TENTATIVE TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT SILICON MONOLITHIC

Lead Free

OVERVIEW

The Rambus XDR™ DRAM device is a general purpose high-performance memory device suitable for use in a broad range of applications including computer memory, graphics, video, and any other application where high bandwidth and low latency are required.

The 256Mb Rambus XDR DRAM device is a CMOS DRAM organized as 16M words by 16 bits. The use of Differential Rambus Signaling Level (DRSL) technology permits 4000/3200/2400 Mb/s transfer rates while using conventional system and board design technologies. XDR DRAM devices are capable of sustained data transfers of 8000/6400/4800 MB/s.

XDR DRAM device architecture allows the highest sustained bandwidth for multiple, interleaved randomly addressed memory transactions. The highly efficient protocol yields over 95% utilization while allowing fine access granularity. The device's 8 banks support up to four interleaved transactions.

FEATURES

- Highest pin bandwidth available
  - 4000/3200/2400 Mb/s Octal Data Rate (ODR) Signaling
  - Bi-directional differential RSL (DRSL)
    Flexible read/write bandwidth allocation
    Minimum pin count
  - Programmable on-chip termination
    Adaptive impedance matching
    Reduced system cost and routing complexity

- Highest sustained bandwidth per DRAM device
  - 8000/6400/4800 MB/s sustained data rate
  - 8 banks: bank-interleaved transactions at full bandwidth
  - Dynamic request scheduling
  - Early-Read-after-Write support for maximum efficiency
  - Zero overhead refresh

- Low latency
  - 2.0/2.5/3.33 ns request packets
  - Point-to-point data interconnect for fastest possible flight time
  - Support for low-latency, fast-cycle cores

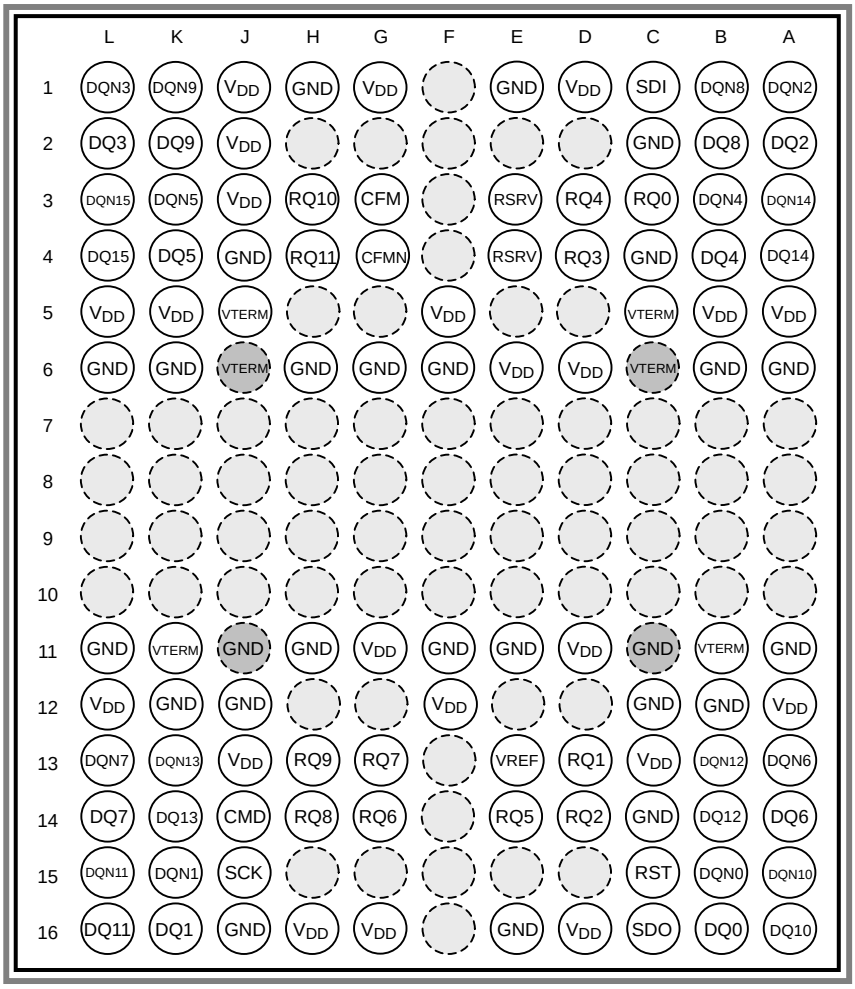
- Low power
  - 1.8V V_{DD}
  - Programmable small-swing I/O signaling (DRSL)
  - Low power PLL/DLL design
  - Power Down Self Refresh support
  - Per pin I/O Power Down for narrow-width operation

- Programmable I/O width
  - ×4 / ×8 / ×16 programmable device I/O width

Note: XDR is a trademark or a registered trademark in Japan and/or other countries.

