6
DQ output window per pin	t_{QW}	Min(t_{QSH}, t_{QSL}) - 0.07		Min(t_{QSH}, t_{QSL}) - 0.07		Min(t_{QSH}, t_{QSL}) - 0.07		Min(t_{QSH}, t_{QSL}) - 0.07		Min(t_{QSH}, t_{QSL}) - 0.07		UI	6
RDQS to DQ skew in Byte T4	$t_{DQSQtra}$		20		20		20		20		20	ps	6
DQ to DQ skew (intra-byte) for read	$t_{DQ2DQtra\_O}$		10		10		10		10		10	ps	6
DQ to DQ skew (inter-byte) for read	$t_{DQ2DQter\_O}$		30		30		30		30		30	ps	6
WDQS to read data and RDQS offset	$t_{WDQS2DQ\_O}$	0.5	2.5	0.5	2.5	0.5	2.5	0.5	2.5	0.5	2.5	ns	6,33,35
WDQS to read data offset voltage variation for read	$t_{WDQS2DQ\_O\_VOLT}$		2.5		2.5		2.5		2.5		2.5	ps/mV	35,46
WDQS to read data offset temperature variation for read	$t_{WDQS2DQ\_O\_TEMP}$		1.0		1.0		1.0		1.0		1.0	ps/C	35,46,47
DQ, DBI high impedance to low impedance time from WDQS	t_{LZ}	Min: $t_{WDQS2DQ\_O}$(min) – t_{QH}(min) Max: $t_{WDQS2DQ\_O}$(max) + $t_{DQSQtra}$(max)										ns	6
DQ, DBI low impedance to high impedance time from WDQS	t_{HZ}	Min: $t_{WDQS2DQ\_O}$(min) Max: $t_{WDQS2DQ\_O}$(max) + $t_{DQSQtra}$(max)										ns	6

10 AC Timings (cont'd)

Table 104 — Timings Parameters (Part 2)

Parameter[1,3]	Symbol	Values		Unit	Notes
		MIN	MAX		
Row Access Timings					
ACTIVATE to ACTIVATE command period	t_{RC}		-	ns	
ACTIVATE to PRECHARGE command period	t_{RAS}		9 x t_{REFI}	ns	7
ACTIVATE to READ command delay	t_{RCDRD}		-	ns	
ACTIVATE to WRITE command delay	t_{RCDWR}		-	ns	
ACTIVATE to ACTIVATE or PER BANK REFRESH bank B command delay same bank group	t_{RRDL}		-	ns	8
ACTIVATE to ACTIVATE or PER BANK REFRESH bank B command delay different bank group	t_{RRDS}		-	ns	9
Four bank activate window	t_{FAW}		-	ns	10
READ to PRECHARGE command delay same bank	t_{RTP}		-	nCK	11, 32
PRECHARGE command period	t_{RP}		-	ns	
WRITE recovery time	t_{WR}		-	ns	32
Auto precharge write recovery + precharge time	t_{DAL}	-	-	nCK	12
PRECHARGE to PRECHARGE delay same pseudo channel	t_{PPD}	2		nCK	
Rolling Accumulated ACTIVATE count	RAA			-	
Column Access Timings					
RD/WR bank A to RD/WR bank B command delay same bank group	t_{CCDL}	Max (4, 2.5 ns/t_{CK})	-	nCK	13, 14
RD/WR bank A to RD/WR bank B command delay different bank group	t_{CCDS}	2	-	nCK	15, 16
RD SID A to RD SID B command delay	t_{CCDR}		-	nCK	17
Internal WRITE to READ command delay same bank group	t_{WTRL}		-	nCK	13
Internal WRITE to READ command delay different bank group	t_{WTRS}		-	nCK	15
READ to WRITE command delay	t_{RTW}		-	ns	18
Power-Down Timings					
POWER-DOWN ENTRY to EXIT time	t_{PD}	t_{CPDED} + 6 x t_{CK}	9 x t_{REFI}	ns	
POWER-DOWN EXIT time	t_{XP}	MAX(10 x t_{CK}, 7.5)	-	ns	
Valid CK clocks required after POWER-DOWN ENTRY	t_{CKPDE}	RU(t_{CPDED} / t_{CK}) + 1		nCK	
Valid CK clocks required before POWER-DOWN EXIT	t_{CKPDX}	5		nCK	
Command path disable delay	t_{CPDED}	MAX(10 x t_{CK}, 7.5)	-	ns	
ACTIVATE to POWER-DOWN ENTRY command delay	t_{ACTPDE}	1	-	nCK	19
PRECHARGE(rising CK edge) to POWER-DOWN ENTRY command delay	t_{PRPDER}	1	-	nCK	
PRECHARGE(falling CK edge) to POWER-DOWN ENTRY command delay	t_{PRPDEF}	1.5	-	nCK	
REFRESH to POWER-DOWN ENTRY command delay	t_{REFPDE}	1	-	nCK	19
PER BANK REFRESH to POWER-DOWN ENTRY command delay	$t_{REFPBPDE}$	1	-	nCK	19

Parameter[1,3]	Symbol	Values		Unit	Notes
		MIN	MAX		

Power-Down Timings (cont'd)								
MODE REGISTER SET to POWER-DOWN ENTRY command delay				t_{MRSPDE}	t_{MOD}(min)	-	nCK	
READ or READ w/ AP to POWER-DOWN ENTRY command delay				t_{RDPDE}	RL + PL + 2 + RU(t_{DQSS}(max) + $t_{WDQS2DQ\_O}$(max) / t_{CK})	-	nCK	
WRITE to POWER-DOWN ENTRY command delay				t_{WRPDE}	WL + PL + 3 + RU(t_{WR} / t_{CK})	-	nCK	20
WRITE w/ AP to POWER-DOWN ENTRY command delay				t_{WRAPDE}	WL + PL + 3 + WR	-	nCK	21
Self Refresh Timings								
SELF REFRESH ENTRY to EXIT time				t_{CKSR}	t_{CPDED} + 6 x t_{CK}	-	ns	
Valid CK clocks required after SELF REFRESH ENTRY				t_{CKSRE}	RU(t_{CPDED} / t_{CK}) + 1		nCK	
Valid CK clocks required before SELF REFRESH or POWER-DOWN EXIT				t_{CKSRX}	5		nCK	
READ or READ w/ AP to SELF REFRESH ENTRY command delay				t_{RDSRE}	RL + PL + 3	-	nCK	
Exit self refresh command delay				t_{XS}	MAX(10 x t_{CK}, t_{RFC}(min) + 10)	-	ns	
Exit self refresh to MODE REGISTER SET command delay				t_{XSMRS}	MAX(10 x t_{CK}, 15)	-	ns	
Exit self refresh to MODE REGISTER SET command delay after frequency change				t_{XSMRSF}		-	ns	
Refresh Timings								
Minimum time in self refresh for per-bank RAA count to be reset to 0				t_{RAASRF}		-	ns	29
REFRESH command period	24 Gb/die	4-High	6 Gb / channel	t_{RFCab}	360		ns	22
		8-High	12 Gb / channel		410	-		
		12-High	18 Gb / channel		450	-		
		16-High	24 Gb / channel		490	-		
	32 Gb/die	4-High	8 Gb / channel		400			
		8-High	16 Gb / channel		450	-		
		12-High	24 Gb / channel		490			
		16-High	32 Gb / channel		530	-		

Parameter[1,3]		Symbol	Values		Unit	Notes
			MIN	MAX		
Refresh Timings (cont'd)						
PER BANK REFRESH command period (same bank)	24 Gb / die	t_{RFCpb}	240	-	ns	28
	32 Gb / die		280	-		
PER BANK REFRESH command period (different bank) and PER BANK REFRESH to ACTIVATE (different bank) command delay		t_{RREFD}	MAX(3 x t_{CK}, 8)	-	ns	
Average periodic refresh interval for REFRESH command		t_{REFI}	-	3.9	µs	23
Average periodic refresh interval for PER BANK REFRESH command	8-High	t_{REFIpb}	-	t_{REFI} / 32	µs	22, 24
	12-High		-	t_{REFI} / 48	µs	
	16-High		-	t_{REFI} / 64	µs	
Secondary ACTIVATE command with DRFM flag to PRECHARGE command delay		$t_{DRFM2PRE}$	-	TBD	ns	
Time interval during which any Row/Bank address combination may be sampled for DRFM		t_{DRFMi}	-	2 x t_{REFI}	µs	
WDQS-to-CK Timings						
WDQS/2 (0° phase) rising edge to CK rising edge delay		t_{DQSS}	Max (-200ps, -0.2tCK)	Min (200ps, 0.2tCK)	ps/ tCK	
WDQS-to-CK phase search range during WDQS-to-CK alignment training		$t_{WDQS2CK}$	-0.4	0.4	tCK	
CK clock to phase detector output delay in WDQS-to-CK alignment training mode		$t_{WDQS2PD}$			ns	
Miscellaneous Timings						
MODE REGISTER SET command update delay		t_{MOD}		-	nCK	
MODE REGISTER SET command cycle time		t_{MRD}		-	nCK	
MODE REGISTER SET command from a preceding READ command		t_{RDMRS}	RL + PL + 2 + RU(t_{DQSS}(max) + $t_{WDQS2DQ\_O}$ (max)/t_{CK})		nCK	
Internal WRITE to MODE REGISTER SET command delay		t_{WRMRS}	MAX(RU(t_{WR} + t_{RP})/t_{CK}), (PL + RU(t_{PARDQ}/t_{CK}), 6)	-	nCK	
Interval VREFD offset single step settling time		t_{VREFD}		-	ns	
Interval VREFD offset full range settling time		t_{FVREFD}		-	ns	
ADD/CMD parity error output delay		t_{PARAC}			ns	25
Write data parity error output delay		t_{PARDQ}			ns	26
Write preamble for WDQS		t_{WPRE1}	4		tWDQS	
Read preamble for WDQS		t_{WPRE2}	16		tWDQS	
Write postamble for WDQS		t_{WPST1}	2		tWDQS	
Read postamble for WDQS		t_{WPST2}	4		tWDQS	
Read preamble for RDQS		t_{RPRE}	2		tWDQS	
Read postamble for RDQS		t_{RPST}	2		tWDQS	

CK clock frequency with DCA enabled	f_{CKDCA}	1200	-	MHz	36
Duty Cycle Monitor Measurement time	t_{DCMM}	1	-	µs	37

Parameter[1,3]	Symbol	Values		Unit	Notes
		MIN	MAX		

NOTE 1 AC timing parameters apply to each channel of the HBM4 device independently. No timing parameters are specified across channels, and all channels operate independently of each other.

NOTE 2 Speed bins are shown as examples. Vendors may define different speed bins; in this case it is recommended to scale the values for the related timing parameters.

NOTE 3 All parameters assume proper device initialization.

NOTE 4 Parameter t_{CK} is calculated as the average clock period across any consecutive 1,000 cycle window, where each clock period is calculated both from rising CK edge to rising CK edge, and falling CK edge to falling CK edge.

NOTE 5 Parameter is based on V_{IHCA} and V_{ILCA}.

NOTE 6 Parameter is measured with Output Timing reference load and Read DBI enabled.

NOTE 7 For Reads and Writes with auto precharge enabled, the device will hold off the internal precharge until $t_{RAS(min)}$ has been satisfied or the number of clock cycles as programmed for RAS in MR4 have elapsed.

NOTE 8 Parameter applies when consecutive commands access the same bank group.

NOTE 9 Parameter applies when consecutive commands access different bank groups.

NOTE 10 No more than 4 ACTIVATE or PER BANK REFRESH commands are allowed within t_{FAW} period.

NOTE 11 Parameter applies when READ and PRECHARGE commands access the same bank.

NOTE 12 $t_{DAL} = (t_{WR}/t_{CK}) + (t_{RP}/t_{CK})$. For each of the terms, if not already an integer, round up to the next integer.

NOTE 13 Parameter applies consecutive commands access the same bank group.

NOTE 14 t_{CCDL} parameter is applied when seamless consecutive Write or Read commands access to the banks in the same bank group.

NOTE 15 Parameter applies when consecutive commands access different bank groups.

NOTE 16 t_{CCDS} is either for seamless consecutive READ or seamless consecutive WRITE commands.

NOTE 17 t_{CCDR} is a parameter for 8,12,16-High HBM devices that is used for seamless consecutive READ commands between different stack IDs (SID) instead of t_{CCDS}. The t_{CCDR}(min) value is vendor specific and a range of t_{CCDS} + 1 to 2nCK is supported. The t_{CCDR}(min) is dependent on the operation frequency. The vendor datasheet should be consulted for details. For seamless WRITE commands the normal t_{CCDS} parameter applies. t_{CCDR} does not apply to DWORD MISR operations when DWORD Loopback is enabled in MR7.

NOTE 18 t_{RTW} is not a DRAM device limit but determined by the system bus turnaround time. Avoid bus contention by setting t_{RTW} (min) = (RL + BL/4 - WL + 0.5) × t_{CK} + $t_{WDQS2DQ\_O}$(max) - $t_{WDQS2DQ\_I}$(min) , and round up to the next integer.

NOTE 19 Upon entering power-down the CK clock may be stopped after the number of clock cycles as programmed for RAS in MR4.

NOTE 20 t_{WR} is defined in ns. For calculation of t_{WRPDE} round up t_{WR}/t_{CK} to the next integer.

NOTE 21 WR in clock cycles as programmed in MR3.

NOTE 22 Density is given per channel.

NOTE 23 A maximum of 8 consecutive REFRESH commands can be posted to an HBM4 device, meaning that the maximum absolute interval between any REFRESH command and the next REFRESH command is 9 × t_{REFI}.

NOTE 24 $t_{REFIPB} = t_{REFI} / N$; N = no. of banks.

NOTE 25 t_{PARAC} may be specified as an analog delay or as a combination of n clock cycles and an analog delay. The nominal AERR HIGH time in case of a parity error is 1 nCK.

NOTE 26 t_{PARDQ} may be specified as an analog delay or as a combination of n clock cycles and an analog delay. The nominal DERR HIGH time in case of a parity error is 1 nCK.

NOTE 27 t_{CIPW} is based on V_{REFCA} level, and t_{DIPW} is based on V_{REFDQ} level.

Parameter[1,3]	Symbol	Values		Unit	Notes
		MIN	MAX		

NOTE 28 Density is given per die.

NOTE 29 Parameter applies only to HBM4 DRAMs that require the use of Refresh Management (RFM).

NOTE 32 PRECHARGE and PRECHARGE ALL commands can be issued on a rising or a falling CK edge. For corresponds to the internal WR or RTP, add 0.5 t_{CK} to the number of clock cycles defined for RTP with reference to t_{RTP} or the number of clock cycles calculated to WR using RU(t_{WR}/t_{CK}).

NOTE 33 PVT variation is included.

NOTE 34 The minimum-to-maximum range does not exceed 400ps. The vendor's datasheet shall be consulted for the minimum and maximum values.

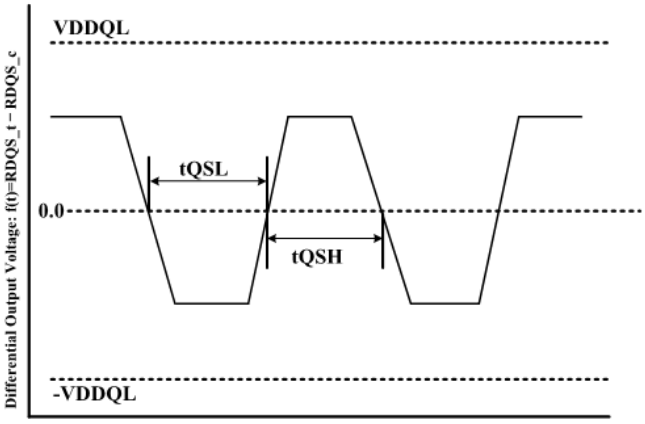
NOTE 35 The minimum-to-maximum range does not exceed 1.5ns. The vendor's datasheet shall be consulted for the minimum and maximum values.

NOTE 36 Parameter f_{CKDCA} applies when a duty correction code other than the default 0000 is programmed in the mode register.

NOTE 37 t_{DCMM} is measured from the MRS command that enables the duty cycle measurement until the measurement result in valid.

NOTE 38 t_{QSH} describes the instantaneous differential output high pulse width on RDQS_t – RDQS_c as it measures the next falling edge from an arbitrary rising edge. t_{QSH} edge measurement is based on zero voltage.

NOTE 39 t_{QSL} describes the instantaneous differential output low pulse width on RDQS_t – RDQS_c as it measures the next rising edge from an arbitrary falling edge. t_{QSL} edge measurement is based on zero voltage.



NOTE 40 $t_{CH(abs)}$ is the absolute instantaneous clock high pulse width, as measured from one rising edge to the following falling edge. $t_{CL(abs)}$ is the absolute instantaneous clock low pulse width, as measured from one falling edge to the following rising edge.

NOTE 41 Parameter t_{WDQS} is calculated as the average write clock period across any consecutive 200 cycle window, where each clock period is calculated both from rising CK edge to rising CK edge, and falling CK edge to falling CK edge.

NOTE 42 $t_{WQSH(avg)}$ is defined as the average high pulse width, as calculated across any consecutive 200 high pulses. $t_{WQSL(avg)}$ is defined as the average low pulse width, as calculated across any consecutive 200 low pulses.

NOTE 43 $t_{WQSH(abs)}$ is the absolute instantaneous write clock high pulse width, as measured from one rising edge to the following falling edge. $t_{WQSL(abs)}$ is the absolute instantaneous write clock low pulse width, as measured from one falling edge to the following rising edge.

NOTE 45 $t_{WDQS2DQ\_I}$ max delay variation as a function of the DC voltage variation for VDDQ. It includes VDDQ AC noise impact for frequencies > 20MHz and max voltage of TBD mVpk-pk from DC to 20 MHz at a fixed temperature on the package.

NOTE 46 Actual values could be positive or negative numbers depending on design implementation and process.

NOTE 47 The parameter is referenced to IEEE1500 TEMPERATURE readout. See 13.5.12 TEMPERATURE.

11 Package (Die) Specification

11.1 Signals

Table 105 — I/O Signal Description

Signals	Type	Description
CK[31:0]_t, CK[31:0]_c	Input	Clock: CK_t and CK_c are differential clock inputs. Row and column command and address inputs are latched on the rising and falling edges of CK.
C[31:0]_[7:0]	Input	Column command and address: the command code, bank and column address for Write and Read operations and the mode register address and code to be loaded with MODE REGISTER SET commands are received on the C[7:0] inputs.
R[31:0]_[9:0]	Input	Row command and address: the command code, bank and row address for Activate, Precharge and Refresh commands are received on the R[9:0] inputs.
ARFU[31:0]	Input	Reserved for future use: unused microbumps in AWORD.
APAR[31:0]	Input	Command / address parity: one parity signal per AWORD. APAR is associated with C[7:0], R[9:0] and ARFU.
DQ[31:0]_[63:0]	I/O	Data Input/Output: 64-bit data bus. DQ[31:0] represents the 32-bit data bus of PC0 and DQ[63:32] represents the 32-bit data bus of PC1.
DBI[31:0]_[7:0]	I/O	Data Bus Inversion: DBI0 is associated with DQ[7:0], DBI1 is associated with DQ[15:8], ... , and DBI7 is associated with DQ[63:56].
ECC[31:0]_[3:0]	I/O	ECC: ECC0, ECC1 are associated with DQ[31:0]. ECC2, ECC3 are associated with DQ[63:32].
SEV[31:0]_[3:0]	I/O	SEV: SEV0, SEV1 are associated with DQ[31:0]. SEV2, SEV3 are associated with DQ[63:32].
DPAR[31:0]_[1:0]	I/O	Data Parity: one data parity signal per DWORD. DPAR0 is associated with DQ[31:0] and DPAR1 is associated with DQ[63:32].
DERR[31:0]_[1:0]	Output	Data parity error: one data parity error bit per DWORD. DERR0 is associated with DQ[31:0] and DERR1 is associated with DQ[63:32].
AERR[31:0]	Output	Address parity error. One address parity error bit for row and column address and command per AWORD.
WDQS[31:0]_[1:0]_t, WDQS[31:0]_[1:0]_c	Input	Write Data Strobe: WDQS_t and WDQS_c are differential strobe inputs. One WDQS pair per DWORD. WDQS0 is associated with DQ[31:0] and WDQS1 is associated with DQ[63:32].
RDQS[31:0]_[1:0]_t, RDQS[31:0]_[1:0]_c	Output	Read Data Strobe: RDQS_t and RDQS_c are differential strobe outputs. Read output data are sent on the rising and falling edges of RDQS. One RDQS pair per DWORD. RDQS0 is associated with DQ[31:0] and RDQS1 is associated with DQ[63:32].
DA[39:0]	I/O	Direct Access Input/Output: These pins are provided for direct access test. They must be routed directly to an external package I/O pin. The function is defined by the memory vendor.