PIN ASSIGNMENT (TOP VIEW)

XDR DRAM CSP x16



Note:
- ◌: Optional ball / Depopulated
- ◌: Depopulated
- RSRV: Reserved pin
- DQ8…DQ15, DQN8…DQN15 are RSRV's for ×8
- DQ4…DQ15, DQN4…DQN15 are RSRV's for ×4

Key Timing Parameters/Part Numbers

Organization [a]	Bandwidth ($1/t_{BIT}$) [b]	Latency (t_{RAC}) [c]	Bin [d]	Part Number
8 × 2K × 1K × 16	2400	36	A	TC59YM816BKG24A
8 × 2K × 1K × 16	3200	27	A	TC59YM816BKG32A
8 × 2K × 1K × 16	3200	35	B	TC59YM816BKG32B
8 × 2K × 1K × 16	3200	35	C	TC59YM816BKG32C
8 × 2K × 1K × 16	4000	28	B	TC59YM816BKG40B
8 × 2K × 1K × 16	4000	28	C	TC59YM816BKG40C

a. Bank × Row × Column × Width

b. Data rate measured in Mbit/s per DQ differential pair. See "Timing Conditions" on page 60 and "Timing Characteristics" on page 62. Note that $t_{BIT} = t_{CYCLE} / 8$.

c. Read access time t_{RAC} ($=t_{RCD-R} + t_{CAC}$) measured in ns. See "Timing Parameters" on page 63.

d. Timing parameter bin. See "Timing Parameters" on page 63. This is a measure of the number of interleaved read transactions needed for maximum efficiency (the value Ceiling (t_{RC-R}/t_{RR-D}).
For bin A, $t_{RC-A} / t_{RR-D} = 4$, and for bin B, $t_{RC-R} / t_{RR-D} = 5$

General Description

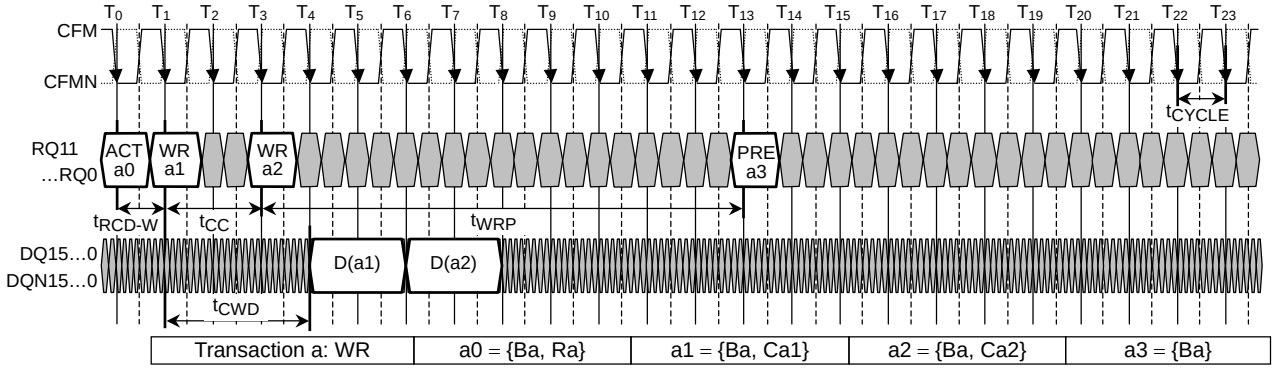
The timing diagrams in Figure 1 illustrate XDR DRAM device write and read transactions. There are three sets of pins used for normal memory access transactions: CFM/CFMN clock pins, RQ11…RQ0 request pins, and DQ15…DQ0/DQN15...DQN0 data pins. The "N" appended to a signal name denotes the complementary signal of a differential pair.

A transaction is a collection of packets needed to complete a memory access. A packet is a set of bit windows on the signals of a bus. There are two buses that carry packets: the RQ bus and DQ bus. Each packet on the RQ bus uses a set of 2 bit-windows on each signal, while the DQ bus uses a set of 16 bit-windows on each signal.

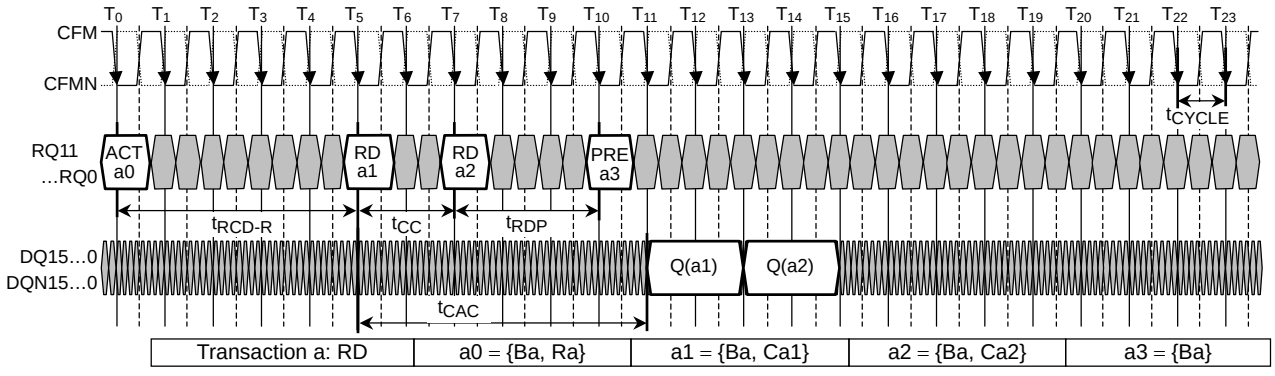
In the write transaction shown in Figure 1, a request packet (on the RQ bus) at clock edge T_0 contains an activate (ACT) command. This causes row Ra of bank Ba in the memory component to be loaded into the sense amp array for the bank. A second request packet at clock edge T_1 contains a write (WR) command. This causes the data packet D (a1) at edge T_4 to be written to column Ca1 of the sense amp array for bank Ba. A third request packet at clock edge T_3 contains another write (WR) command. This causes the data packet D (a2) at edge T_6 to be also be written to column Ca2. A final request packet at clock edge T_{13} contains a precharge (PRE) command.

The spacing between the request packets are constrained by the following timing parameters in the diagram: t_{RCD-W}, t_{CC}, and t_{WRP}. In addition, the spacing between the request packets and data packets are constrained by the t_{CWRD} parameter. The spacing of the CFM/CFMN clock edges is constrained by t_{CYCLE}.

Figure 1. XDR DRAM Device Write and Read Transactions



Write Transaction



Read Transaction

The read transaction shows a request packet at clock edge T_0 containing an ACT command. This causes row Ra of bank Ba of the memory component to load into the sense amp array for the bank. A second request packet at clock edge T_5 contains a read (RD) command. This causes the data packet Q (a1) at edge T_{11} to be read from column Ca1 of the sense amp array for bank Ba. A third request packet at clock edge T_7 contains another RD command. This causes the data packet Q (a2) at edge T_{13} to also be read from column Ca2. A final request packet at clock edge T_{10} contains a PRE command.

The spacing between the request packets are constrained by the following timing parameters in the diagram: t_{RCD-R}, t_{CC}, and t_{RDP}. In addition, the spacing between the request and data packets is constrained by the t_{CAC} parameter.

Table of Contents

Table of Tables

Table of Figures

Pin Description

Table 1 summarizes the pin functionality of the XDR DRAM device. The first group of pins provide the necessary supply voltages. These include VDD and GND for the core and interface logic, VREF for receiving input signals, and VTERM for driving output signals.

The next group of pins is used for high bandwidth memory accesses. These include DQ15…DQ0 and DQN15...DQN0 for carrying read and write data signals, RQ11...RQ0 for carrying request signals, and CFM and CFMN for carrying timing information used by the DQ, DQN, and RQ signals.

The final set of pins comprises the serial interface that is used for control register accesses. These include RST for initializing the state of the device, CMD for carrying command signals, SDI and SDO for carrying register read data, and SCK for carrying the timing information used by the RST, SDI, SDO, and CMD signals.