Signals	Type	Description
RESET_n	Input	Reset: RESET_n LOW asynchronously initiates a full chip reset of the HBM4 device.
NC		No connect pad: electrically isolated
WRCK	Input	IEEE-1500 Wrapper Serial Port Clock
WRST_n	Input	IEEE-1500 Wrapper Serial Port Reset
SelectWIR	Input	IEEE-1500 Wrapper Serial Port Instruction Register Select
ShiftWR	Input	IEEE-1500 Wrapper Serial Port Shift
CaptureWR	Input	IEEE-1500 Wrapper Serial Port Capture
UpdateWR	Input	IEEE-1500 Wrapper Serial Port Update
WSI	Input	IEEE-1500 Wrapper Serial Port Data
WSO[31:0]	Output	IEEE-1500 Wrapper Serial Port Data Out
RD[31:0]_[3:0]	I/O	Redundant microbumps in DWORD
RA[31:0]	Input	Redundant command and address microbump in AWORD
RM[1:0]	Output	Redundant WSO microbump in MIDSTACK
MRFU[3:0]		Reserved for future use, unused microbumps in mid-stack region
NOBUMP		Depopulated pad: reserved as test pad for probing
CATTRIP	Output	DRAM Catastrophic Temperature Report
VSS	Supply	Ground
VDDC, VDDQ, VPP, VDDQL	Supply	Power supply

NOTE 1 Index [31:0] represents the channel indicator “0” to “31” of the HBM device. Signal names including the channel indicators are used whenever more than one channel and/or pseudo channel is referenced, as e.g., with the HBM4Ballout. The channel indicators is omitted whenever features and functions common to all channels and/or all pseudo channels are described

NOTE 2 HBM4 devices supporting less than 32 channels are allowed to have input/output buffers physically present at the pins associated with the unavailable channels, however these input/output buffers will be disabled. The host shall leave those pins floating. The availability of each channel [31:0] has to be coded in IEEE1500 DEVICE_ID Wrapper Data Register bits [39:8].

NOTE 3 All power supply microbumps defined in Table 108 must be present and connected with their respective power nets even if the related channel is not present or marked non-working.

11.2 MicroBump Positions

The MicroBump array of the DRAM stack employs a staggered pattern as depicted in Figure 99 where a 'staggered' bump is located halfway between major row and column, hence its location is determined by X/2 and Y/2. Table 106 shows geometric parameters of the Staggered MicroBump pattern. Parameter PMin is the minimum bump pitch anywhere in the MicroBump field; for chosen X and Y parameters.

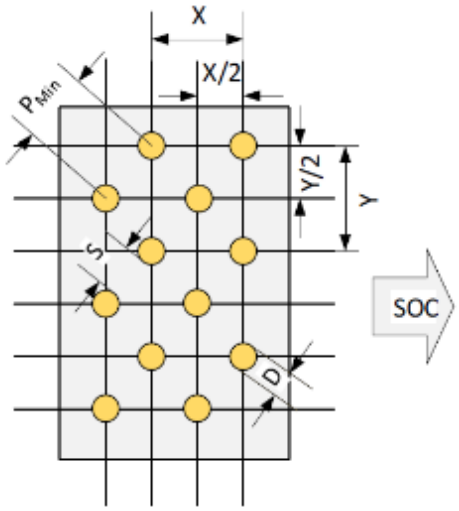


Figure 99 — Staggered MicroBump Pattern

Table 106 — Geometric Parameters of the Staggered MicroBump Pattern

Label	Nominal Value	Description
X	70 um	Horizontal pitch of two adjacent MicroBumps
Y	110 um	Vertical pitch of two adjacent MicroBumps
P_{Min}	65 um	Minimum pitch of the bump field
D	28 um	MicroBump diameter
S		Bump-to-bump air gap; S = P_{Min} - D

The HBM4 bump matrix is defined as shown in subsequent tables. Please refer to MO-362 for device dimensions.

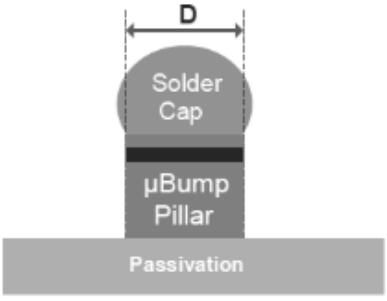


Figure 100 — MicroBump Pillar Diameter

11.2 MicroBump Positions (cont'd)

Footprint consists of 161 rows with a pitch of Y/2 and 308 columns with a pitch of X/2. The overall array size is (307 x X/2 + D) x (160 x Y/2 + D) = 10773.0 µm x 8828.0 µm. The ball matrix is center aligned with the die. The ball array center is the origin of the ball location coordinates. Ball A1 is located at the top left at X = -5372.5 µm, Y = +4400.0 µm.

11.3 HBM4 Device Dimensions

Table 107 — HBM4 Device Dimensions

Parameter	Symbol	Configuration	Minimum	Nominal	Maximum	Unit	Notes
Width	X	24Gb/die		12.775		mm	3
		32Gb/die		TBD			
Length	Y			10.975		mm	
Height	Z	8-High	750	775	800	µm	1, 2
		12-High	750	775	800	µm	
		16-High	750	775	800	µm	

NOTE 1 The configuration refers to the number of memory dies in the stack. The stack may include an additional base (interface) die.
NOTE 2 Refer to MO-362 for related package drawings.
NOTE 3 Refer to MO-362 for X and Y dimension min/max tolerances and related package drawings.

11.4 HBM4 Bump Map

A geographical overview of the HBM4 bump matrices is provided in Table 108 for Footprint, an overviewof the bump matrix in Figure 101 and the detailed bump matrix is provided in the HBM4 Ballout Spreadsheet, please refer to (tg423b3^20240117^1883.03^Micron^HBM4_final_bump_map_r1)

11.4 HBM4 Bump Map (cont'd)

Table 108 — HBM4 Bump Map Footprint – Geographical Overview (not to scale)

Columns	1, 2	3 … 62	63 … 85	86 … 94	95 … 177	178 … 188	189 … 202	203 … 216	217 … 230	231 … 244	…	251 … 264	265 … 278	279 … 292	293 … 306	307, 308
Rows A … K	Mechanical Bumps	Upper Left Edge Power Supply Region		Upper Center Edge Power Supply Region			Upper Right Edge Power Supply Region									Mechanical Bumps
L … AD		Power Supply Region	Left Depopulated Micropillar Area Dedicated for (Optional) Probe Pads	Direct Access Test Port	Center Supply Region	Right Depopluated Area	DWORD0 Channel 28	DWORD0 Channel 24	DWORD0 Channel 20	DWORD0 Channel 16		DWORD0 Channel 12	DWORD0 Channel 8	DWORD0 Channel 4	DWORD0 Channel 0	
AE … AK							AWORD Channel 28	AWORD Channel 24	AWORD Channel 20	AWORD Channel 16		AWORD Channel 12	AWORD Channel 8	AWORD Channel 4	AWORD Channel 0	
AL … BD							DWORD1 Channel 28	DWORD1 Channel 24	DWORD1 Channel 20	DWORD1 Channel 16		DWORD1 Channel 12	DWORD1 Channel 8	DWORD1 Channel 4	DWORD1 Channel 0	
BE … BV							DWORD0 Channel 29	DWORD0 Channel 25	DWORD0 Channel 21	DWORD0 Channel 17		DWORD0 Channel 13	DWORD0 Channel 9	DWORD0 Channel 5	DWORD0 Channel 1	
BW … CD							AWORD Channel 29	AWORD Channel 25	AWORD Channel 21	AWORD Channel 17		AWORD Channel 13	AWORD Channel 9	AWORD Channel 5	AWORD Channel 1	
CE … CT							DWORD1 Channel 29	DWORD1 Channel 25	DWORD1 Channel 21	DWORD1 Channel 17		DWORD1 Channel 13	DWORD1 Channel 9	DWORD1 Channel 5	DWORD1 Channel 1	
CU, CV																
CW … DC							Reset, IEEE1500 Port, etc…				VDDC	Reset, IEEE1500 Port, etc…				
DD, DE																
DF … DU							DWORD1 Channel 30	DWORD1 Channel 26	DWORD1 Channel 22	DWORD1 Channel 18		DWORD1 Channel 14	DWORD1 Channel 10	DWORD1 Channel 6	DWORD1 Channel 2	
DW … EC							AWORD Channel 30	AWORD Channel 26	AWORD Channel 22	AWORD Channel 18		AWORD Channel 14	AWORD Channel 10	AWORD Channel 6	AWORD Channel 2	
ED … EU							DWORD0 Channel 30	DWORD0 Channel 26	DWORD0 Channel 22	DWORD0 Channel 18		DWORD0 Channel 14	DWORD0 Channel 10	DWORD0 Channel 6	DWORD0 Channel 2	
EV … FL							DWORD1 Channel 31	DWORD1 Channel 27	DWORD1 Channel 23	DWORD1 Channel 19		DWORD1 Channel 15	DWORD1 Channel 11	DWORD1 Channel 7	DWORD1 Channel 3	
FM … FU							AWORD Channel 31	AWORD Channel 27	AWORD Channel 23	AWORD Channel 19		AWORD Channel 15	AWORD Channel 11	AWORD Channel 7	AWORD Channel 3	
FV … GL							DWORD0 Channel 31	DWORD0 Channel 27	DWORD0 Channel 23	DWORD0 Channel 19		DWORD0 Channel 15	DWORD0 Channel 11	DWORD0 Channel 7	DWORD0 Channel 3	
GM … HA		Lower Left Edge Power Supply Region		Lower Center Edge Power Supply Region			Lower Right Edge Power Supply Region									

11.4 HBM4 Bump Map (cont'd)

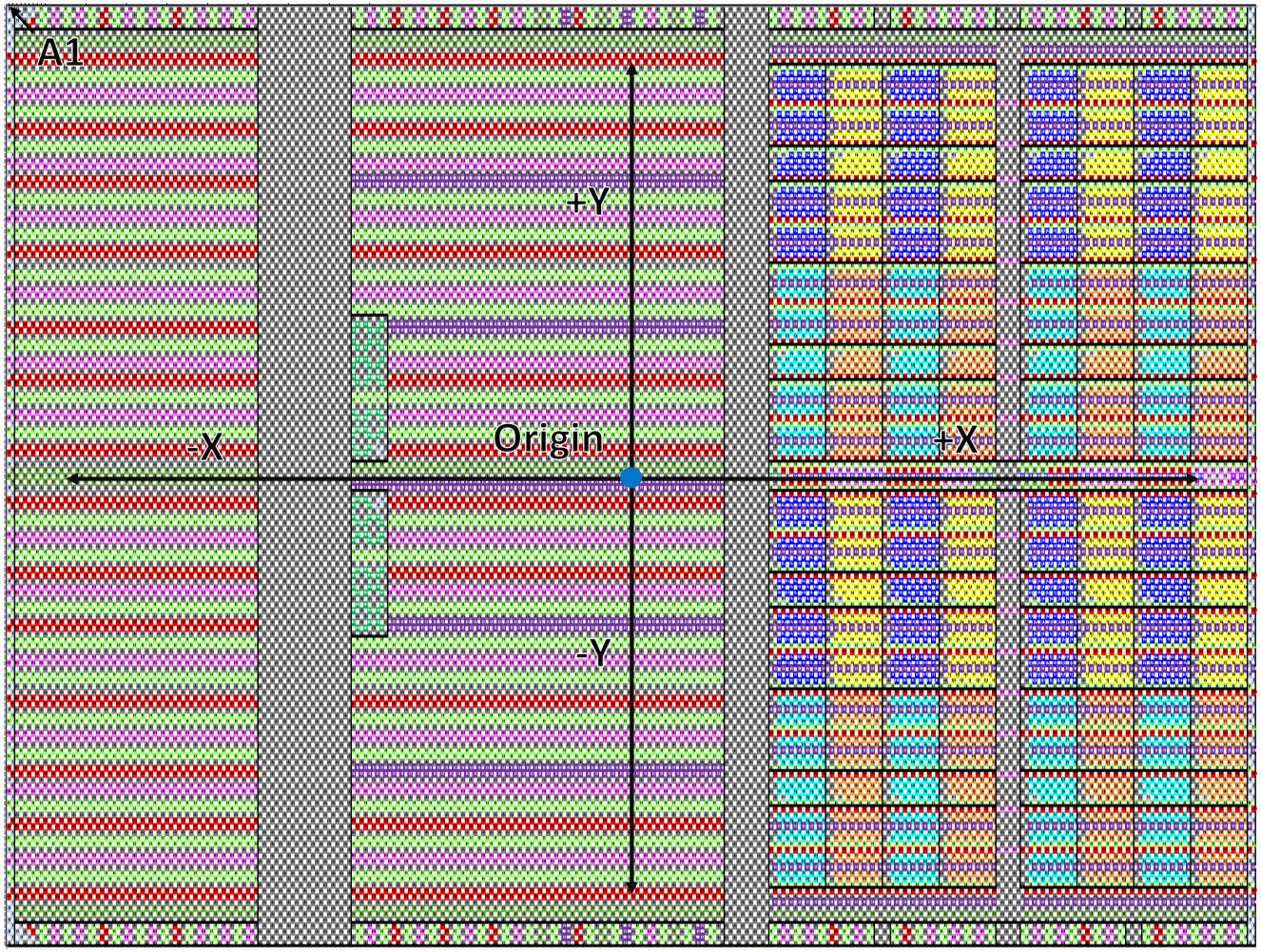


Figure 101 — Overview of HBM4 Bump Map Footprint

12 HBM DRAM Assembly

The HBM4 DRAM assembly is not defined by this standard. The shape and materials of the die to die interfaces between the die in the HBM4 DRAM are not defined in this standard and the shape (annular, cone, cylinder, etc.) and materials (Cu, W) are not defined or restricted in this standard. However, these interfaces must fit within the electrical requirements of the channel interface.

13 Test and Boundary Scan

HBM4 DRAMs provide two separate test interfaces as described below:

- a direct access(DA) test port intended for the vendor to access the HBM4 device independent of the host;
- an IEEE 1500 Standard test port, to be controlled by the host.

13.1 Direct Access (DA) Test Port

A direct access (DA) test port is available via DA[39:0] for vendor specific test implementations. Two microbumps are associated with each DA pin. A depopulated area for probing is located close to the DA port region in columns 86 to 94 of the HBM4 bump matrix (see 11.4 HBM4 Bump Map).

Access to the DA test port is controlled via pin DA12. When DA12 = LOW, DA[39:13][11:0] drivers are in Hi-Z and input receivers are disabled allowing the bus to float. When DA12 = HIGH, DA[39:13][11:0] are enabled for vendor specific test features and the IEEE 1500 port is disabled. CATTRIP output remain active but their state may not be valid and shall be ignored. The DA12 input is equipped with an internal pull-down resistor which ensures that DA12 is held LOW and the test port remains inactive even if the pin is left floating.

The DA test port may be enabled at any time after the power ramp has been completed and all supply voltages are within their defined ranges (t_{INIT0}), and after waiting for at least t_{INIT1} time. The level of the RESET_n pin shall be irrelevant for DA test port enabling.

The DA test port may be disabled at any time by pulling DA12 to LOW. The HBM4 DRAM may then resume normal operation after performing a device initialization as described in 4.3 Initialization Sequence with Stable Power the Initialization Sequence with Stable Power sections.

8 DA pins are designated to connect point-to-point to each HBM4 DRAM. 32 pins are designated to connect in parallel to up to four HBM4 DRAM devices on a multi drop bus as shown in Figure 102. The function of each of these pins is vendor specific. Table 109 defines which DA pins are allocated for point-to-point and for multi drop.

Table 109 — Direct Access (DA) Pin Allocation

Pin Group	DA Pin List	Pin Count
Point to Point	DA[19:12]	8
Multi Drop	DA[39:20][11:0]	32

13.1 Direct Access (DA) Test Port (cont'd)

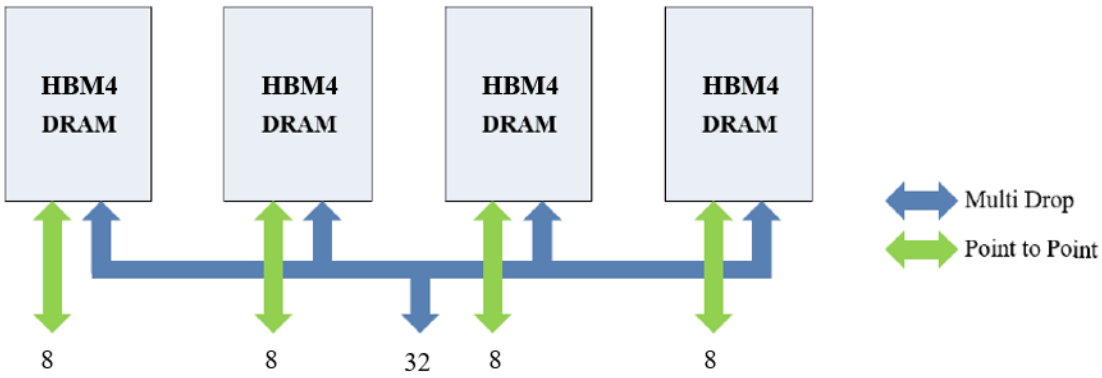


Figure 102 — DA Port Connection Diagram For Multiple HBM4 DRAM Devices

13.1.1 DA Test Port Lockout

The DA test port can be disabled (locked) by setting MR8 OP0 bit to 1. The bit is defined for channels 0 or 4 only. Once the bit is set to 1, the DA test port will remain disabled unless power is removed from the HBM4 DRAM. Any chip reset through pulling RESET_n LOW or via IEEE1500 HBM_RESET instruction, or writing a 0 via an MRS command or IEEE1500 instruction MODE_REGISTER_DUMP_SET will not clear the locked state.

13.2 IEEE Standard 1500

The IEEE Standard 1500 compliant test access port provides a direct test connection between a host and the HBM4 DRAM. The HBM4 DRAM's test port extends the standard specification and replicates the WSO output per channel. This allows some instructions to be executed in parallel across channels, and eliminates the need for cross-channel arbitration for WSO.

IEEE 1500 operations may be asserted at any time after device initialization and during normal memory operation including when the HBM4 DRAM is in power-down or self refresh mode. See an 13.6 Interaction with Mission Mode Operation section for how the various instructions interact with normal operation, and requirements for returning to normal operation. See also section of 4.4 Initialization Sequence For Use Of IEEE 1500 Instruction Including Lane Repairs and Channel Disable for a subset of operations that are allowed before the device initialization has been completed.

Please refer to ieee.org for more details about the IEEE1500 standard.

13.2.1 Interaction Between DA Test Port and IEEE1500 Test Access Port

DA12 = LOW selects the IEEE1500 test access port and DA12 = HIGH selects the DA test port. It is possible to operate the HBM4 DRAM without using the test ports. In this case the internal pull-down resistor on DA12 or pulling DA12 LOW in the system will keep the DA test port disabled, and pulling WRST_n LOW in the system will keep the IEEE1500 test port disabled.

13.2.1 Interaction Between DA Test Port and IEEE1500 Test Access Port (cont'd)

Table 110 summarizes the status of the test access port signals.

Table 110 — Test Access Port Signal Status

WRST_n	DA12, MR8 OP0	Signal Name	Type	Status
LOW	DA12 = LOW or MR8 OP0 = 1	Other IEEE1500 inputs[1]	Input	X (Don't Care)
		WSO	Output	V (Valid)[2]
		DA[39:13][11:0]	I/O	X (Don't Care)
HIGH	DA12 = LOW or MR8 OP0 = 1	Other IEEE1500 inputs[1]	Input	Active
		WSO	Output	V (Valid)[2]
		DA[39:13][11:0]	I/O	X (Don't Care)
Don't Care	DA12 = HIGH and MR8 OP0 = 0	Other IEEE1500 inputs[1]	Input	X (Don't Care)
		WSO	Output	V (Valid)[2]
		DA[39:13][11:0]	I/O	Vendor specific[3]

NOTE 1 WRCK, SelectWIR, ShiftWR, CaptureWR, UpdateWR, WSI.
NOTE 2 V = Valid Signal (either HIGH or LOW, but not floating).
NOTE 3 Please refer to vendor's datasheet.

13.2.2 IEEE1500 Test Access Port I/O Signals

Table 111 — IEEE1500 Test Port Signal List and Description

Symbol	Type	Description
WRCK	Input	Dedicated clock used to operate IEEE Std 1500 functions.
WRST_n	Input	When pulled LOW, WRST_n asynchronously puts the IEEE1500 test port into its normal system mode. No WRCK clocks are required when WRST_n is LOW. See WDR Reset State
WSI	Input	IEEE1500 test port serial input
SelectWIR	Input	SelectWIR determines whether the instruction register (WIR) or a wrapper data register is being accessed.
CaptureWR	Input	Controls a Capture operation in the selected wrapper register (WR)
ShiftWR	Input	Controls a Shift operation in the selected wrapper register (WR)
UpdateWR	Input	Controls an Update operation in the selected wrapper register (WR)
WSO[31:0]	Output	IEEE1500 test port per-channel serial output

13.2.3 IEEE1500 Test Access Port Functional Description

Figure 103 shows the HBM4 DRAM's IEEE1500 compliant architecture that uses an asymmetrical WSP (Wrapper Serial Port) with a single WSI and sixteen per channel WSOs. The standard compliant register stack is shown in the figure, including the Wrapper Bypass Register (WBY), Wrapper Boundary Register (WBR), and Wrapper Data Registers (WDR). The C, S and U notation for the registers refer to Capture, Shift and Update respectively, and indicate for each of the registers which functions are supported by that register. For example, the WBY only provides a Shift stage, whereas the WDRs provide Shift/Capture and Update stages.

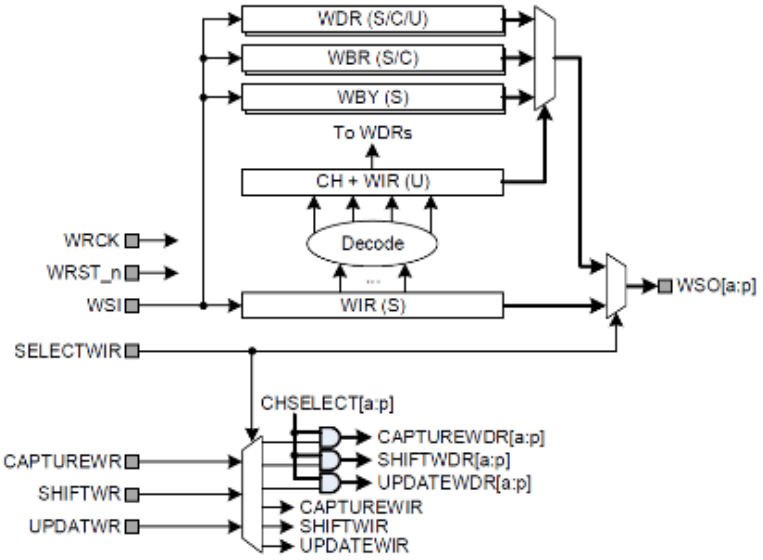


Figure 103 — IEEE Std. 1500 Logic Diagram

The WSO[31:0] output drivers are permanently enabled, with their drive state being LOW, HIGH, or undefined based on the current instruction loaded into the WIR. For example, if BYPASS is the current instruction, then WSO output data is defined only after one or more WRCK clock cycles have been applied. A WSO output will drive a LOW when a channel is disabled via the CHANNEL_DISABLE instruction or marked as “not present / not working” in the DEVICE_ID WDR.

The Wrapper Instruction Register (WIR) logic is included in Figure 103, and Figure 104 shows further details of the WIR implementation. The WIR and instruction opcodes are described in ~~clause~~ IEEE1500 Test Access Port Instruction Register section, and the instructions supported by IEEE1500 Test Instructions~~the HBM4 DRAM in clause Test Instructions~~. The five channel select bits of the WIR shift stage in Figure 104 are decoded to generate the CHSelect[31:0] outputs which control the per channel operation of the instructions. When a channel is not selected for an active instruction, then the CaptureWDR[31:0], ShiftWDR[31:0] and UpdateWDR[31:0] enables of the WSP are gated off. This will disable the WDRs of unselected channels for the decoded instruction. This gating is shown by the logic AND gates at the output of the de-multiplexer in Figure 103.

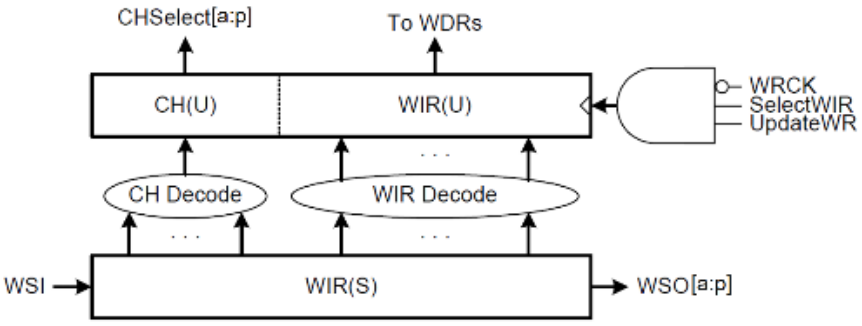


Figure 104 — WIR Channel Select Logic Diagram

HBM4 DRAMs are allowed to support less than 32 channels. The availability of each channel is coded in the DEVICE_ID WDR bits [39:8]. Unavailable channels do not respond to IEEE1500 instructions.

Figure 105 illustrates an IEEE1500 port operation sequence with a minimum number of WRCK cycles:

- Signal SelectWIR is set at clock edge T0. Control signals CaptureWR, ShiftWR and UpdateWR are all inactive as they are not allowed to change coincident with SelectWIR. SelectWIR must be kept stable until after completion of the complete sequence which spans until clock edge T4.
- A WDR capture operation is performed at clock edge T1 with CaptureWR sampled High at T1.
- A single WDR shift operation is performed at clock edge T2 with ShiftWR sampled High at T2.
- A WDR update operation is performed at clock edge T3b with UpdateWR sampled High at T3b. Please note that the update operation occurs on the falling WRCK clock edge.
- For some IEEE1500 port instructions a capture, shift or update event may not be specified; please refer to the description of each instruction for details.

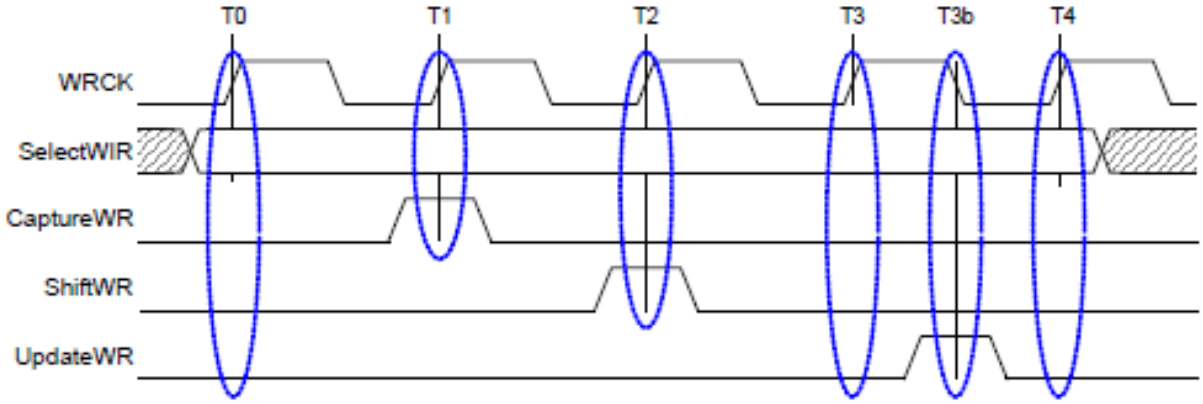


Figure 105 — IEEE1500 Port Operation

13.3 Wrapper Data Register (WDR) Types

13.3.1 Read Only (R) Wrapper Data Registers

WDR bit fields that are specified as read-only capture data into the shift stage register when a CaptureWR event is performed. The read-only WDRs keep their state during an UpdateWR event and do not have an update stage register. Read-only WDRs shift out their content during a ShiftWR event. Data shifted into WSI during the ShiftWR event is ignored.

13.3.2 Write Only (W) Wrapper Data Registers

WDR bit fields that are specified as write-only copy all data bits into the update stage register when an UpdateWR event is performed. When a write-only WDR is connected between WSI and WSO, any CaptureWR event has no effect on the WDR. Write-only WDRs shift out their content during the ShiftWR event.

13.3.3 Read and Write (R/W) Wrapper Data Registers

R/W WDRs operate as merged function of write-only and read-only WDRs. They capture data bits into the shift stage register during a CaptureWR event and copy bits from the shift stage into the update stage simultaneously when the UpdateWR event is performed.

13.3.4 WDR Reset State

Asserting WRST_n to LOW asynchronously asserts these states on the HBM4 DRAM's IEEE1500 test port logic:

- all WDRs place their update and / or shift stages (where applicable) into a state that ensures that the HBM4 DRAM returns to mission mode operation and all test modes are disabled;
- the WIR is set to BYPASS, effectively clearing any prior EXTEST_RX, EXTEST_TX, or CHANNEL_ID instruction, thus returning all functional pins to their normal functional mode. Boundary scan chain content is undefined;
- No change to any previously loaded SOFT_REPAIR, HARD_REPAIR, SOFT_LANE_REPAIR, or HARD_LANE_REPAIR register content;
- the content of the DWORD_MISR and AWORD_MISR registers is undefined;
- the AWORD MISR is disabled by setting bit 2 in the AWORD_MISR_CONFIG WDR to 0;
- the CHANNEL_DISABLE WDR is reset (refer to the CHANNEL_DISABLE instruction for conditions to re-enable a disabled channel);
- any ongoing MBIST operation will be terminated.

13.4 IEEE1500 Test Access Port Instruction Encodings

The HBM4 DRAM supports a 14-bit Wrapper Instruction Register (WIR). Bits WIR[13:8] select the channel and bits WIR[7:0] encode the test instruction. When SelectWIR is asserted, the WIR will not respond to the CaptureWR event and nothing will be captured into the WIR.

The WIR channel selection definition applies only to instructions defined in Table 114 (WIR[7:0] = 00h to 1Fh). The definition does not apply to vendor specific instructions, and vendors may use bits WIR[13:8] for different purposes.

Table 112 — WIR Channel Selection Definition

WIR [13:8]	Channel Select	WIR [13:8]	Channel Select	WIR [13:8]	Channel Select	WIR [13:8]	Channel Select
00h	Channel 0	08h	Channel 8	10h	Channel 16	18h	Channel 24
01h	Channel 1	09h	Channel 9	11h	Channel 17	19h	Channel 25
02h	Channel 2	0Ah	Channel 10	12h	Channel 18	1Ah	Channel 26
03h	Channel 3	0Bh	Channel 11	13h	Channel 19	1Bh	Channel 27
04h	Channel 4	0Ch	Channel 12	14h	Channel 20	1Ch	Channel 28
05h	Channel 5	0Dh	Channel 13	15h	Channel 21	1Dh	Channel 29
06h	Channel 6	0Eh	Channel 14	16h	Channel 22	1Eh	Channel 30
07h	Channel 7	0Fh	Channel 15	17h	Channel 23	1Fh	Channel 31

Table 113 — WIR Channel Selection Definition

WIR[13:8]	Channel Select
38h	16 Channels – 1st group
39h	16 Channels – 2nd group
3Ah	8 Channels - 1st group
3Bh	8 Channels - 2nd group
3Ch	8 Channels - 3rd group
3Dh	8 Channels - 4th group
3Eh	Not used. (all channels selected)
3Fh	All channels -32 channels
Xh	Ignored (all channels selected)
NOTE1 See the vendor datasheets for the mapping of channels for 3Ah, 3Bh, 3Ch and 3Dh	

13.5 Test Instructions

Test instructions supported by the HBM4 DRAM are listed in Table 114 and subsequently described in detail.

Table 114 — Instruction Register Encodings

WIR [13:8]	WIR [7:0]	Instruction	Description	Register Type	WDR Length
Xh	00h	BYPASS	Bypass	R/W	1
3Fh, 1Fh-00h	01h	EXTEST_RX	Microbump boundary scan Rx test (open/ short)	R	122
3Fh, 1Fh-00h	02h	EXTEST_TX	Microbump boundary scan Tx test (open/ short)	W	122
	03h	RFU			
	04h	RFU			
Xh	05h	HBM_RESET	Functional reset excluding Wrapper Data Registers (WDRs) and any IEEE1500 test port logic or I/Os	W	1
3Fh, 1Fh-00h	06h	MBIST	HBM4 DRAM resident Memory BIST engine test		Vendor specific
1Fh-00h	07h	SOFT_REPAIR	Soft repair of failing memory array bit cell		26
1Fh-00h	08h	HARD_REPAIR	Hard repair of DRAM failing memory array bit cell		26
3Fh, 1Fh-00h	09h	DWORD_MISR	Read back for DWORD MISR and write of a seed value	R/W	320
3Fh, 1Fh-00h	0Ah	AWORD_MISR	Read back for AWORD MISR	R	38
3Fh, 1Fh-00h	0Bh	CHANNEL_ID	All TX I/Os go HIGH (except I/Os in MIDSTACK region)	W	1
	0Ch	RFU			
3Fh, 1Fh-00h	0Dh	AWORD_MISR_ CONFIG	Allows IEEE1500 test port access to configure the AWORD MISR test feature	W	8
3Fh, 1Fh-00h	0Eh	DEVICE_ID	Returns the HBM4 DRAM's unique identification code	R	175
Xh	0Fh	TEMPERATURE	Returns a 9-bit binary temperature code	R	9
3Fh, 1Fh-00h	10h	MODE_REGISTER_ DUMP_SET	Returns and set the HBM4 DRAM's Mode Register values	R/W	160[9]
3Fh, 1Fh-00h	11h	READ_LFSR_ COMPARE_STICKY	Reads the sticky bit error for LFSR Compare feature	R	99
1Fh-00h	12h	SOFT_LANE_REPAIR	Soft Lane Remapping	R/W	45
1Fh-00h	13h	HARD_LANE_REPAIR	Hard Lane Remapping	R/W	45

Table 109 — Instruction Register Encodings (cont'd)

WIR [13:8]	WIR [7:0]	Instruction	Description	Register Type	WDR Length
3Fh, 1Fh-00h	14h	CHANNEL_DISABLE	Disables a channel (All-channel disable is optional)	W	1
3Fh, 1Fh-00h	15h	CHANNEL TEMPERATURE	Returns a 9-bit binary channel temperature code per SID	R	36
Xh	16h	WOSC_RUN	WDQS Interval Oscillator	W	1
Xh	17h	WOSC_COUNT	WDQS Interval Oscillator Count	R	25
1Fh-00h	18h	ECS_ERROR_LOG	Error Check and Scrub (ECS) Error Log Information	R	216
1Fh-00h	19h	HS_REP_CAP	Returns whether banks have repair resources or not	R	256
38h, 39H,[4] 3Ah-3Dh[5]	1Ah	SELF_REP	Self repair	R/W	9
38h, 39H,[6] 3Ah-3Dh[7]	1Bh	SELF_REP_RESULTS	Self repair results	R	8
	1Ch-3Fh	RFU			
Vendor specific	40h-FFh	Vendor specific			

NOTE 1 Unsupported instruction codes will default to the BYPASS instruction when the WIR is updated with the unsupported encoding.

NOTE 2 Channels that are not selected by WIR[13:8] do not respond to the instruction and ignore any Update, Capture and Shift events.

NOTE 3 WDRs shift out the least significant bit on the WSO port at the first WRCK of the shift sequence. WSO output timing and valid data window are defined in Table 140 – IEEE1500 Test Port AC Timings

NOTE 4 WIR[13:8] value of 38h or 39h is for enabling self repair on groups of 16 channels. The channels associated with 38h and 39h are vendor specific.

NOTE 5 WIR[13:8] value of 3Ah, 3Bh, 3Ch, and 3Dh is for enabling self repair on groups of 8 channels. The channels associated with 3Ah, 3Bh, 3Ch, and 3Dh are vendor specific.

NOTE 6 WIR[13:8] value of 38h or 39h is for enabling self repair results on groups of 16 channels. The channels associated with 38h and 39h are vendor specific.

NOTE 7 WIR[13:8] value of 3Ah, 3Bh, 3Ch, and 3Dh is for enabling self repair results on groups of 8 channels. The channels associated with 3Ah, 3Bh, 3Ch, and 3Dh are vendor specific.

NOTE 8 Global test instructions (WIR[13:8] = Xh) drive the same data on the WSO outputs of all active channels during ShiftWR events. Inactive channels (channels that are marked as "not present / not working" in the DEVICE_ID WDR and channels that have been disabled using the CHANNEL_DISABLE instruction) drive a static LOW on their WSO outputs.

NOTE 9 The maximum WDR length for MODE_REGISTER_DUMP_SET is 160 bits but is vendor specific. Refer to supplier datasheet for more information.

13.5.1 BYPASS

The BYPASS instruction places a single bit WDR between WSI and each channel's WSO and RM. Data is shifted from WSI to WSO, RM0, and RM1 through the one bit WDR by WRCK.

BYPASS is the default instruction after asserting WRST_n to LOW.

Wrapper Data Register

When BYPASS is the current instruction, the 1-bit shift register as shown in Table 115 is connected between WSI, RM[1:0], and WSO[31:0], and the WSO and RM outputs of all active channels drive the same data during ShiftWR events.