Table 1. Pin Descriptions

Signal	I/O	Type	No. of Pins	Description
V_{DD}	—	—	22	Supply voltage for the core and interface of the device.
GND	—	—	26 [a]	Ground reference for the core and interface logic of the device.
V_{REF}	—	—	1	Logic threshold reference voltage for RSL signals.
V_{TERM}	—	—	6 [a]	Termination voltage for DRSL signals.
DQ15…DQ0	I/O	DRSL [b]	16	Positive data signals that carry write or read data to and from the device.
DQN15…DQN0	I/O	DRSL [b]	16	Negative data signals that carry write or read data to and from the device.
RQ11…RQ0	I	RSL [b]	12	Request signals that carry control and address information to the device.
CFM	I	DIFFCLK [b]	1	Clock from master – Positive interface clock used for receiving RSL signals, and receiving and transmitting DRSL signals from the Channel.
CFMN	I	DIFFCLK [b]	1	Clock from master – Negative interface clock used for receiving RSL signals, and receiving and transmitting DRSL signals from the Channel.
RST	I	RSL [b]	1	Reset input – This pin is used to initialize the device.
CMD	I	RSL [b]	1	Command input – This pin carries command, address, and control register write data into the device.
SCK	I	RSL [b]	1	Serial clock input – Clock source used for reading from and writing to the control registers.
SDI	I	RSL [b]	1	Serial data input – This pin carries control register read data through the device. This pin is also used to initialize the device.
SDO	O	CMOS [b]	1	Serial data output – This pin carries control register read data from the device. This pin also used to initialize the device.
RSRV	—	—	2	Reserved pins – Follow Rambus XDR system design guidelines for connecting RSRV pins.
Total pin count per package			108	

a. J6 / J11 / C6 / C11 are optional balls. This table represent a superset across all the generations and densities pf XDR DRAM.

b. All DQ and CFM signals are high-true; low voltage is logic 0 and high voltage is logic 1.
All DQN, CFMN, RQ, RSL and CMOS signals are low-true; high voltage is logic 0 and low voltage is logic 1.

Block Diagram

A block diagram of the XDR DRAM device is shown in Figure 2. It shows all interface pins and major internal blocks.

The CFM and CFMN clock signals are received and used by the clock generation logic to produce three virtual clock signals: $1/t_{CYCLE}$, $2/t_{CYCLE}$, and $16/t_{CC}$. The frequency of these signals are 1x, 2x, and 8x that of the CFM and CFMN signals. These virtual signals show the effective data rate of the logic blocks to which they connect; they are not necessarily present in the actual memory component.

The RQ11...RQ0 pins receive the request packet. Two 12-bit words are received in one t_{CYCLE} interval. This is indicated by the $2/t_{CYCLE}$ clocking signal connected to the 1: 2 Demux Block that assembles the 24-bit request packet. These 24 bits are loaded into a register (clocked by the $1/t_{CYCLE}$ clocking signal) and decoded by the Decode Block. The VREF pin supplies a reference voltage used by the RQ receivers.

Three sets of control signals are produced by the Decode Block. These include the bank (BA) and row (R) addresses for an activate (ACT) command, the bank (BR) and row (REFr) addresses for a precharge (PRE) command, the bank (BP) address for a precharge (PRE) command, the bank (BR) address for a refresh precharge (REFP) command, and the bank (BC) and column (C and SC) addresses for a read (RD) or write (WR or WRM) command. In addition, a mask (M) is used for a masked write (WRM) command.

These commands can all be optionally delayed in increments of t_{CYCLE} under control of delay fields in the request. The control signals of the commands are loaded into registers and presented to the memory core. These registers are clocked at maximum rates determined by core timing parameters, in this case $1/t_{RR}$, $1/t_{PP}$, and $1/t_{CC}$ (1/4, 1/4, and 1/2 the frequency of CFM). These registers may be loaded at any t_{CYCLE} rising edge. Once loaded, they should not be changed until a t_{RR}, t_{PP}, or t_{CC} time later because timing paths of the memory core need time to settle.

A bank address is decoded for an ACT command. The indicated row of the selected bank is sensed and placed into the associated sense amp array for the bank. Sensing a row is also referred to as “opening a page” for the bank.

Another bank address is decoded for a PRE command. The indicated bank and associated sense amp array are precharged to a state in which a subsequent ACT command can be applied. Precharging a bank is also called “closing the page” for the bank.

After a bank is given an ACT command and before it is given a PRE command, it may receive read (RD) and write (WR) column commands. These commands permit the data in the bank’s associated sense amp array to be accessed.

For a WR command, the bank address is decoded. The indicated column of the associated sense amp array of the selected bank is written with the data received from the DQ15…DQ0 pins.

The bank address is decoded for a RD command. The indicated column of the selected bank’s associated sense amp array is read. The data is transmitted onto the DQ15...DQ0 pins.