CaptureWR

When BYPASS is the current instruction, the CaptureWR event will have no effect.

UpdateWR

When BYPASS is the current instruction, the UpdateWR event will have no effect.

Table 115 — BYPASS Wrapper Data Register

Bit Position	Bit Field	Type	Description
0	BYPASS	-	Single bit bypass shift register per IEEE1500 Standard

13.5.2 EXTEST_RX and EXTEST_TX

EXTEST_RX and EXTEST_TX are both intended for DC I/O connectivity testing similar to board level boundary scan. The receive notation in EXTEST_RX designates that the HBM4 I/O will sample the logic value and capture into the data register the value that is present at the micro bump interface. The transmit notation in EXTEST_TX designates that the HBM4 I/O will drive the logic value shifted into the data register at the micro bump interface. All HBM4 bidirectional I/O, inputs and outputs support both instructions. Differential inputs and outputs (CK_t/CK_c, WDQS_t/WDQS_c and RDQS_t/RDQS_c) also support both instructions on both the true and complement pins.

While EXTEST_RX is the current instruction, all functional pins of the selected channel(s) enter a High-Z state, including the output-only pins AERR, DERR, RDQS_t/RDQS_c, and CATTRIP. See also the Boundary Scan section.

I/O signals power up in input mode by default. The host will put all AWORD and DWORD drivers into High-Z state prior to loading the EXTEST_TX instruction into the WIR. As soon as EXTEST_TX becomes the current instruction, all AWORD and DWORD signals will change to output mode and remain in output mode until reset of the test logic or until a different instruction is updated on the channel.

A channel disabled either via the corresponding CHANNEL_AVAILABLE bit in the DEVICE ID WDR or via the CHANNEL_DISABLE instruction will not respond to the EXTEST_TX or EXTEST_RX instructions.

Wrapper Data Register

When EXTEST_RX or EXTEST_TX is the current instruction, the Wrapper Boundary Register (WBR) as shown in Table 116 is connected between WSI and WSO.

CaptureWR

When EXTEST_RX is the current instruction, the CaptureWR event will capture the input values into the shift stage of the WDR. The captured data is shifted out on WSO during a subsequent ShiftWR event. Inputs must be stable for the setup and hold times t_{SEXT} and t_{HEXT}.

When EXTEST_TX is the current instruction, the CaptureWR event will have no effect.

ShiftWR

The Wrapper Boundary Register (WBR) does not provide an update stage. When EXTEST_TX is the current instruction, the value driven on the outputs is directly derived from the WDR's shift stage and will update with each ShiftWR event. The new data will be stable after t_{OVEXT} time.

UpdateWR

When EXTEST_RX or EXTEST_TX is the current instruction, the UpdateWR event will have no effect.

13.5.2 EXTEST_RX and EXTEST_TX (cont'd)

Table 116 — Wrapper Boundary Register (WBR)

Bit Position	Bit Field	Type	Description
121	RFU	I	
120	RFU	I	
119	DBI7	I/O	DWORD1 (PC1)
118	DQ63	I/O	
117	DQ62	I/O	
116	DQ61	I/O	
115	DQ60	I/O	
114	RD3	I/O	
113	DERR1	I/O	
112	DQ59	I/O	
111	DQ58	I/O	
110	DQ57	I/O	
109	DQ56	I/O	
108	SEV3	I/O	
107	DBI6	I/O	
106	DQ55	I/O	
105	DQ54	I/O	
104	DQ53	I/O	
103	DQ52	I/O	
102	RDQS1_c	I/O	
101	RDQS1_t	I/O	
100	DQ51	I/O	
99	DQ50	I/O	
98	DQ49	I/O	
97	DQ48	I/O	
96	SEV2	I/O	
95	DBI5	I/O	
94	DQ47	I/O	
93	DQ46	I/O	
92	DQ45	I/O	
91	DQ44	I/O	
90	WDQS1_c	I/O	
89	WDQS1_t	I/O	
88	DQ43	I/O	
87	DQ42	I/O	
86	DQ41	I/O	

Table 115 — Wrapper Boundary Register (WBR) (cont'd)

Bit Position	Bit Field	Type	Description
85	DQ40	I/O	DWORD1 (PC1) (cont'd)

84	ECC3	I/O	
83	DBI4	I/O	
82	DQ39	I/O	
81	DQ38	I/O	
80	DQ37	I/O	
79	DQ36	I/O	
78	RD2	I/O	
77	DPAR1	I/O	
76	DQ35	I/O	
75	DQ34	I/O	
74	DQ33	I/O	
73	DQ32	I/O	
72	ECC2	I/O	
71	AERR	I/O	AWORD
70	R9	I/O	
69	R8	I/O	
68	R7	I/O	
67	R6	I/O	
66	CK_c	I/O	
65	R5	I/O	
64	R4	I/O	
63	R0	I/O	
62	R3	I/O	
61	R2	I/O	
60	R1	I/O	
59	RA	I/O	
58	ARFU	I/O	
57	APAR	I/O	
56	C7	I/O	
55	C6	I/O	
54	C5	I/O	
53	CK_t	I/O	
52	C4	I/O	
51	C3	I/O	
50	C2	I/O	
49	C1	I/O	
48	C0	I/O	
47	DBI3	I/O	DWORD0 (PC0)
46	DQ31	I/O	
45	DQ30	I/O	
44	DQ29	I/O	
43	DQ28	I/O	
42	RD1	I/O	

Table 111 — Wrapper Boundary Register (WBR) (cont'd)

Bit Position	Bit Field	Type	Description
41	DERR0	I/O	DWORD0 (PC0) (cont'd)
40	DQ27	I/O	
39	DQ26	I/O	
38	DQ25	I/O	
37	DQ24	I/O	
36	SEV1	I/O	
35	DBI2	I/O	
34	DQ23	I/O	
33	DQ22	I/O	
32	DQ21	I/O	
31	DQ20	I/O	
30	RDQS0_c	I/O	
29	RDQS0_t	I/O	
28	DQ19	I/O	
27	DQ18	I/O	
26	DQ17	I/O	
25	DQ16	I/O	
24	SEV0	I/O	
23	DBI1	I/O	
22	DQ15	I/O	
21	DQ14	I/O	
20	DQ13	I/O	
19	DQ12	I/O	
18	WDQS0_c	I/O	
17	WDQS0_t	I/O	
16	DQ11	I/O	
15	DQ10	I/O	
14	DQ9	I/O	
13	DQ8	I/O	
12	ECC1	I/O	
11	DBI0	I/O	
10	DQ7	I/O	
9	DQ6	I/O	
8	DQ5	I/O	
7	DQ4	I/O	
6	RD0	I/O	
5	DPAR0	I/O	
4	DQ3	I/O	
3	DQ2	I/O	
2	DQ1	I/O	
1	DQ0	I/O	
0	ECC0	I/O	

13.5.3 HBM_RESET

The HBM_RESET instruction initiates an asynchronous functional reset of the HBM4 DRAM, equivalent to asserting RESET_n to LOW.

The HBM_RESET condition is not self-clearing. Instead, the reset state must explicitly be set and cleared. To accomplish an HBM4 reset, the HBM_RESET bit must be held as 1 for a minimum duration of t_{RES} which equals t_{PW_RESET} (see Initialization Sequence with Stable Power).

It is pointed out that the Wrapper Serial Port (WSP) itself including the associated control logic and WDRs is not reset by the HBM_RESET instruction. The DA port signal pins are also not affected by the HBM_RESET instruction.

Wrapper Data Register

When HBM_RESET is the current instruction, the data register as shown in Table 117 is connected between WSI and WSO[31:0], and the WSO outputs of all active channels drive the same data during ShiftWR events.

CaptureWR

When HBM_RESET is the current instruction, the CaptureWR event will have no effect.

UpdateWR

When HBM_RESET is the current instruction, the UpdateWR event will load the value from the shift stage into the update stage and initiate or clear the functional reset.

Table 117 — HBM_RESET Wrapper Data Register

Bit Position	Bit Field	Type	Description
0	HBM_RESET	W	0 - Clear the functional reset 1 - Initiate the functional reset

Internally, the RESET_n pin and the HBM_RESET instruction are logically combined such that when either is true then the internal reset state is true. During power-up it is required that WRST_n be driven LOW, thus ensuring that the uninitialized IEEE1500 test port logic does not interfere with the power-up initialization sequence.

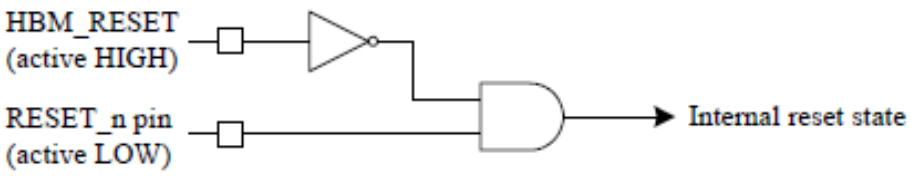


Figure 106 — RESET_n and HBM_RESET Logic

13.5.3 HBM4_RESET (cont'd)

After the power-up initialization, the RESET_n input is HIGH, and subsequent stable power resets may be asserted by either driving the RESET_n input LOW, or by using the HBM_RESET instruction. Note that the HBM_RESET instruction does not bring the HBM4 DRAM out of reset while the external RESET_n input is driven LOW. Similarly, the HBM4 DRAM cannot be brought out of reset using the RESET_n input while reset is asserted using the HBM_RESET instruction.

Table 118 — RESET_n and HBM_RESET Truth Table

RESET_n	HBM_RESET	Internal Reset State
LOW	0	Reset asserted by RESET_n
LOW	1	Reset asserted by both RESET_n pin and HBM_RESET instruction
HIGH	0	Exit reset state
HIGH	1	Reset asserted by HBM_RESET instruction

13.5.4 MBIST

The MBIST instruction is used for HBM4 DRAM hosted memory built in self-test. HBM devices must support memory MBIST. This instruction format and data register field configuration is required for IEEE Std 1500 access to the test feature. MBIST engine clock source can be WRCK as a direct clock source or reference clock source or an internal clocked mode independent of WRCK and independent of any I/O functional clocks is also acceptable.

Wrapper Data Register

When MBIST is the current instruction, the vendor specific data register as shown in Table 119 is connected between WSI and WSO.

CaptureWR

When MBIST is the current instruction, the CaptureWR event will capture R or R/W bit fields into the shift stage of the WDR.

UpdateWR

When MBIST is the current instruction, the UpdateWR event will load the W and R/W bit fields from the shift stage to the update stage of the WDR simultaneously.

Table 119 — MBIST Wrapper Data Register

Bit Position	Bit Field	Type	Description
Vendor specific	Vendor specific		Vendor specific

13.5.5 SOFT_REPAIR

The SOFT_REPAIR instruction allows the user to temporarily repair bit cells in the HBM4 DRAM without using permanent fusing mechanism to initiate the repair. This feature is intended to enable validation that the intended repair works as expected. Once a soft repair is validated, the user may choose to perform a fused hard repair via the HARD_REPAIR instruction. If DRAM power is removed or the DRAM is RESET, the SOFT_REPAIR will revert to the un-repaired state.

Repair resources (redundant rows) are provided per PC and per bank. The actual number of repair resources are vendor specific. The number and availability of repair resources are provided via the HS_REP_CAP instruction. The use of SOFT_REPAIR will not decrement the HS_REP_CAP register so the host controller must track the resources used. If there is no repair resource available in a certain bank then the host controller should not issue a SOFT_REPAIR to that bank. However, if a SOFT_REPAIR sequence is issued to a bank with no repair resource available, the DRAM will ignore the programming sequence.

The SOFT_REPAIR granularity indicating the number of repaired rows per SOFT_REPAIR is vendor specific. The address bits associated with the granularity are also vendor specific and indicated in the PPR_RA[13:0] field of the DEVICE_ID WDR. See Table 126 for more details.

The SOFT_REPAIR supports an Undo and Lock function. The SOFT_REPAIR Undo will restore a previously used repair resource back to its unused state and the same time reactivate the original (unrepaired) row instead. The complete address information comprising the PC, SID, bank and row address must be provided with the SOFT_REPAIR instruction as described in Table 120, and the SOFT_REPAIR_UNDO and SOFT_REPAIR_START fields must set to “1”.

The host controller can lock down a used soft repair resource by issuing the SOFT_REPAIR instruction with the SOFT_REPAIR_LOCK and SOFT_REPAIR_START bits as “1”. Each SOFT_REPAIR resource supports the Lock feature. For both UNDO/LOCK cases, the HBM4 DRAM may ignore the row address bit if it so chooses, as the SID, PC, BK are enough to uniquely identify the SOFT_REPAIR resource. The row address may be ignored if there is only single repair resource. If a host issues a SOFT_REPAIR on an already repaired but unlocked row then HBM4 DRAM will allocate another repair resource in response to a host request if an available resource exists. Support for the feature is vendor specific. A locked repair resource cannot be used to replace another row or being set back to the unused state using the Undo function. Only a chip reset (RESET_n pulled LOW) or power-cycling can unlock a locked repair resource.

When using soft repair specifically with the Undo function, the host controller must manage and schedule the refresh operation properly on the valid data of a row address. If a row has been repaired, all refresh commands will exclude the original row from being refreshed and refresh the repair row instead. Similarly, unused repair resources will not be refreshed which includes those resources that had been allocated but were then set back to the unused state using the Undo operation. Especially when switching back and forth between an original and a repair row, regular refresh commands may not hit both rows within the required refresh interval. A possible method to prevent a potential data loss is to explicitly issue ACTIVATE and PRECHARGE commands to the mapped-out rows before and after the SOFT_REPAIR operations.

The SOFT_REPAIR UNDO and LOCK are mutually exclusive. So, in the case of any SOFT_REPAIR instruction issued, the SOFT_REPAIR UNDO and LOCK must not be set “1” at the same time.

A channel must be in bank idle state as long as the SOFT_REPAIR instruction is loaded in the WIR.

13.5.5 SOFT_REPAIR (cont'd)

Wrapper Data Register

When the SOFT_REPAIR instruction is updated the data register as shown in Table 120 is connected between WSI and WSO.

CaptureWR

When SOFT_REPAIR is the current instruction, the CaptureWR event will have no effect.

UpdateWR

When SOFT_REPAIR is the current instruction, the UpdateWR event will load the write only bit field from the shift stage into the update stage simultaneously. Completion of the update event will initiate the soft repair sequence.

Table 120 — SOFT_REPAIR Wrapper Data Register

Bit Position	Bit Field	Type	Description
[25]	SOFT_REPAIR_LOCK	W	0b – SOFT_REPAIR is open 1b – SOFT_REPAIR is hard-locked
[24]	SOFT_REPAIR_UNDO	W	0b – Do SOFT_REPAIR (SOFT_REPAIR enabled) 1b – Undo SOFT_REPAIR (SOFT_REPAIR not enabled)
[23]	SOFT_PC	W	PC
[22:21]	SOFT_SID	W	SID[1:0]
[20:17]	SOFT_BK	W	BA[3:0]
[16:1]	SOFT_ROW	W	RFU, RA[13:0][1]
[0]	SOFT_REPAIR_START	W	0b – Disabled (Default) 1b – Enabled

NOTE 1 SOFT_ROW includes an additional bit to support future row addressing, i.e., RA14.

13.5.6 HARD_REPAIR

The HARD_REPAIR instruction is used to permanently repair failing bit cells detected in the HBM4 DRAM. A fuse rupture scheme is used to implement the repair. The repair sequence will be initiated on update of the data register. After some vendor specified time period fuse rupture automatically completes and repair is affected. Hard repair will be permanent. Completion of HARD_REPAIR requires a subsequent chip reset (RESET_n pulled LOW) as described in Interaction with Mission Mode Operation. The HBM vendor is required to specify the time to wait after updating the HARD_REPAIR WDR as well as any requirements for WRCK clocking if required to perform the repair.

All channels of the HBM4 DRAM must be in bank idle state as long as the HARD_REPAIR instruction is loaded in the WIR.

Wrapper Data Register

When HARD_REPAIR is the current instruction, the data register as shown in Table 121 is connected between WSI and WSO.

CaptureWR

When HARD_REPAIR is the current instruction, the CaptureWR event will have no effect.

UpdateWR

When HARD_REPAIR is the current instruction, the UpdateWR event will load the write-only bit fields from the shift stage into the update stage and initiate the hard repair sequence. The hard repair is completed after a waiting time of t_{HREP}.

Table 121 — HARD_REPAIR Wrapper Data Register

Bit Position	Bit Field	Type	Description
[25:24]	RESERVED		
[23]	HARD_PC	W	PC
[22:21]	HARD_SID	W	SID[1:0]
[20:17]	HARD_BK	W	BA[3:0]
[16:1]	HARD_ROW	W	RFU, RA[13:0][1]
[0]	HARD_REPAIR_START	W	0b: Disabled (default) 1b: Enabled

NOTE 1 HARD_ROW includes an additional bit to support future row addressing, i.e., RA14.

13.5.7 DWORD_MISR

This instruction captures and shifts out the DWORD MISR value on the WSO output. The instruction may also be used to preload data for use in LFSR mode. The DWORD MISR is associated with the DWORD IO test feature.

Note that the MISR content is not specified after shifting out the MISR content. The host should reinitialize the MISR before continuing with additional testing, e.g. by using the MISR Preset function in Mode Register 7 (MR7). See section 6.8 HBM4 Loopback Test Modes features and usage.

Wrapper Data Register

When DWORD_MISR is the current instruction, the data register as shown in Table 122 is connected between WSI and WSO. The notation is "…_Q0" to "…_Q3" for the 4 UI per CK clock cycle latched by WDQS in MISR mode or driven along with RDQS in LFSR mode.

CaptureWR

When DWORD_MISR is the current instruction, the CaptureWR event will load the respective MISR values into the shift stage of the WDR. A minimum waiting time of tSMISR between the last data capture into the DWORD MISR and this CaptureWR event must be observed.

UpdateWR

When DWORD_MISR is the current instruction, the UpdateWR event will load the bits from the shift stage of the WDR into the DWORD MISR.