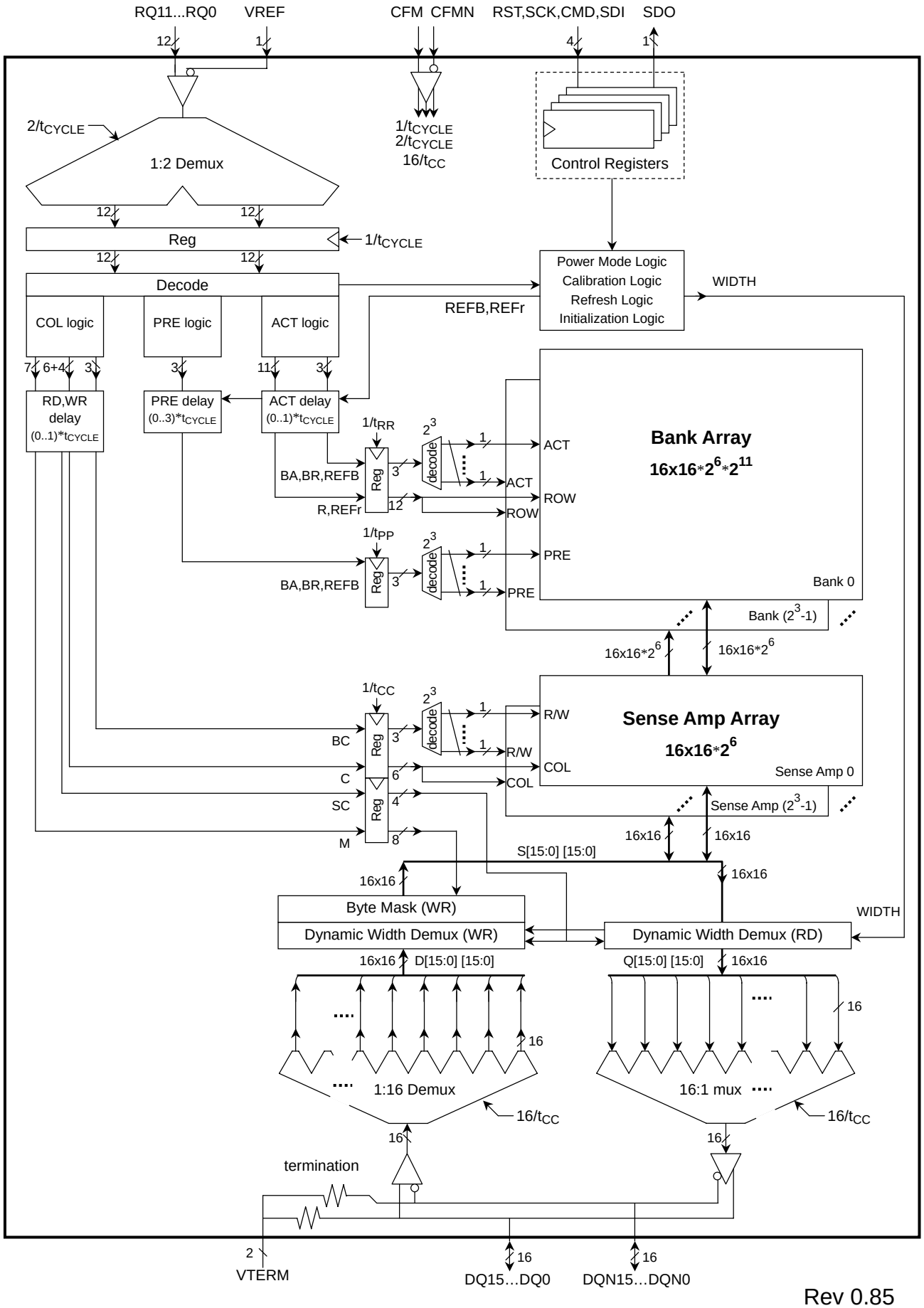
The DQ15...DQ0 pins receive the write data packet (D) for a write transaction. 16 sixteen-bit words are received in one t_{CC} interval. This is indicated by the $16/t_{CC}$ clocking signal connected to the 1:16 Demux Block that assembles the 16x16-bit write data packet. The write data is then driven to the selected Sense Amp Array Bank.

16 sixteen-bit words are accessed in the selected Sense Amp Array Bank for a read transaction. The DQ15…0 pins transmit this read data packet (Q) in one t_{CC} interval. This is indicated by the $16/t_{CC}$ clocking signal connected to the 16:1 Mux Block. The VTERM pin supplies a termination voltage for the DQ pins.

The RST, SCK, and CMD pins connect to the Control Register block. These pins supply the data, address and control needed to write the control registers. The read data for these registers is accessed through the SDO/SDI pins. These pins are also used to initialize the device.

The controls registers are used to transition between power modes, and are also used for calibrating the high speed transmit and receive circuits of the device. The control registers also supply bank (REFB) and row (REFr) addresses for refresh operations.

Figure 2. 512Mb (8x4Mx16) XDR DRAM Block Diagram



Rev 0.85

Request Packets

A request packet carries address and control information to the memory device. This section contains tables and diagrams for packet formats, field encoding and packet interactions.