Table 122 — DWORD_MISR Wrapper Data Register

Bit Position	Bit Field	Type	Description
[319:160]	DWORD1	R/W	DWORD1: DQ[63:32], DBI[7:4], ECC[3:2] and SEV[3:2] (Same bit ordering as DWORD0)
[159:120]	DWORD0_BYTE3	R/W	Byte 3 of DWORD0: DQ[31:24], DBI3 and SEV1 (Same ordering as Byte 0)
[119:80]	DWORD0_BYTE2	R/W	Byte 2 of DWORD0: DQ[23:16], DBI2 and SEV0 (Same ordering as Byte 0)
[79:40]	DWORD0_BYTE1	R/W	Byte 1 of DWORD0: DQ[15:8], DBI1 and ECC1 (Same ordering as Byte 0)
39	DWORD0_DBI0_Q0	R/W	Byte 0 of DWORD0
38	DWORD0_DBI0_Q1	R/W	
37	DWORD0_DBI0_Q2	R/W	
36	DWORD0_DBI0_Q3	R/W	
35	DWORD0_DQ7_Q0	R/W	
34	DWORD0_DQ7_Q1	R/W	
33	DWORD0_DQ7_Q2	R/W	

Bit Position	Bit Field	Type	Description
32	DWORD0_DQ7_Q3	R/W	Byte 0 of DWORD0 (cont'd)
31	DWORD0_DQ6_Q0	R/W	
30	DWORD0_DQ6_Q1	R/W	
29	DWORD0_DQ6_Q2	R/W	
28	DWORD0_DQ6_Q3	R/W	
27	DWORD0_DQ5_Q0	R/W	
26	DWORD0_DQ5_Q1	R/W	
25	DWORD0_DQ5_Q2	R/W	
24	DWORD0_DQ5_Q3	R/W	
23	DWORD0_DQ4_Q0	R/W	
22	DWORD0_DQ4_Q1	R/W	
21	DWORD0_DQ4_Q2	R/W	
20	DWORD0_DQ4_Q3	R/W	
19	DWORD0_DQ3_Q0	R/W	
18	DWORD0_DQ3_Q1	R/W	
17	DWORD0_DQ3_Q2	R/W	
16	DWORD0_DQ3_Q3	R/W	
15	DWORD0_DQ2_Q0	R/W	
14	DWORD0_DQ2_Q1	R/W	
13	DWORD0_DQ2_Q2	R/W	
12	DWORD0_DQ2_Q3	R/W	
11	DWORD0_DQ1_Q0	R/W	
10	DWORD0_DQ1_Q1	R/W	
9	DWORD0_DQ1_Q2	R/W	
8	DWORD0_DQ1_Q3	R/W	
7	DWORD0_DQ0_Q0	R/W	
6	DWORD0_DQ0_Q1	R/W	
5	DWORD0_DQ0_Q2	R/W	
4	DWORD0_DQ0_Q3	R/W	
3	DWORD0_ECC0_Q0	R/W	
2	DWORD0_ ECC0_Q1	R/W	
1	DWORD0_ ECC0_Q2	R/W	
0	DWORD0_ ECC0_Q3	R/W	

13.5.8 AWORD_MISR

This instruction captures and shifts out the AWORD MISR value on the WSO output. The MISR in this instruction is associated with the AWORD loopback test feature. The data register bit positions are specified in Table 123.

Note that the content of the MISR is not specified after shifting out the MISR content. The host should reinitialize the MISR using the AWORD_MISR_CONFIG instruction before continuing with additional testing. See 6.8 HBM4 Loopback Test Modes for MISR mode features and usage.

Wrapper Data Register

When AWORD_MISR is the current instruction, the data register as shown in Table 123 is connected between WSI and WSO. The notation is "…_R" for bits latched on the rising CK clock edge and "…_F" for bits latched on the falling CK clock edge.

CaptureWR

When AWORD_MISR is the current instruction, the CaptureWR event will load the respective MISR values into the shift stage of the WDR. A minimum waiting time of tSMISR between the last data capture into the AWORD MISR and this CaptureWR event must be observed.

UpdateWR

When AWORD_MISR is the current instruction, the UpdateWR event will have no effect.

13.5.8 AWORD_MISR (cont'd)

Table 123 – AWORD_MISR Wrapper Data Register

Bit Position	Bit Field	Type	Description
37	R1_R	R	AWORD
36	R1_F	R	
35	R2_R	R	
34	R2_F	R	
33	R3_R	R	
32	R3_F	R	
31	R0_R	R	
30	R0_F	R	
29	R4_R	R	
28	R4_F	R	
27	R5_R	R	
26	R5_F	R	
25	R6_R	R	
24	R6_F	R	AWORD (cont'd)
23	R7_R	R	
22	R7_F	R	
21	R8_R	R	
20	R8_F	R	
19	R9_R	R	
18	R9_F	R	
17	ARFU_R	R	
16	ARFU_F	R	
15	C7_R	R	
14	C7_F	R	
13	C6_R	R	
12	C6_F	R	
11	C5_R	R	
10	C5_F	R	
9	C4_R	R	
8	C4_F	R	
7	C3_R	R	
6	C3_F	R	
5	C2_R	R	
4	C2_F	R	
3	C1_R	R	
2	C1_F	R	
1	C0_R	R	
0	C0_F	R	

13.5.9 CHANNEL_ID

This instruction enables the HBM4 channel identification by driving all bidirectional DWORD I/Os to HIGH, unless a channel is disabled either via the corresponding CHANNEL_AVAILABLE bit in the DEVICE ID WDR or via the CHANNEL_DISABLE instruction. In these cases a channel will not respond to the CHANNEL_ID instruction.

Wrapper Data Register

When CHANNEL_ID is the current instruction, the data register as shown in Table 124 is connected between WSI and WSO.

CaptureWR

When CHANNEL_ID is the current instruction, the CaptureWR event will have no effect.

UpdateWR

When CHANNEL_ID is the current instruction, the UpdateWR event will load the enable bit from the shift stage into the update stage of the WDR. All DWORD bidirectional I/Os (DQ, DBI, RD, ECC/SEV and DPAR) will drive a HIGH latest after t_{OVCHN} when the enable bit is 1, and return to their default state latest after t_{OZCHN} when the enable bit is 0 or a different instruction has been loaded in the WIR. Output pins (RDQS_t/c, DERR) maintain the default state regardless of the CHANNEL_ID instruction. DBI, ECC and DPAR will drive a HIGH even if the respective function is disabled in the Mode Register.

Table 124 — CHANNEL_ID Wrapper Data Register

Bit Position	Bit Field	Type	Description
0	ENABLE	W	0 - TX return to their default state 1 - TX drive a HIGH

13.5.10 AWORD_MISR_CONFIG

This instruction configures the AWORD MISR for subsequent tests. See 6.8 HBM4 Loopback Test Modes for MISR mode features and usage.

Wrapper Data Register

When AWORD_MISR_CONFIG is the current instruction, the data register as shown in Table 125 is connected between WSI and WSO.

CaptureWR

When AWORD_MISR_CONFIG is the current instruction, the CaptureWR event will have no effect.

13.5.10 AWORD_MISR_CONFIG (cont'd)

UpdateWR

When AWORD_MISR_CONFIG is the current instruction, the UpdateWR event will load the configuration bits from the shift stage into the update stage of the WDR and configures the AWORD MISR into the desired mode. The new configuration is valid for subsequent AWORD MISR operation once the t_{CMISR} timing has elapsed.

Table 125 — AWORD_MISR_CONFIG Wrapper Data Register

Bit Position	Bit Field	Type	Description
[7:3]	VENDOR_SPECIFIC	W	00000 - No action (default) All others - Reserved for vendor specific the AWORD MISR Configuration
2	ENABLE	W	0 - Off 1 - On
[1:0]	MODE[1:0]	W	00 - Preset a) The 38-bit AWORD MISR is preset to 0x2AAAAAAAAAh; b) the AWORD LFSR_COMPARE_STICKY bits are all cleared to 0; c) the AWORD preamble clock filter circuit is enabled. 01 - LFSR Compare mode 10 - Register mode: AWORD transfers are captured directly to the MISR register. Note that Register mode cannot be used to set an alternate seed value (see Test method for AWORD (Write) Register Mode) 11 - MISR mode

13.5.11 DEVICE_ID

This instruction allows shift out of the DEVICE_ID that provides various information about the HBM4 DRAM including a device specific unique serial number. The per channel ID data registers are intended to support vendors who require additional resolution for identifying the device.

Wrapper Data Register

When the DEVICE_ID instruction is updated the data register as shown in Table 126 is connected between WSI and WSO.

CaptureWR

When DEVICE_ID is the current instruction, the CaptureWR event will load the respective identification field values into the shift stage of the WDR.

UpdateWR

When DEVICE_ID is the current instruction, the UpdateWR event will have no effect.

13.5.11 DEVICE_ID (cont'd)

Table 126 — DEVICE_ID Wrapper Data Register

Bit Position	Bit Field	Type	Description
[175:171]	OPT_FEATURES[4:0]	R	Reserved to indicate the support of optional features that may be added in a future revision of this standard. 00000 - default
[170]	SHARED_REP_RES	R	Sharing of HS_REP_CAP with self repair 0: Self repair resources are separate from hard/soft resources 1: Self repair resources are shared with hard/soft resources
[169]	PPR_RSVD [2]	R	Reserved row addresses associated with a single soft or hard repair. 0 - default
[168:155]	PPR_RA[13:0]	R	Row addresses associated with a single soft or hard repair. Encoding: 0 - row address is evaluated 1 - row address is ignored bit 0: RA0 bit 1: RA1 ... bit 12: RA13 bit 13: RFU
154	RAADEC_C	R	RAA Counter Decrement per REF Command for RFM level C (MR8 OP[5:4] = 11). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAADEC
[153:152]	RAAMMT_C[1:0]	R	RAA Maximum Management Threshold (RAAMMT) for RFM level C (MR8 OP[5:4] = 11). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAAMMT
[151:149]	RAAIMT_C[2:0]	R	RAA Initial Management Threshold (RAAIMT) for RFM level C (MR8 OP[5:4] = 11) The field shall be ignored when the ARFM bit is 0. Same encoding as in RAAIMT
148	RAADEC_B	R	RAA Counter Decrement per REF Command for RFM level B (MR8 OP[5:4] = 10). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAADEC
[147:146]	RAAMMT_B[1:0]	R	RAA Maximum Management Threshold (RAAMMT) for RFM level B (MR8 OP[5:4] = 10). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAAMMT
[145:143]	RAAIMT_B[2:0]	R	RAA Initial Management Threshold (RAAIMT) for RFM level B (MR8 OP[5:4] = 10). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAAIMT

Bit Position	Bit Field	Type	Description
142	RAADEC_A	R	RAA Counter Decrement per REF Command for RFM level A (MR8 OP[5:4] = 01). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAADEC
[141:140]	RAAMMT_A[1:0]	R	RAA Maximum Management Threshold (RAAMMT) for RFM level A (MR8 OP[5:4] = 01). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAAMMT
[139:137]	RAAIMT_A[2:0]	R	RAA Initial Management Threshold (RAAIMT) for RFM level A (MR8 OP[5:4] = 01). The field shall be ignored when the ARFM bit is 0 Same encoding as in RAAIMT
136	RAADEC	R	Default RAA Counter Decrement per REF Command. The field shall be ignored when the RFM bit is 0. 0 - 1.0 × RAAIMT 1 - 0.5 × RAAIMT
[135:134]	RAAMMT[1:0]	R	Default RAA Maximum Management Threshold (RAAMMT) The field shall be ignored when the RFM bit is 0. 00 - 3 × RAAIMT 01 - 4 × RAAIMT 10 - 5 × RAAIMT 11 - 6 × RAAIMT
[133:131]	RAAIMT[2:0]	R	Default RAA Initial Management Threshold (RAAIMT) The field shall be ignored when the RFM bit is 0. 000 - 32 001 - 40 010 - 48 011 - 56 100 - 64 101 - 72 110 - 80 111 - Reserved
130	DRFM	R	Directed Refresh Management (DRFM) 0 –Directed Refresh Management is not supported 1 –Directed Refresh Management is supported
129	ARFM	R	Adaptive Refresh Management (ARFM) 0 – Adaptive Refresh Management is not supported 1 – Adaptive Refresh Management is supported
128	RFM	R	Refresh Management (RFM) 0 - Refresh Management not required 1 - Refresh Management required
[127:120]	MANUFACTURING_YEAR[7:0]	R	Binary encoded year: 2020 = 00000000; 2024 = 00000100
[119:112]	MANUFACTURING_WEEK[7:0]	R	Binary encoded week: WW52 = 00110100

Bit Position	Bit Field	Type	Description
[111:48]	SERIAL_NO[63:0]	R	Unique device ID
[47:44]	MANUFACTURER_ ID[3:0]	R	0001 - Samsung 0110 - SK Hynix 1100 - CXMT 1111 - Micron All others - Reserved
[43:40]	DENSITY[3:0]	R	Memory density per channel (see 3.2 Channel Addressing) 0000 - 3 Gb 0001 - 6 Gb (24Gb 8-High) 0010 - 9 Gb (24Gb 12-High) 0011 - 12 Gb (24Gb 16-High) 0100 - 4 Gb 0101 - 8 Gb (32Gb 8-High) 0110 - 12 Gb (32Gb 12-High) 0111 - 16 Gb (32Gb 16-High)
[39:8]	CHANNEL_ AVAILABLE[31:0][1]	R	Channel Available 0 - Channel not present / not working 1 - Channel present / working Channel encoding (1 bit per channel): bit 8: channel 0 bit 9: channel 1 ... bit 38: channel 30 bit 39: channel 31
[7:0]	MODEL_PART_ NUMBER[7:0]	R	Vendor reserved

NOTE 1 A channel marked as “not present / not working” keeps all AWORD and DWORD input and output buffers permanently disabled and drives that channel’s WSO to LOW.

NOTE 2 The PPR_RVSD field will only be used in the future if additions to the addressing table increase the row address to include RA15. If future additions to the addressing table increase the bank, column, or pseudo channel address then the PPR_RSVD will remain at the default.

NOTE 3 HBM4 devices must output bits [174:170] even if the optional features are not used.

13.5.12 TEMPERATURE

This instruction captures the HBM4 DRAM's junction temperature. Temperature reporting is specified as a 9-bit field: bit 0 or LSB indicates the validity of the temperature sensor read-out, and the remaining 8 bits indicate the temperature in degrees Celsius.

Wrapper Data Register

When TEMPERATURE is the current instruction, the data register as shown in Table 127 is connected between WSI and WSO[31:0], and the WSO outputs of all active channels drive the same data during ShiftWR events.

CaptureWR

When TEMPERATURE is the current instruction, the CaptureWR event will load the temperature field values into the shift stage of the WDR.

UpdateWR

When TEMPERATURE is the current instruction, the UpdateWR event will have no effect.

Table 127 — TEMPERATURE Wrapper Data Register

Bit Position	Bit Field	Type	Description
[8:1]	TEMP[7:0]	R	Temperature in degrees Celsius. Examples: 8'b 0000 0000 = – 40 ºC 8'b 0010 1000 = 0 ºC 8'b 0100 0001 = 25 ºC 8'b 1010 0111 = 127 ºC
[0]	VALID	R	Temperature sensor output valid 0 – Invalid 1 – Valid

13.5.13 MODE_REGISTER_DUMP_SET

The MODE_REGISTER_DUMP_SET instruction provides read (dump) and write (set) access to the HBM4 Mode Registers.

Wrapper Data Register

When MODE_REGISTER_DUMP_SET is the current instruction, the data register as shown in Table 128 is connected between WSI and WSO.

CaptureWR

When MODE_REGISTER_DUMP_SET is the current instruction, the CaptureWR event will capture the Mode Register content into the shift stage of the WDR. Reserved Mode Registers and bit fields marked as "RFU" capture an 'X' value. The Mode Register content itself does not change with the CaptureWR event. A minimum waiting time of $t_{MRSS} = t_{MOD}$ must be observed between an MRS command and this Capture WR event.

UpdateWR

When MODE_REGISTER_DUMP_SET is the current instruction, the UpdateWR event will simultaneously load the bits from the shift stage to the mode registers. The updated mode register content will be valid for subsequent mission mode operation after t_{UPDMRS}.

Table 128 — MODE_REGISTER_DUMP_SET Wrapper Data Register

Bit Position[1]	Bit Field	Type	Description
[159:152]	MR19	R/W	MR19[7:0]
[151:144]	MR18	R/W	MR18[7:0]
[143:136]	MR17	R/W	MR17[7:0]
[135:128]	MR16	R/W	MR16[7:0]
[127:120]	MR15	R/W	MR15[7:0]
[119:112]	MR14	R/W	MR14[7:0]
[111:104]	MR13	R/W	MR13[7:0]
[103:96]	MR12	R/W	MR12[7:0]
[95:88]	MR11	R/W	MR11[7:0]
[87:80]	MR10	R/W	MR10[7:0]
[79:72]	MR9	R/W	MR9[7:0]
[71:64]	MR8	R/W	MR8[7:0]
[63:56]	MR7	R/W	MR7[7:0]
[55:48]	MR6	R/W	MR6[7:0]
[47:40]	MR5	R/W	MR5[7:0]
[39:32]	MR4	R/W	MR4[7:0]
[31:24]	MR3	R/W	MR3[7:0]

[23:16]	MR2	R/W	MR2[7:0]
[15:8]	MR1	R/W	MR1[7:0]
[7:0]	MR0	R/W	MR0[7:0]
NOTE 1 The maximum bit position for MODE_REGISTER_DUMP_SET is 160 bits but is vendor specific. Refer tosupplier datasheet for more information.			

13.5.14 READ_LFSR_COMPARE_STICKY

This instruction is used to capture the LFSR Compare Sticky error data to be shifted out on the WSO output. The instruction is associated with the AWORD and DWORD I/O loopback test features. Data register bit positions are specified in the Data Register section of this instruction in Table 129. Note that the content of the MISR and LFSR Compare Sticky error data registers is not specified after shifting out the sticky error content. The host should reinitialize the MISR registers (such as with MR7 Preset and AWORD_MISR_CONFIG preset) before continuing with additional testing. See section 6.8 HBM4 Loopback Test Modes for MISR mode features and usage.

While both the AWORD and DWORD sticky error bits share a common WDR, the bits are set and cleared only by their respective AWORD or DWORD Preset and LFSR Compare operations. For example, an AWORD_MISR_CONFIG preset operation clears the AWORD sticky error bits, and the state of the DWORD sticky error bits is undefined; therefore, the host should ignore the DWORD sticky error bits when operating the AWORD LFSR Compare mode. Conversely, the DWORD_MISR_CONFIG preset operation clears the DWORD sticky error bits, and the state of the AWORD sticky error bits is undefined; therefore, the host should ignore the AWORD sticky error bits when operating the DWORD LFSR Compare mode.

Wrapper Data Register

When READ_LFSR_COMPARE_STICKY is the current instruction, the data register as shown in Table 129 is connected between WSI and WSO.

CaptureWR

When READ_LFSR_COMPARE_STICKY is the current instruction, the CaptureWR event will load the sticky error values into the shift stage of the WDR.

UpdateWR

When READ_LFSR_COMPARE_STICKY is the current instruction, the UpdateWR event will have no effect.

13.5.14 READ_LFSR_COMPARE_STICKY (cont'd)

Table 129 — READ_LFSR_COMPARE_STICKY Wrapper Data Register

Bit Position	Bit Field	Type	Description
[98:59]	DWORD1	R	DWORD1: DQ[63:32], DBI[7:4], ECC[3:2] and SEV[3:2] (same ordering as DWORD0)
58	AWORD_R1	R	AWORD
57	AWORD_R2	R	
56	AWORD_R3	R	
55	AWORD_R0	R	
54	AWORD_R4	R	
53	AWORD_R5	R	
52	AWORD_R6	R	
51	AWORD_R7	R	
50	AWORD_R8	R	
49	AWORD_R9	R	
48	AWORD_ARFU	R	
47	AWORD_C7	R	
46	AWORD_C6	R	
45	AWORD_C5	R	
44	AWORD_C4	R	
43	AWORD_C3	R	
42	AWORD_C2	R	
41	AWORD_C1	R	
40	AWORD_C0	R	
[39:30]	DWORD0_BYTE_3	R	Byte 3 of DWORD0: DQ[31:24], DBI3 and SEV1 (same ordering as Byte 0)
[29:20]	DWORD0_BYTE_2	R	Byte 2 of DWORD0: DQ[23:16], DBI2 and SEV0 (same ordering as Byte 0)
[19:10]	DWORD0_BYTE_1	R	Byte 1 of DWORD0: DQ[15:8], DBI1 and ECC1 (same ordering as Byte 0)
9	DWORD0_DBI0	R	Byte 0 of DWORD0 (PC0)
8	DWORD0_DQ7	R	
7	DWORD0_DQ6	R	
6	DWORD0_DQ5	R	
5	DWORD0_DQ4	R	
4	DWORD0_DQ3	R	
3	DWORD0_DQ2	R	
2	DWORD0_DQ1	R	
1	DWORD0_DQ0	R	
0	DWORD0_ECC0	R	

13.5.15 SOFT_LANE_REPAIR and HARD_LANE_REPAIR

SOFT_LANE_REPAIR and HARD_LANE_REPAIR instructions can only be issued as part of the device initialization and before normal memory operation has commenced, e.g. before the CK clock has started to toggle.