Request Packet Formats

There are five types of request packets:

1. ROWA - specifies an ACT command
2. COL - specifies RD and WR commands
3. COLM - specifies a WRM command
4. ROWP - specifies PRE and REF commands
5. COLX - specifies the remaining commands

Table 2 describes fields within different request packet types. Various request packet type formats are illustrated in Figure 3.

Each packet type consists of 24 bits sampled on the RQ11..RQ0 pins on two successive edges of the CFM/CFMN clock. The request packet formats are distinguished by the OP3..OP0 field. This field also specifies the operation code of the desired command.

In the ROWA packet, a bank address (BA), row address (R), and command delay (DELA) are specified for the activate (ACT) command.

In the COL packet, a bank address (BC), column address (C), sub-column address (SC), command delay (DELC), and sub-opcode (WRX) are specified for the read (RD) and write (WR) commands.

In the COLM packet, a bank address (BC), column address (C), sub-column address (SC), command delay (DELC), and mask field (M) are specified for the masked write (WRM) command.

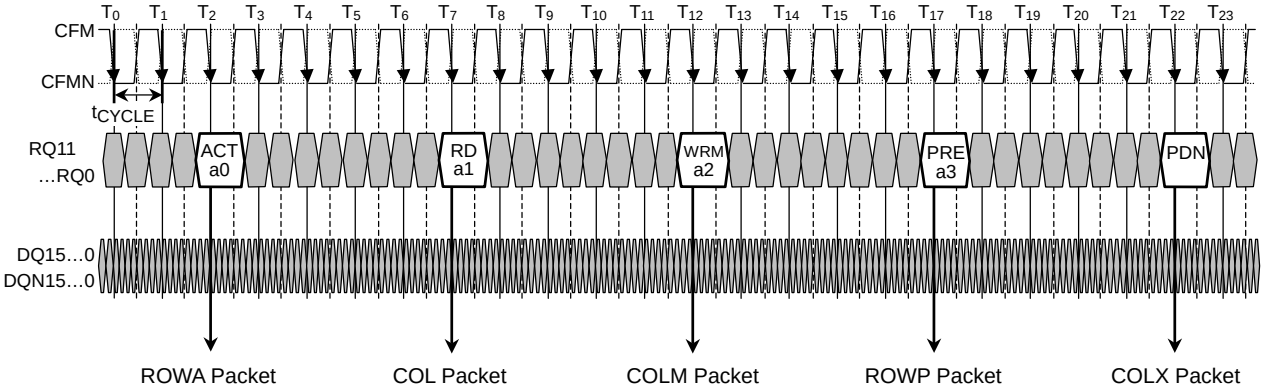
In the ROWP packet, two independent commands may be specified. A bank address (BP) and sub-opcode (POP) are specified for the precharge (PRE) commands. An address field (RA) and sub-opcode (ROP) are specified for the refresh (REF) commands.