Wrapper Data Register

When either SOFT_LANE_REPAIR or HARD_LANE_REPAIR is the current instruction, the LANE_REPAIR wrapper data register as shown in Table 130 is connected between WSI and WSO.

Figure 107 illustrates the interaction between SOFT_LANE_REPAIR and HARD_LANE_REPAIR instructions and the associated registers. It is pointed out that the actual I/O lane remapping is derived from the content of the lane repair shadow register.

CaptureWR

When either SOFT_LANE_REPAIR or HARD_LANE_REPAIR is the current instruction, the CaptureWR event will load the lane remapping data from the lane repair shadow register into the shift stage of the WDR. This internal lane repair shadow register is pre-loaded with the repair data from a preceding HARD_LANE_REPAIR operation upon HBM4 DRAM initialization (RESET_n pulled Low). The memory controller may use these data to configure the lane repair accordingly at the host; it may also use these data as a seed value for subsequent lane repair operations.

UpdateWR

When SOFT_LANE_REPAIR is the current instruction, the UpdateWR event will load the lane remapping data from the shift stage of the WDR into the lane repair shadow register and force the I/O lanes to be remapped accordingly. This remapping is non-persistent; it will be lost when RESET_n is pulled low or the device loses power. Pulling WRST_n low does not reset the lane repair shadow register.

When HARD_LANE_REPAIR is the current instruction, the UpdateWR event will load the lane remapping data from the shift stage of the WDR into the hard lane repair register. The controller must wait t_{HLREP} to allow the HBM4 DRAM to complete this operation and permanently store the repair vector.

Only a single broken lane can be repaired at a time, in order to limit the current constraint of the associated circuits. If multiple lanes are to be repaired, it is required to shift in the repair vectors for each broken lane sequentially, with all other lane repair setting = Fh, and initiate each actual lane repair with a separate UpdateWR event.

The UpdateWR event itself does not lead to an actual re-mapping of the I/O lanes. For such re-mapping to get effective it is required to initiate a chip reset by pulling RESET_n LOW for at least t_{PW_RESET}, which copies the repair vector from the hard lane repair register into the lane repair shadow register as shown in Figure 107.

13.5.15 SOFT_LANE_REPAIR and HARD_LANE_REPAIR (cont'd)

Table 130 — LANE_REPAIR Wrapper Data Register

Bit Position	Bit Field	Type	Description
[44:40]	WSO[4:0]	R/W	Lane remapping applied to WSO 0h – Fh: WSO0 - WSO15 or WSO16 - WSO31 10h – 1Eh: Reserved 1Fh: No lane remapping (default) Channel 0 for repair of WSO0 - 15 Channel 16 for repair of WSO16 - 31 Other channels (reserved bits with default 11111)
[39:36]	DWORD1_BYTE3[3:0]	R/W	Lane remapping applied to DWORD1 byte 3 0h: SEV3 1h - 8h: DQ56 - DQ63 9h: DBI7 Ah - Eh: Reserved Fh: No lane remapping (default)
[35:32]	DWORD1_BYTE2[3:0]	R/W	Lane remapping applied to DWORD1 byte 2 0h: SEV2 1h - 8h: DQ48 - DQ55 9h: DBI6 Ah - Eh: Reserved Fh: No lane remapping (default)
[31:28]	DWORD1_BYTE1[3:0]	R/W	Lane remapping applied to DWORD1 byte 1 0h: ECC3 1h - 8h: DQ40 - DQ47 9h: DBI5 Ah - Eh: Reserved Fh: No lane remapping (default)
[27:24]	DWORD1_BYTE0[3:0]	R/W	Lane remapping applied to DWORD1 byte 0 0h: ECC2 1h - 8h: DQ32 - DQ39 9h: DBI4 Ah - Eh: Reserved Fh: No lane remapping (default)
[23:20]	AWORD_RA[3:0]	R/W	Lane remapping applied to AWORD. 0h – 9h: R0 – R9 Ah – Eh: Reserved Fh: No lane remapping (default)
[19:16]	AWORD_CA[3:0]	R/W	Lane remapping applied to AWORD. 0h – 7h: C0 – C7 8h: APAR 9h: ARFU Ah – Eh: Reserved Fh: No lane remapping (default)

[15:12]	DWORD0_BYTE3[3:0]	R/W	Lane remapping applied to DWORD 0 byte 3 0h: SEV1 1h – 8h: DQ24 – DQ31 9h: DBI3 Ah – Eh: Reserved Fh: No lane remapping (default)
[11:8]	DWORD0_BYTE2[3:0]	R/W	Lane remapping applied to DWORD0 byte 2 0h: SEV0 1h - 8h: DQ16 - DQ23 9h: DBI2 Ah - Eh: Reserved Fh: No lane remapping (default)
[7:4]	DWORD0_BYTE1[3:0]	R/W	Lane remapping applied to DWORD0 byte 1 0h: ECC1 1h - 8h: DQ8 - DQ15 9h: DBI1 Ah - Eh: Reserved Fh: No lane remapping (default)
[3:0]	DWORD0_BYTE0[3:0]	R/W	Lane remapping applied to DWORD0 byte 0 0h: ECC0 1h - 8h: DQ0 - DQ7 9h: DBI0 Ah - Eh: Reserved Fh: No lane remapping (default)

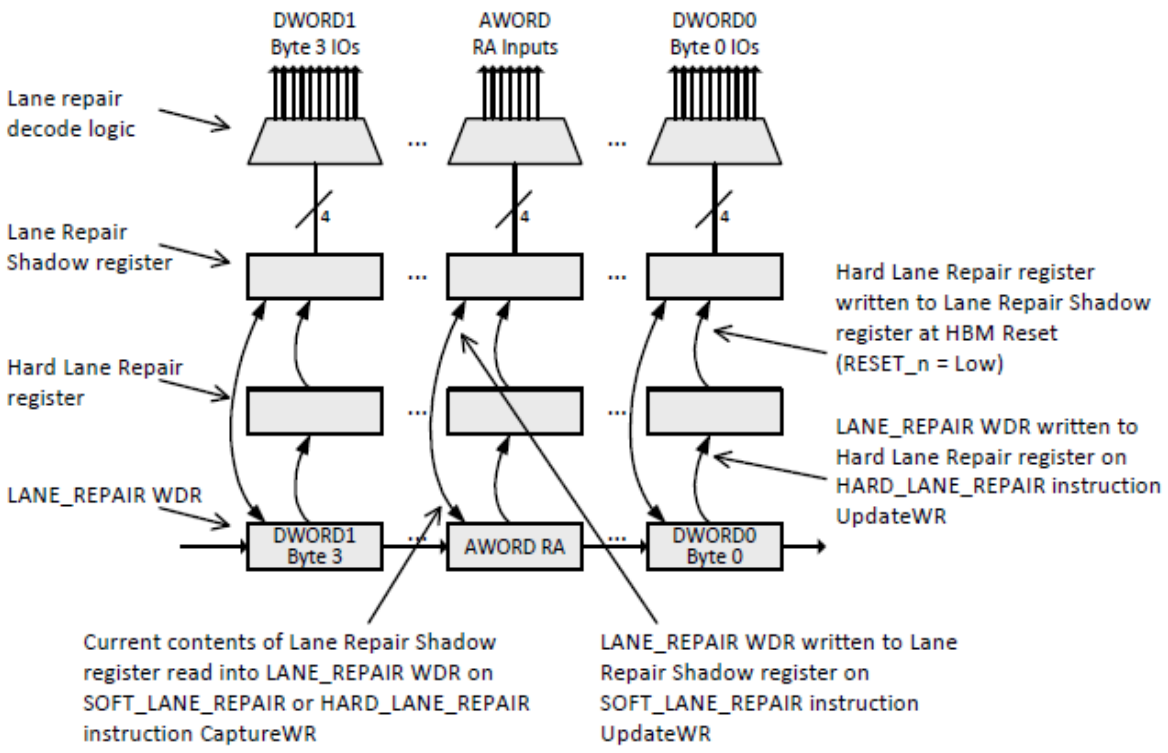


Figure 107 — Registers Associated with Lane Repair Instructions

13.5.16 CHANNEL_DISABLE

This instruction disables one channel or all channels specified in WIR[13:8]. The instruction may only be issued after t_{INIT3} timing has been met and only before the CK clock is started for the first time. The disabled channel will transition into a safe low-power state where they do not respond to commands. All AWORD and DWORD input and output buffers will be turned off, thus allowing all external signals to float. It will also not respond to EXTEST_RX, EXTEST_TX and CHANNEL_ID instructions. Both RESET_n and WRST_n must be maintained HIGH during the all-channel disable state. The channel's WSO output will remain active and drive a LOW.

The all-channel disable option shall be used during host-side HTOL and/or other reliability tests to prevent unintentional degradation of the HBM4 DRAM. This option may only be issued after tINIT3 timing has been met, before the CK clock is started for the first time after exit from reset state and only if no other IEEE1500 instruction other than BYPASS or DEVICE_ID has been loaded before.

A disabled channel can be enabled again by pulling both RESET_n and WRST_n to LOW and following the procedure described in the Initialization Sequence with Stable Power section.

Wrapper Data Register

When CHANNEL_DISABLE is the current instruction, the data register as shown in Figure 108. Table 131 is connected between WSI and WSO.

CaptureWR

When CHANNEL_DISABLE is the current instruction, the CaptureWR event will have no effect.

UpdateWR

When CHANNEL_DISABLE is the current instruction, the UpdateWR event will load the disable bit from the shift stage into the update stage of the WDR and asynchronously transition into a safe low power state when the bit is 1. Input and output buffers will be disabled latest after t_{CHDIS}.

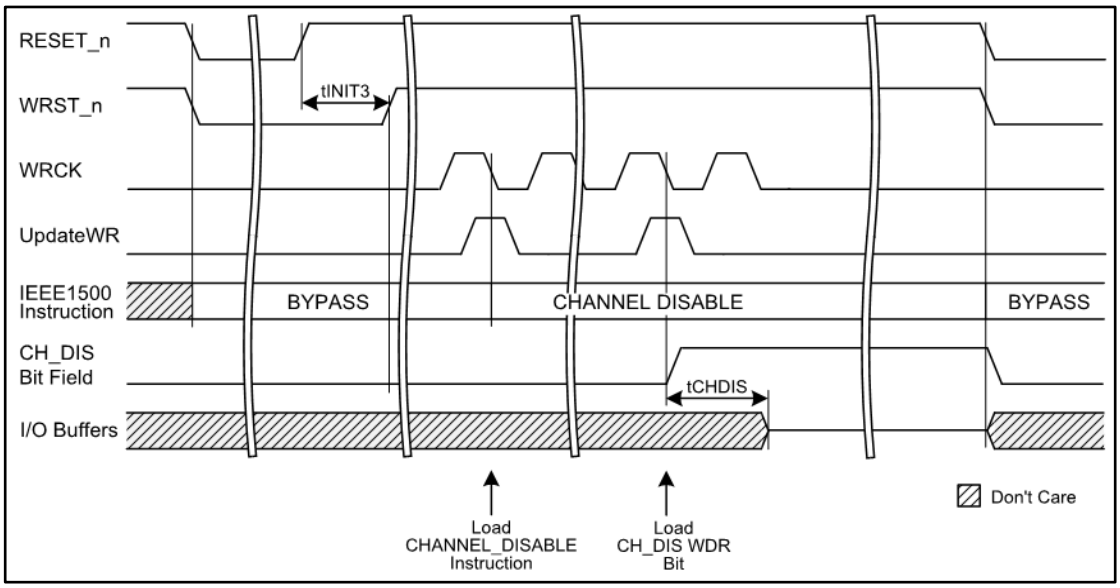


Figure 108 — Channel Disable Instruction

13.5.16 CHANNEL_DISABLE (cont'd)

Table 131 — CHANNEL_DISABLE Wrapper Data Register

Bit Position	Bit Field	Type	Description
0	CH_DIS	W	0 – Channel is enabled (default). Clearing the bit to 0 does not re-enable a channel. 1 – Channel is disabled

13.5.17 CHANNEL TEMPERATURE

This instruction captures the channel's junction temperature. Temperature reporting is specified as a 9-bit field per SID: the LSB indicates the validity of the temperature sensor read-out, and the remaining 8 bits indicate the temperature in degrees Celsius.

Wrapper Data Register

When CHANNEL_TEMPERATURE is the current instruction, the data register as shown in Table 132 is connected between WSI and WSO.

CaptureWR

When CHANNEL_TEMPERATURE is the current instruction, the CaptureWR event load the temperature field values into the shift stage of the WDR.

UpdateWR

When CHANNEL TEMPERATURE is the current instruction, the UpdateWR event will have no effect.

13.5.17 CHANNEL TEMPERATURE (cont'd)

Table 132 — CHANNEL_TEMPERATURE Wrapper Data Register

Bit Position	Bit Field	Type	Description
[35:28]	CHANNEL_SID3_ TEMP[7:0]	R	Maximum temperature per channel for SID3 in Degrees Celsius. Same encoding as in CHANNEL_SID0_TEMP[7:0]
27	CHANNEL_SID3_ VALID	R	Channel Temperature sensor output valid for SID3 0 – Invalid 1 – Valid
[26:19]	CHANNEL_SID2_ TEMP[7:0]	R	Maximum temperature per channel for SID2 in Degrees Celsius. Same encoding as in CHANNEL_SID0_TEMP[7:0]
18	CHANNEL_SID2_ VALID	R	Channel Temperature sensor output valid for SID2 0 – Invalid 1 – Valid
[17:10]	CHANNEL_SID1_ TEMP[7:0]	R	Maximum temperature per channel for SID1 in Degrees Celsius. Same encoding as in CHANNEL_SID0_TEMP[7:0]
9	CHANNEL_SID1_ VALID	R	Channel Temperature sensor output valid for SID1 0 – Invalid 1 – Valid
[8:1]	CHANNEL_SID0_ TEMP[7:0]	R	Maximum temperature per channel for SID0 in Degree Celsius. Examples: 8'b 0000 0000 = – 40 ºC 8'b 0010 1000 = 0 ºC 8'b 0100 0001 = 25 ºC 8'b 1010 0111 = 127 ºC
0	CHANNEL_SID0_ VALID	R	Channel Temperature sensor output valid for SID0 0 – Invalid 1 – Valid

NOTE 1 For unsupported SID fields the HBM4 DRAM will report the sensor output as invalid and the temperature as all zeros.

NOTE 2 Device may report the same maximum temperature value for all the channels within the same core-die or individual channel temperatures based on number of unique temperature sensors (Figure 109)

NOTE 3 If a channel is distributed across multiple dies (within the same SID), as shown in Figure 110, the device reports the maximum channel temperature value between the core-dies across which the channel is distributed

13.5.17 CHANNEL TEMPERATURE (cont'd)

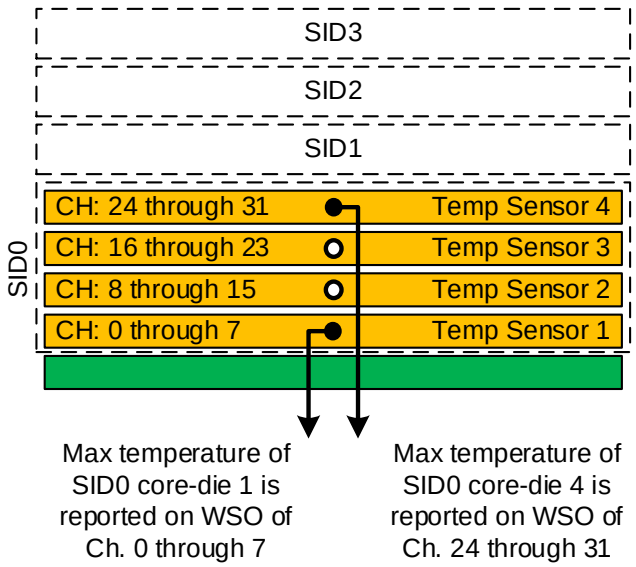


Figure 109 — Example Channel Configuration 1

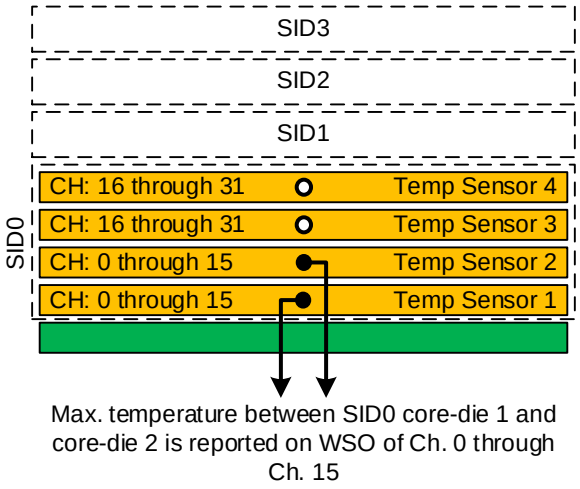


Figure 110 — Example Channel Configuration 2

13.5.18 WOSC_RUN and WOSC_COUNT

The WOSC_RUN and WOSC_COUNT instructions are associated with the WDQS Interval Oscillator in the HBM4 DRAM.

Wrapper Data Register

When WOSC_RUN is the current instruction, the WOSC_RUN wrapper data register as shown in Table 133 is connected between WSI and WSO[31:0], and the WSO outputs of all active channels drive the same data during ShiftWR events.

When WOSC_COUNT is the current instruction, the WOSC_COUNT wrapper data register as shown in Table 134 is connected between WSI and WSO[31:0], and the WSO outputs of all active channels drive the same data during ShiftWR events.

CaptureWR

When WOSC_RUN is the current instruction, the CaptureWR event will have no effect.

When WOSC_COUNT is the current instruction, the CaptureWR event will load the WDQS oscillator count value into the shift stage of the WDR and update the WOSC_COUNT_VALID field of the WDR.

UpdateWR

When WOSC_RUN is the current instruction, the UpdateWR event will load the value from the shift stage into the update stage and start or stop the WDQS Interval Oscillator.

When WOSC_COUNT is the current instruction, the UpdateWR event will have no effect.

Table 133 — WOSC_RUN Wrapper Data Register

Bit Position	Bit Field	Type	Description
0	WOSC_START_STOP	W	0 – Stop WDQS Interval Oscillator (default) 1 – Start WDQS Interval Oscillator

Table 134 — WOSC_COUNT Wrapper Data Register

Bit Position	Bit Field	Type	Description
[24:1]	WOSC_COUNT_VALUE	R	Oscillator count value. Range 0 to 2^{24}-1
[0]	WOSC_COUNT_VALID	R	0 – Count is invalid (default) 1 – Count is valid
NOTE 1 The WDQS oscillator count value is used to train WDQS to the data valid window. The value reported in this WDR can be used by the memory controller to periodically adjust the phase of WDQS relative to data. NOTE 2 The contents of bits [23:0] is reset by starting the oscillator.			