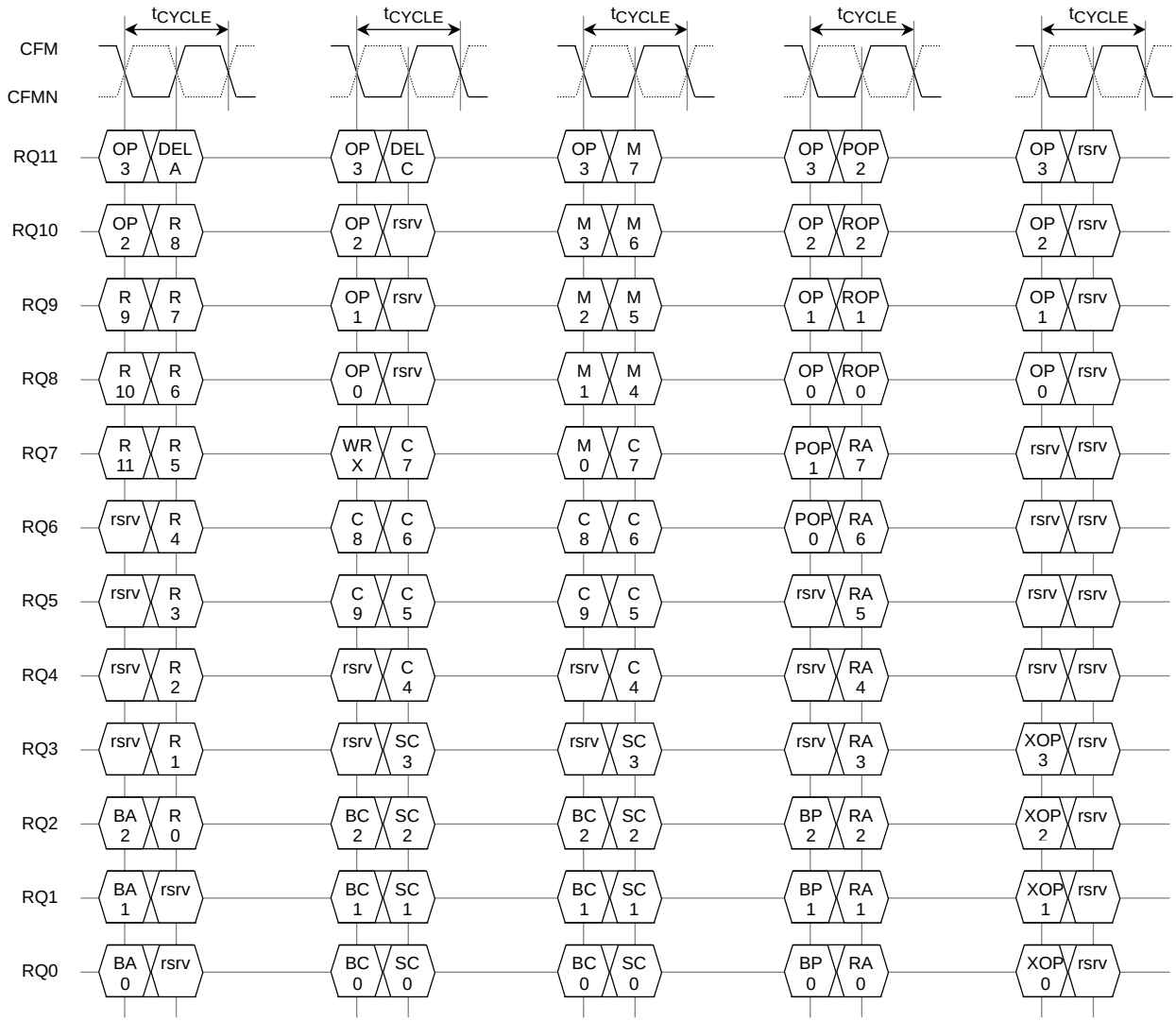
In the COLX packet, a sub-operation code field (XOP) is specified for the remaining commands.

Table 2. Request Field Description

Field	Packet Types	Description
OP3..OP0	ROWA/ROWP/COL/COLM/COLX	4-bit operation code that specifies packet format. (Encoded commands are in a Table 3 on page 12.)
DELA	ROWA	Delay the associated row activate command by 0 or 1 t_{CYCLE}.
BA2..BA0	ROWA	3-bit bank address for row activate command.
R10..R0	ROWA	11-bit row address for row activate command.
WRX	COL	Specifies RD (=0) or WR (=1) command.
DELC	COL	Delay the column read or write command by 0 or 1 t_{CYCLE}.
BC2..BC0	COL/COLM	3-bit bank address for column read or write command.
C9..C4	COL/COLM	6-bit column address for column read or write command.
SC3..SC0	COL/COLM	4-bit sub-column address for dynamic width (see “Dynamic Width Control” on page 49).
M7..M0	COLM	8-bit mask for masked-write command WRM.
POP2..POP0	ROWP	3-bit operation code that specifies row precharge command with a delay of 0 to 3 t_{CYCLE}. (Encoded commands are in Table 5 on page 13).
BP2..BP0	ROWP	3-bit bank address for row precharge command.
ROP2..ROP0	ROWP	3-bit operation code that specifies refresh commands. (Encoded commands are in Table 4 on page 12).
RA7..RA0	ROWP	8-bit refresh address field (specifies BR bank address, delay value, and REFr load value)
XOP3..XOP0	COLX	4-bit extended operation code that specifies column preload, calibration and Power Down commands. (Encoded commands are in Table 6 on page13).

Figure 3. Request Packet Formats





Request Field Encoding

Operation-code fields are encoded within different packet types to specify commands. Table 3 through Table 6 provides packet type and encoding summaries.