13.5.19 ECS Error Log

This instruction is used to capture ECS Error Log information reading from the WDR. The registers in this instruction are associated with the Error type and Error address(position) as the ECS operation result during ECS period. The error types are NE, CE_S, CE_M, UE. The encoding of error types is the same as burst position 4~7 of severity transmission in Table 67. See section of Error Check and Scrub(ECS). HBM4 device will store detailed information about errors detected during ECS operation period. The latest logging information will follow priority update rule in the following order such as UE, CE_M, CE_S, NE of Table 65. In case of CE_S, error information would be logged in HBM4 for host-polling when #ERRECS during current or previous ECS period is larger than the #ERRTH, where the #ERRECS means accumulated the number of errors events detected during previous ECS period, not the number of error bits. At this time the error address will be logged and readout through IEEE1500 WSO. The registers logged ECS error information will be cleared by host.

Wrapper Data Register

When the ECS Error Log instruction is updated the data register as shown in Table 135 is connected between WSI and WSO.

CaptureWR

When ECS Error Log is the current instruction, the CaptureWR event will load the respective error log field values into the shift stage of the WDR.

UpdateWR

When ECS Error Log is the current instruction, the UpdateWR event will have no effect.

13.5.19 ECS Error Log (cont'd)

Table 135 — ECS Error Log Wrapper Data Register

Bit Position	Bit Field	Type	Description
[215:190]	SID3_PC1_ECS[25:0]	R	ECS error log for SID3, PC1 Same encoding as in SID0_PC0_ECS
189	SID3_PC1_ECS_VALID	R	ECS error log valid of SID3, PC1 Same encoding as in SID0_PC0_ECS_VALID
[188:163]	SID3_PC0_ECS[25:0]	R	ECS error log for SID3, PC0 Same encoding as in SID0_PC0_ECS
162	SID3_PC0_ECS_VALID	R	ECS error log valid of SID3, PC0 Same encoding as in SID0_PC0_ECS_VALID
[161:136]	SID2_PC1_ECS[25:0]	R	ECS error log for SID2, PC1 Same encoding as in SID0_PC0_ECS
135	SID2_PC1_ECS_VALID	R	ECS error log valid of SID2, PC1 Same encoding as in SID0_PC0_ECS_VALID
[134:109]	SID2_PC0_ECS[25:0]	R	ECS error log for SID2, PC0 Same encoding as in SID0_PC0_ECS
108	SID2_PC0_ECS_VALID	R	ECS error log valid of SID2, PC0 Same encoding as in SID0_PC0_ECS_VALID
[107:82]	SID1_PC1_ECS[25:0]	R	ECS error log for SID1, PC1 Same encoding as in SID0_PC0_ECS
81	SID1_PC1_ECS_VALID	R	ECS error log valid of SID1, PC1 Same encoding as in SID0_PC0_ECS_VALID
[80:55]	SID1_PC0_ECS[25:0]	R	ECS error log for SID1, PC0 Same encoding as in SID0_PC0_ECS
54	SID1_PC0_ECS_VALID	R	ECS error log valid of SID1, PC0 Same encoding as in SID0_PC0_ECS_VALID
[53:28]	SID0_PC1_ECS[25:0]	R	ECS error log for SID0, PC1 Same encoding as in SID0_PC0_ECS
27	SID0_PC1_ECS_VALID	R	ECS error log valid of SID0, PC1 Same encoding as in SID0_PC0_ECS_VALID
[26:1]	SID0_PC0_ECS[25:0]	R	ECS error log for SID0, PC0 [25:22] – Bank address BA[3:0] [21:7] – Row address RA[14:0][1] [6:2] – Column address CA[4:0] [1:0] – Error Type[1:0] 00b: NE 01b: CEs 10b: UE 11b: CEm
0	SID0_PC0_ECS_VALID	R	ECS error log valid for SID0, PC0 0 – Invalid (default) 1 – Valid

NOTE 1 Address bit RA14 is not defined for 24 Gb die densities and shall be kept LOW in the ECS Error Log output.

13.5.20 HS_REP_CAP

The HS_REP_CAP instruction tells the host whether a bank(s) has resources for either hard repair or soft repair. For each bank of the DRAM, the Gray-coded encoding indicates whether there are no resources, 1 resource or 2 or more resources.

This instruction shall be used by the host before performing any hard of soft repair. This register will be updated with the completion of a hard repair only. The DRAM may also share resources with self repair. In this case the completion of self repair will also update this register. The sharing of resources between hard/soft repair and self repair is vendor specific and identified in the SHARED_REP_RES field of the DEVICE_ID.

Wrapper Data Register

When HS_REP_CAP is the current instruction, the wrapper data register as shown in Table 136 is connected between WSI and WSO.

CaptureWR

When HS_REP_CAP is the current instruction, the CaptureWR event will load the resource field value into the shift stage register.

UpdateWR

When HS_REP_CAP is the current instruction, the UpdateWR event will have no effect.

Table 136 — HS_REP_CAP Wrapper Data Register

Bit Position	Bit Field	Type	Description
[255:192]	HS_REPAIR_RES_SID3	R	SID3 PC0 Banks [15:0] and PC1 Banks [15:0]
[191:128]	HS_REPAIR_RES_SID2	R	SID2 PC0 Banks [15:0] and PC1 Banks [15:0]
[127:64]	HS_REPAIR_RES_SID1	R	SID1 PC0 Banks [15:0] and PC1 Banks [15:0]
[63:32]	HS_REPAIR_RES_SID0	R	SID0 PC1 Banks [15:0]
[31:2]			SID0 PC0 Banks [15:1]
[1:0]			Resources available for SID0 PC0 Bank 0 00b: No resource available 01b: 1 resource 10b: Reserved 11b: 2 or more resources

13.5.21 SELF_REP and SELF_REP_RESULTS

The SELF_REP and SELF_REP_RESULTS instructions are associated with HBM4 self repair.

Wrapper Data Register

When SELF_REP is the current instruction, the SELF_REP wrapper data register as shown in Table 137 is connected between WSI and WSO.

When SELF_REP_RESULTS is the current instruction, the SELF_REP_RESULTS wrapper data register as shown in Table 138 is connected between WSI and WSO.

CaptureWR

When SELF_REP is the current instruction, the CaptureWR event will capture the SR_PROGRESS to the shift stage register.

When SELF_REP_RESULTS is the current instruction, the CaptureWR event will capture the SID[3:0]_RESULTS to the shift stage register.

UpdateWR

When SELF_REP is the current instruction, the UpdateWR event will copy the SELFR_REF_RATE, SID-SELECT and REP_TYPE into the update stage register.

When SELF_REP_RESULTS is the current instruction, the UpdateWR event will have no effect.

13.5.21 SELF_REP and SELF_REP_RESULTS (cont'd)

Table 137 — SELF_REP Wrapper Data Register

Bit Position	Bit Field	Type	Description
[8]	SHARED_OVERRIDE[2]	W	0b: DRAM must leave 1 resource when resources shared (default) 1b: DRAM can use all resources for Self-repair when shared
[7:6]	SELFR_REP_RATE	W	00b: 1 x tREFI 01b: 0.5 x tREFI 10b: 0.25 x tREFI 11b: Reserved
[5:4]	SID_SELECT	W	00b: SID0 01b: SID1 10b: SID2 11b: SID3
[3:2]	REP_TYPE	W	11b: Run self-test and auto-repair 10b: Auto-repair only 01b: Run self-test only and no auto-repair 00b: Disabled/Cancel[1]
[1:0]	SR_PROGRESS	R	00b: Not running (default) 01b: Self-test in progress 10b: Auto-repair in progress 11b: Complete

NOTE 1 Cancel will only stop the SELF_REPAIR when the self-test is in progress (SR_PROGRESS = 01b) and SR_PROGRESS is set to 00b (not running) after cancel.
NOTE 2 When resources are shared between self and hard/soft repair the SHARED_OVERRIDE field allows the DRAM to use all the resources for self repair. When the resources are not shared the field is do not care.

Table 138 — SELF_REP_RESULTS Wrapper Data Register

Bit Position	Bit Field	Type	Description
[7:6]	SID3_RESULTS	R	SID3 SELF_REP results
[5:4]	SID2_RESULTS	R	SID2 SELF_REP results
[3:2]	SID1_RESULTS	R	SID1 SELF_REP results
[1:0]	SID0_RESULTS	R	SID0 SELF_REP results 00b: SELF_REP test has not run since INIT or No fails remain after most recent run 01b: Fail(s) remain 10b: Unrepairable fail(s) remain 11b: SELF_REP should be run again

NOTE 1 For unsupported SID fields, the DRAM will report the results as 00b (SELF-REPAIR test has not run).
NOTE 2 DRAM may report the same SELF_REP_RESULTS value for up to 16 channels, 8 channels or individual channel results as defined in Table 112 — WIR Channel Selection Definition.

13.6 Interaction with Mission Mode Operation

Table 139 defines the interaction of the various IEEE1500 instructions with mission mode operation, and any instruction exit requirements (see also Table 114 for all IEEE1500 instructions).

Table 139 — IEEE1500 Port Instruction Interactions

Instruction	Interaction with Mission Mode	Post Instruction Requirements
BYPASS DWORD_MISR [1] AWORD_MISR [1] READ_LFSR_COMPARE_STICKY [1] DEVICE_ID TEMPERATURE MODE_REGISTER_DUMP_SET (dump) CHANNEL_TEMPERATURE WOSC_RUN WOSC_COUNT ECS_ERROR_LOG SELF_REP_RESULTS HS_REP_CAP	Instructions may be used at any time. Core memory content is retained if refresh specifications are met.	None
SOFT_REPAIR [2, 5] SOFT_LANE_REPAIR [2] MODE_REGISTER_DUMP_SET (set) AWORD_MISR_CONFIG [3]	Core memory content is retained if refresh specifications are met.	Meet IEEE1500 Port AC Timings (see Table 140)
EXTEST_RX, EXTEST_TX MBIST CHANNEL_ID HARD_REPAIR [4] HARD_LANE_REPAIR [4] CHANNEL_DISABLE [6] SELF_REP [4]	HBM4 interface state and core memory content are not defined.	Reset

NOTE 1 While accessing these MISR-related registers has no interaction with mission mode, operating the AWORD and DWORD MISR modes may result in memory content loss unless the channel is put into self refresh mode. See 6.8 HBM4 Loopback Test Modes.

NOTE 2 Soft memory array and lane repairs imply that memory content is at least partially incorrect. While the HBM4 DRAM imposes no restrictions on the interface state and memory content, the host should consider the health of The memory content based on the repair(s) being applied.

NOTE 3 See HBM4 Loopback Test Modes for proper sequencing of the AWORD MISR test modes.

NOTE 4 Self/Hard memory array and hard lane repairs involve blowing fuses. Normal operation on any channel is not supported when the self/hard repair operations are used. A chip reset with RESET_n pulled LOW is required after self/hard repair operations before returning the HBM4 DRAM to normal operation.

NOTE 5 For SOFT_REPAIR it is required that the channel is held in bank idle state from the time when the SOFT_REPAIR instruction is loaded in the WIR until the SOFT_REPAIR instruction is unloaded.

NOTE 6 In case of all channel operation, refer to 13.5.16 CHANNEL_DISABLE instruction for conditions to re-enable a disabled channel

13.7 IEEE1500 Test Port AC Timing Parameters

Table 140 — IEEE1500 Test Port AC Timings

Parameter	Symbol	Values		Unit	Notes
		Min	Max		
IEEE1500 Port I/O Timings					
WRCK clock period	t_{CKTP}	20	-	ns	
WRCK clock high pulse width	t_{CKTPH}	0.45	-	t_{CKTP}	
WRCK clock low pulse width	t_{CKTPL}	0.45	-	t_{CKTP}	
WRST_n pulse width low	t_{WRSTL}	100	-	ns	
IEEE1500 port operation after WRST_n deassertion	t_{WINIT1}	3	-	t_{CKTP}	
Rising WRST_n edge to WRCK setup time	t_{SWRST}		-	ns	
WSP input setup time to WRCK rising edge	t_{SR}		-	ns	3
WSP input hold time from WRCK rising edge	t_{HR}		-	ns	3
WSP input setup time to WRCK falling edge	t_{SF}		-	ns	4
WSP input hold time from WRCK falling edge	t_{HF}		-	ns	4
WSO output valid time from WRCK falling edge	t_{OVWSO}	-		ns	5
EXTEST_RX Instruction Related Timings					
Input setup time to WRCK rising edge	t_{SEXT}			ns	6
Input hold time from WRCK rising edge	t_{HEXT}			ns	6
EXTEST_TX Instruction Related Timings					
Output valid time from WRCK rising edge	t_{OVEXT}	-		ns	7
HBM4_RESET Instruction Related Timings					
HBM_RESET instruction minimum active time	t_{RES}	t_{PW_RESET}	-	ns	8
SOFT_REPAIR and HARD_REPAIR Instruction Related Timings					
SOFT_REPAIR minimum waiting time	t_{SREP}		-	ns, µs or nWRCK	9
HARD_REPAIR minimum waiting time	t_{HREP}		-	ns, µs or nWRCK	10
DWORD_MISR and AWORD_MISR Instruction Related Timings					
DWORD and AWORD MISR data capture to WDR data capture delay	t_{SMISR}		-	ns	11
CHANNEL_ID Instruction Related Timings					
Output high time from WRCK falling edge	t_{OVCHN}	-		ns	12
Output return to default state delay	t_{OZCHN}	-		ns	13

Parameter	Symbol	Values		Unit	Notes
		Min	Max		
Mode_Register_Dump_Set Instruction Related Timings					
WDR update to Mode Register valid delay	t_{UPDMRS}		-	ns	14
MRS command to WDR data capture delay	t_{MRSS}	t_{MOD}	-	nCK	15
AWORD_MISR_CONFIG Instruction Related Timings					
AWORD MISR configuration to MISR operation delay	t_{CMISR}		-	ns	16
SOFT_LANE_REPAIR and HARD_LANE_REPAIR Instruction Related Timings					
SOFT_LANE_REPAIR minimum waiting time	t_{SLREP}		-	ns or µs	17
HARD_LANE_REPAIR minimum waiting time	t_{HLREP}		-	ns or µs	18
CHANNEL_DISABLE Instruction Related Timing					
Channel disable to input and output buffer disable delay	t_{CHDIS}			ns or µs	19
SELF_REPAIR Instruction Related Timings					
Self-test and Auto-repair test time	t_{SELF_HEAL}	-		s	20
Auto-repair test time	t_{SELF_REP}	-		s	20
Self-test test time	t_{SELF_NR}	-		s	20
Self-test cancel time	t_{SELF_CANCEL}	-		us	20

NOTE 1 AC timing parameters apply to each channel of the HBM4 device independently except for timings related to IEEE1500 input pins that are common to all channels. No timing parameters are specified across channels, and all channels operate independently of each other.
NOTE 2 All parameters assume proper device initialization.
NOTE 3 Parameter applies to WSI, SelectWIR, ShiftWR, and CaptureWR inputs.
NOTE 4 Parameter applies to UpdateWR input.
NOTE 5 Parameter applies to WSO output changes resulting from Wrapper Instruction Register (WIR), Wrapper Bypass Register (WBY) or any Wrapper Data Register (WDR) shift operation.
NOTE 6 Parameter applies to all HBM4 inputs and bidirectional IOs in the CaptureWR cycle when the active instruction is EXTEST_RX.
NOTE 7 Parameter applies to all HBM outputs and bidirectional IOs in the ShiftWR cycle when the active instruction is EXTEST_TX.
NOTE 8 Parameter applies when the active instruction is HBM_RESET; it is measured from either the falling WRCK edge that loads the HBM_RESET instruction in the UpdateWIR cycle (in case no WDR is associated with the instruction) or the falling WRCK edge that sets the WDR bit to '1' in the UpdateWR cycle (in case a WDR is associated with the instruction) until either the HBM_RESET instruction is invalidated or the WDR bit is set back to '0'. The minimum value equals the RESET_n minimum low time with stable power (t_{PW_RESET}).
NOTE 9 Parameter applies when the active instruction is SOFT_REPAIR; it describes the minimum time for the HBM4 device to perform the internal soft repair; it is measured from the falling WRCK edge that loads the repair vector and repair start bit in the UpdateWR cycle until the instruction is invalidated.
NOTE 10 Parameter applies when the active instruction is HARD_REPAIR; it describes the minimum time for the HBM4 device to perform the internal hard repair; it is measured from the falling WRCK edge that loads the repair vector and repair start bit in the UpdateWR cycle until the instruction is invalidated.
NOTE 11 Parameter applies when the active instruction is DWORD_MISR or AWORD_MISR; it is measured from the last CK clock that updates the data in the respective MISR until the rising WRCK edge associated with the CaptureWR cycle that copies the MISR data into the WDR shift register.

NOTE 12	Parameter applies when the active instruction is CHANNEL_ID; it describes the maximum duration from either the falling WRCK edge that sets the CHANNEL_ID instruction in the UpdateWIR cycle (when no WDR is associated with the instruction) or the falling WRCK edge in the UpdateWR cycle that sets the enable bit in the WDR to '1' (when a WDR is associated with the instruction) until the bidirectional IOs drive a High.
NOTE 13	Parameter applies when the active instruction is CHANNEL_ID; it describes the maximum duration from either the falling WRCK edge that sets any instruction other than CHANNEL_ID instruction in the UpdateWIR cycle (when no WDR is associated with the instruction) or the falling WRCK edge in the UpdateWR cycle that sets the enable bit in the WDR to '0' (when a WDR is associated with the instruction) until the bidirectional IOs return to their default state.
NOTE 14	Parameter applies when the active instruction is MODE_REGISTER_DUMP_SET; it describes the minimum required delay between the falling WRCK edge in the UpdateWR cycle that loads the Mode Registers from the WDR shift register until any valid command other than RNOP and CNOP can be issued at the command interface.
NOTE 15	Parameter applies when the active instruction is MODE_REGISTER_DUMP_SET; it describes the minimum required delay between the last MRS command that loads any Mode Register and the rising WRCK edge in the CaptureWR cycle that copies the Mode Register content into the WDR shift register.
NOTE 16	Parameter applies when the active instruction is AWORD_MISR_CONFIG; it describes the minimum required delay between the falling WRCK edge in the UpdateWR cycle that loads the AWORD MISR configuration until the MISR configuration is valid for any subsequent AWORD or DWORD MISR operation in the CK clock domain.
NOTE 17	Parameter applies when the active instruction is SOFT_LANE_REPAIR; it describes the minimum time for the HBM4 device to perform the internal soft lane repair; it is measured from the falling WRCK edge that loads the repair vector in the UpdateWR cycle until the instruction is invalidated.
NOTE 18	Parameter applies when the active instruction is HARD_LANE_REPAIR; it describes the minimum time for the HBM4 device to perform the internal hard lane repair; it is measured from the falling WRCK edge that loads the repair vector in the UpdateWR cycle until the instruction is invalidated.
NOTE 19	Parameter applies when the active instruction is CHANNEL_DISABLE; it describes the minimum waiting time for disabling the channel's input and output buffers after the UpdateWR event that sets the CHANNEL_DISABLE WDR to 1.
NOTE 20	Parameter represents the maximum time the operation will take to complete and therefore the minimum time the host must wait.

13.7 IEEE1500 Test Port AC Timing Parameters (cont'd)

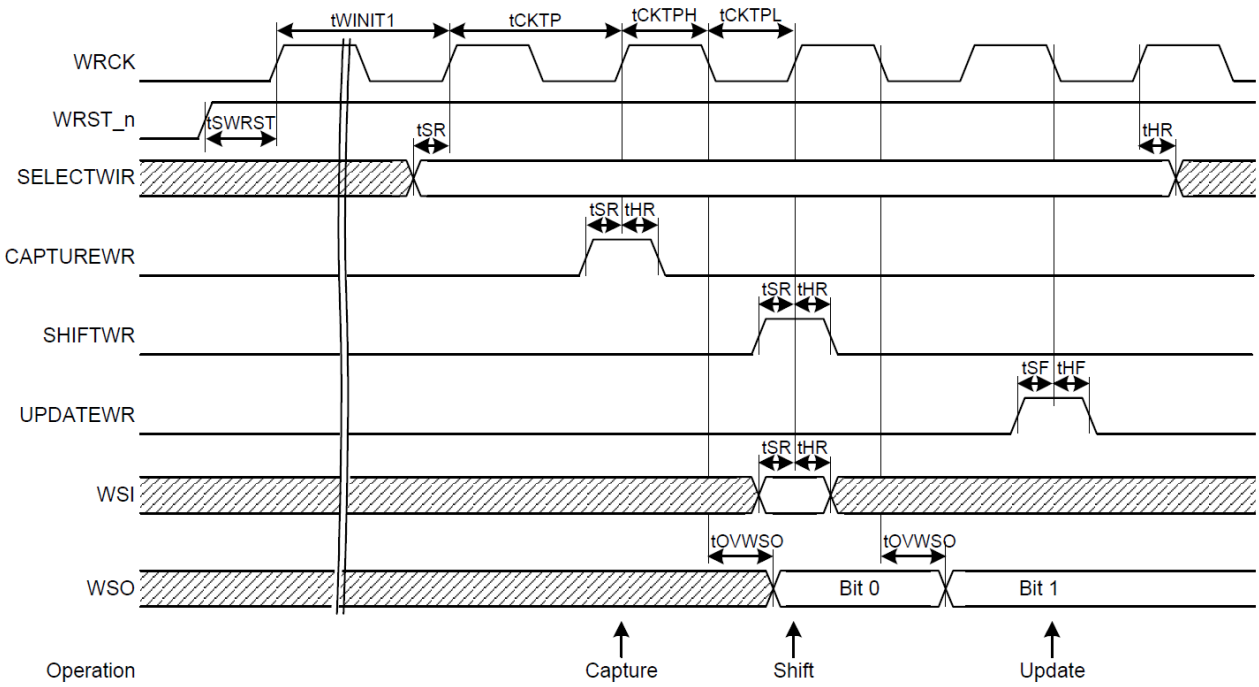


Figure 111 — IEEE1500 Port Input and Output Timings

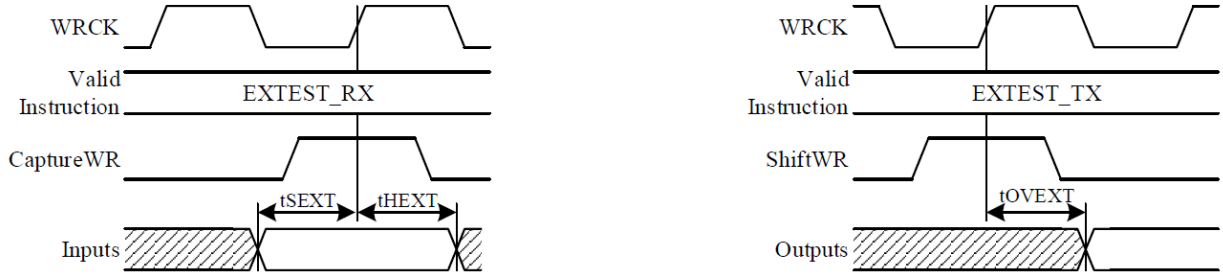


Figure 112 — IEEE1500 EXTEST_RX and EXTEST_TX Instruction Related Timings

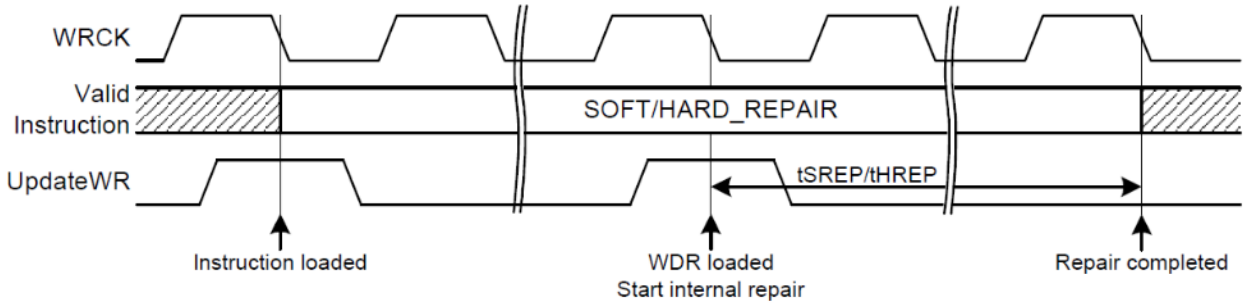


Figure 113 — IEEE1500 SOFT_REPAIR and HARD_REPAIR Instruction Related Timings

13.7 IEEE1500 Test Port AC Timing Parameters (cont'd)

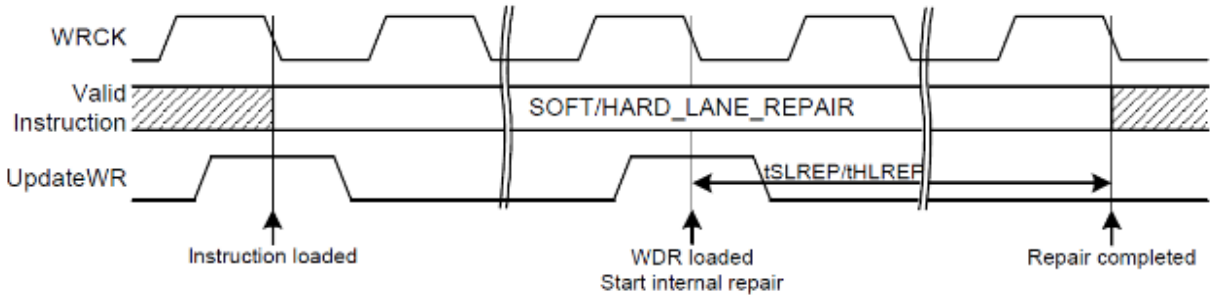
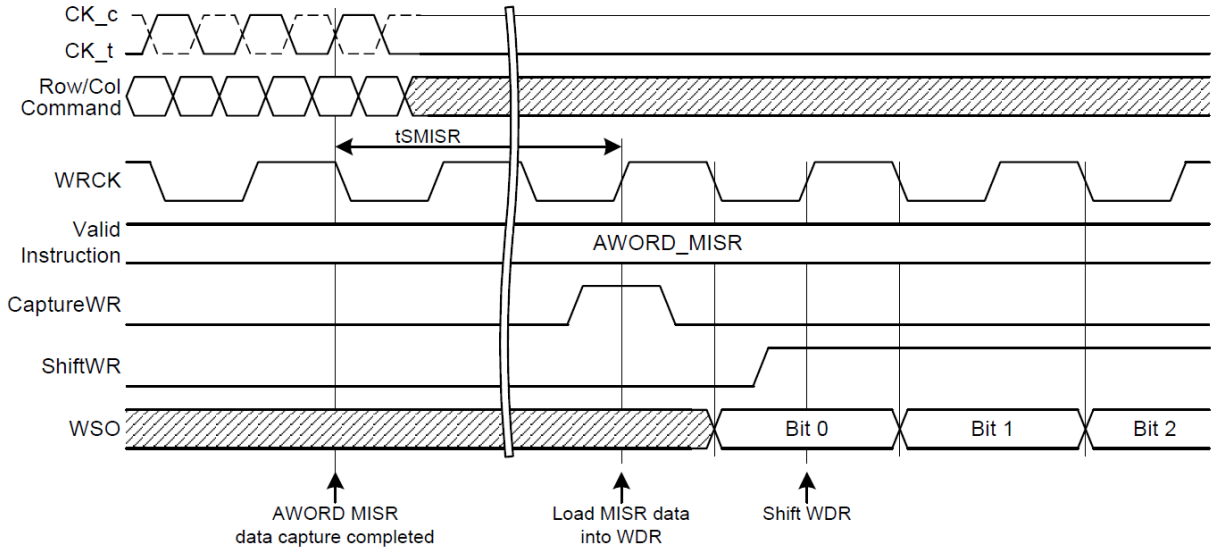
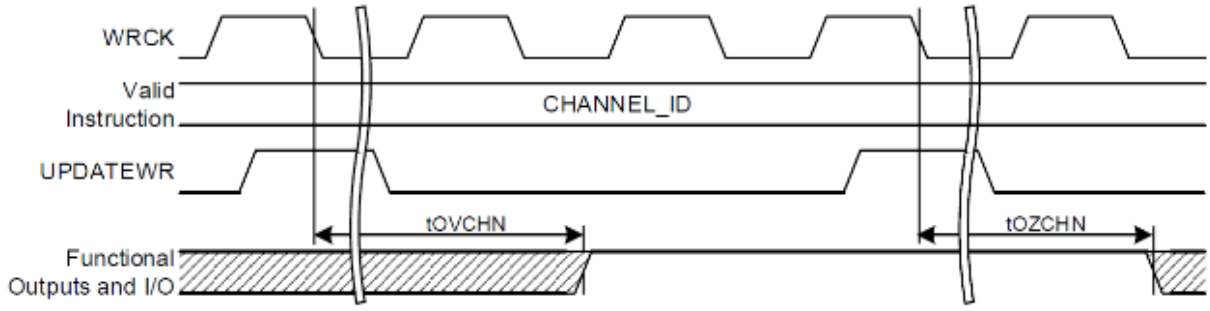


Figure 114 — IEEE1500 Soft_Lane_Repair And Hard_Lane_Repair Instruction Related Timings



NOTE 1 Same timings for data inputs and DWORD MISR with DWORD_MISR instruction.
NOTE 2 tOVWSO = 0 for illustration purpose.

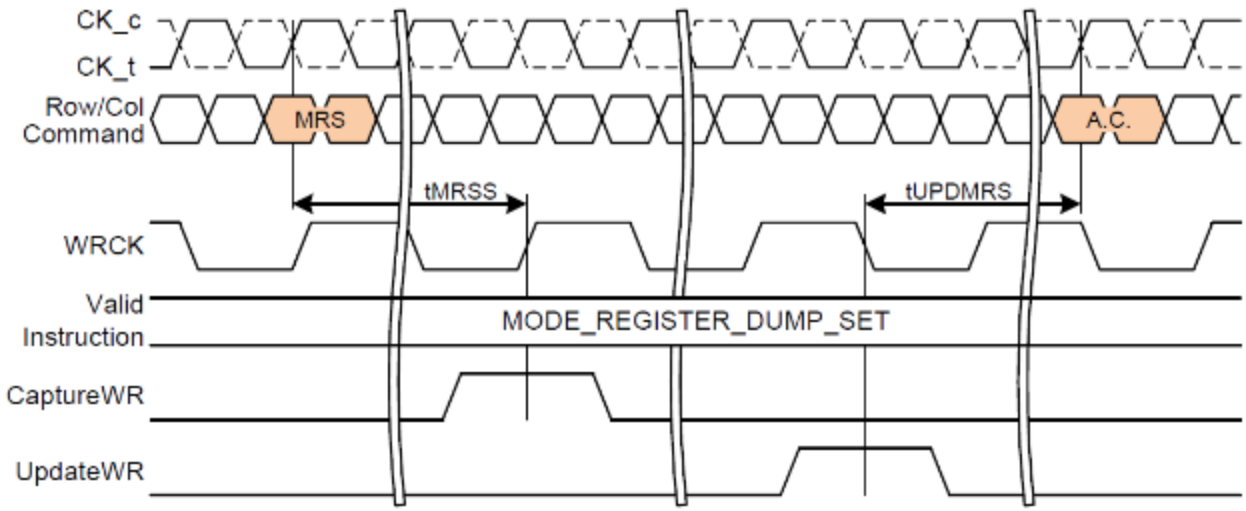
Figure 115 — IEEE1500 DWORD_MISR / AWORD_MISR Instruction Related Timings



NOTE 1 tOVCHN and tOZCHN refer to set/reset of the Enable bit in the WDR.

Figure 116 — IEEE1500 CHANNEL_ID Instruction Related Timings

13.7 IEEE1500 Test Port AC Timing Parameters (cont'd)



A.C. = any command allowed in bank idle state.

Figure 117 — IEEE1500 MODE_REGISTER_DUMP_SET Instruction Related Timings

13.8 Boundary Scan

The HBM4 DRAM supports a boundary scan chain per channel via the IEEE1500 test port. The boundary scan operation is associated with IEEE1500 test port instructions EXTEST_RX and EXTEST_TX.

Scan data is shifted in through WSI and out through the respective WSO, based on the active channel selections in the WIR (see Table 112). All functional pins are included in the boundary scan chains. Table 116 lists the micro-bump boundary scan chain order. Bit position 0 is the first bit shifted in on WSI and out on the WSOs.

The boundary scan chain length for all channels is 122 bits (see Table 116), with dummy WDR bit padding as needed on the MSB end of the chains. Matched length boundary scan chains allow all channel chains to be loaded with matching data with one shift operation when WIR[13:8] =3Fh.

All input-only and output-only pins are implemented as bi-directionals to aid in SIP package level testingand fault isolation. Effectively, all pins support both EXTEST_RX and EXTEST_TX instructions.

I/O signals power up in input mode by default. As soon as EXTEST_TX becomes the current instruction, the I/Os will change to output mode, and the outputs will drive the values shifted into the WDR shift stage.

When EXTEST_RX is the current instruction, all functional pins of the selected channel(s) enter a High-Z state, including the output-only pins AERR, DERR, RDQS_t/RDQS_c, and CATTRIP. On a subsequent CaptureWR event the pins will capture the input values into the WDR shift stage.

Boundary scan mode entry may be asserted at any time after device initialization and before normal memory operation has commenced, e.g. before the CK clock has started to toggle. Upon exiting the scan mode, the state of the HBM4 DRAM is unknown and the integrity of the data content of the memory array is not guaranteed and therefore the reset initialization sequence is required before returning to normal operation.

14 Mechanical Test Vechicle (MTV) Specification

MTV Daisy Chain or netlists is used for providing testability through mimicking electrical or mechanical features of HBM product in prior to setting up manufacturing infrastructures. These netlists must include TSVDC, C4DC, Heater, EDM, NC, No bump, EDM, and Vendor Specific bumps. (Naming convention of all nets follows rule designated in Table 141) Named as Vendor Specific nets may cover EM and associated sensor structures upon supplier's supports. MTV daisy chain must have identical geometrical dimensions and micro bump layouts of actual HBM products, defined in Table 108 and section 11 Package (Die) Specification.

14.1 Scope

14.2 Features

14.3 Requirements

14.3.1 Etch Damage Monitor (EDM)

14.3.2 Heater and Thermal Sensor

14.3.3 C4EM, TSVEM, and Thermal Sensor

14.3.4 TSVDC

14.3.5 C4DC

14.4 Daisy Chain Matrix

This section is intended for providing overview of daisy chain definition. Figure 118 shows top-view of daisy chains and relative locations of each nets. Bump X-Y coordinates, legends of daisy chain matrix, net name, signal names, and signal lists are shown in the part of separate spreadsheet (HBM4 TMTV Bump Map Excel Intel 01_29_2024). Functionalities and usage of Vendor Specific nets should be consulted by vendor's application notes or datasheets.

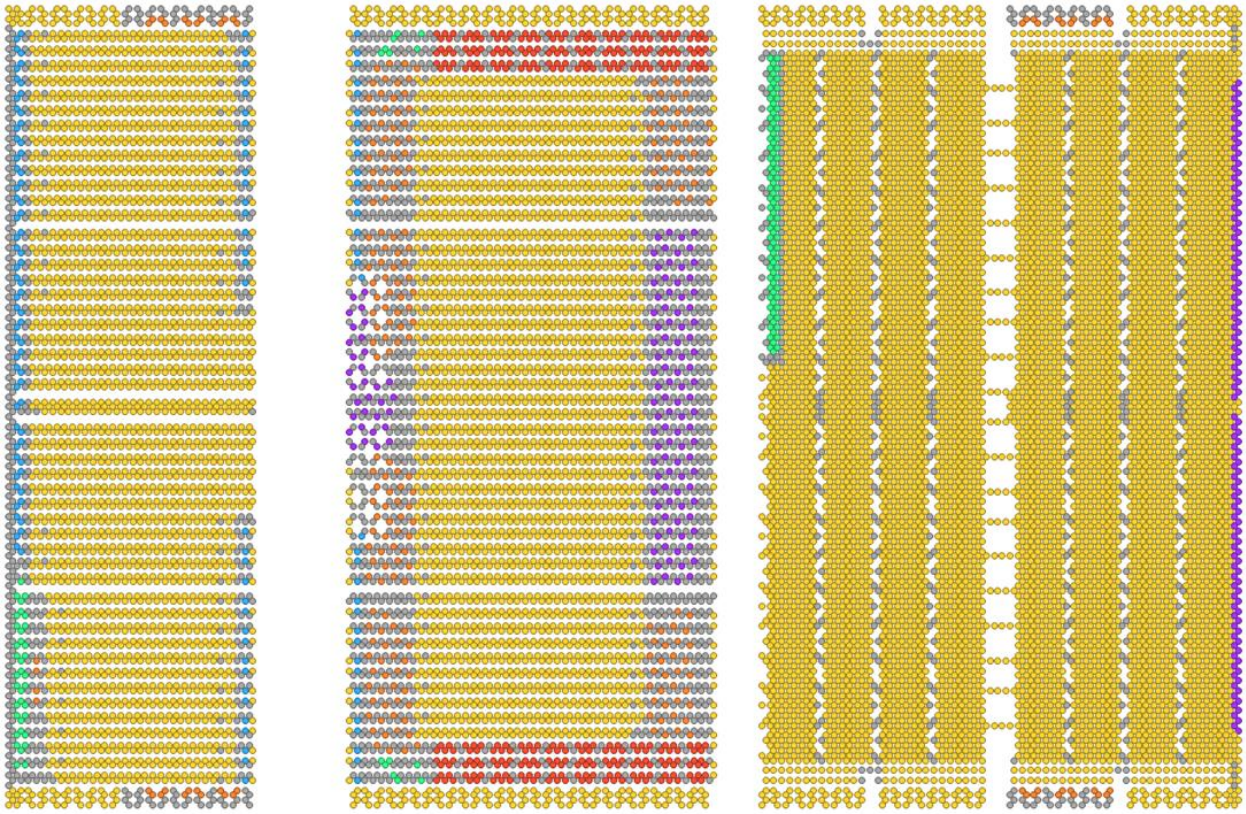


Figure 118 — Top-view of HBM4 Mechanical Test Vehicle (MTV4) Netlists

Table 141 — MTV4 Net names

Label Color	Legend
	TS_BASE / TS_TOP / TS_STACK
	HTR_BASE / HTR_STACK / HTR_TOP
	No Connects (NC)
	Vendor Specific / IMAX (C4EM)
	Edge Damage Monior (EDM)
	TSVDC
	C4DC
Note 1	This table only shows listed net names by functions and their relative locations. Refer to Table XX and section XX for detailed rules and connection rules

Standard Improvement Form **JEDEC Standard JESD238**

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

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

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

Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

_____

_____

_____

_____

3. Other suggestions for document improvement:

_____

_____

_____

_____

Submitted by

Name: _____ Phone: _____

Company: _____ E-mail: _____

Address: _____

City/State/Zip: _____ Date: _____


JEDEC®