Table 3 shows the OP field encoding for the five packet types. The COLM and ROWA packets each specify a single command: ACT and WRM. The COL, COLX, and ROWP packets each use additional fields to specify multiple commands: WRX, XOP, and POP/ROP, respectively. The COLM packet specifies the masked write command WRM. This is like the WR unmasked write command, except that a mask field M7...M0 indicates whether each byte of the write data packet is written or not written. The ROWA packet specifies the row activate command ACT. The COL packet uses the WRX field to specify the column read and column write (unmasked) commands

Table 3. OP Field Encoding Summary

OP [3:0]	Packet	Command	Description
0000	—	NOP	No operation
0001	COL	RD	Column read (WRX = 0). Column C9…C4 of sense amp in bank BC2…BC0 is read to DQ bus after DELC*t_{CYCLE}.
		WR	Column write (WRX = 1). Write DQ bus to column C9…C4 of sense amp in bank BC2…BC0 after DELC*t_{CYCLE}.
0010	COLX	CALy	XOP3…XOP0 specifies a calibrate or Power Down command – see Table 6 on page 13.
0011	ROWP	PREx	POP2…POP0 specifies a row precharge command – see Table 5 on page 13.
		REFy, LRRr	ROP2…ROP0 specifies a row refresh command or load REFr register command – see Table 4 on page 12.
01xx	ROWA	ACT	Row activate command. Row R11…R0 of bank BA2…BA0 is placed into the sense amp of the bank after DELA*t_{CYCLE}.
1xxx	COLM	WRM	Column write command (masked) – mask M7…M0 specifies which bytes are written.

Encoding of the ROP field in the ROWP packet is shown in Table 4. The first encoding specifies a NOPR (no operation) command. The REFP command uses the RA field to select a bank to be precharged. The REFA and REFI commands use the RA field and REFH/M/L registers to select a bank and row to be activated for refresh. The REFI command also increments the REFH/M/L register. The REFP, REFA, and REFI commands may also be delayed by up to 3*t_{CYCLE} using the RA [7:6] field. The LRR0, LRR1, and LRR2 commands load the REFH/M/L registers from the RA [7:0] field.

Table 4. ROP Field Encoding Summary

ROP [2:0]	Command	Description
000	NOPR	No operation
001	REFP	Refresh precharge command. Bank RA2…RA0 is precharged. This command is delayed by {0, 1, 2, 3}*t_{CYCLE} (the value is given by the expression (2*RA [7] + RA [6]).
010	REFA	Refresh activate command. Row R [11:0] (from REFH/M/L register) of bank RA2…RA0 is placed into sense amp. This command is delayed by {0, 1, 2, 3}*t_{CYCLE} (the value is given by the expression (2*RA [7] + RA [6]).
011	REFI	Refresh activate command. Row R [11:0] (from REFH/M/L register) of bank RA2…RA0 is placed into sense amp. This command is delayed by {0, 1, 2, 3}*t_{CYCLE} (the value is given by the expression (2*RA [7] + RA [6]). R[11 : 0] field of REFH/M/L register is incremented after the activate command has completed.
100	LRR0	Load Refresh Low Row register (REFL). RA [7:0] is stored in R [7:0] field.
101	LRR1	Load Refresh Middle Row register (REFM). RA [3:0] is stored in R [11:8] field.
110	LRR2	Load Refresh High Row register – not used with this device
111	—	Reserved

The REFH/M/L registers are also referred to as the REFr registers. Note that only the bits that are needed for specifying the refresh row (12 bits in all) are implemented in the REFr registers - the rest are reserved. Note also that the RA2…RA0 field that specifies the refresh bank address is also referred to as BR2…BR0. See “Refresh Transactions” on page 40.

Table 5 shows the POP field encoding in the ROWP packet. The first encoding specifies a NOPP (no operation) command. There are four variations of PRE (precharge) command. Each uses the BP field to specify the bank to be precharged. Each also specifies a different delay of up to 3*t_{CYCLE} using the POP [1:0] field. A precharge command may be specified in addition to a refresh command using the ROP field.

Table 5. POP Field Encoding Summary

POP [2:0]	Command	Description
000	NOPP	No operation
001	—	Reserved.
010	—	Reserved.
011	—	Reserved.
100	PRE0	Row precharge command – Bank BP2… BP0 is precharged. This command is delayed by 0*t_{CYCLE}.
101	PRE1	Row precharge command – Bank BP2… BP0 is precharged. This command is delayed by 1*t_{CYCLE}.
110	PRE2	Row precharge command – Bank BP2… BP0 is precharged. This command is delayed by 2*t_{CYCLE}.
111	PRE3	Row precharge command – Bank BP2… BP0 is precharged. This command is delayed by 3*t_{CYCLE}.

Table 6 shows the XOP field encoding in the COLX packet. This field encodes the remaining commands.

The CALC and CALE commands perform calibration operations to ensure signal integrity on the Channel. See “Calibration Transactions” on page 42.

The PDN command causes the device to enter a power-down state. See “Power State Management” on page 43.

Table 6. XOP Field Encoding Summary

XOP [3:0]	Command	Command and Description	XOP [3:0]	Command	Command and Description
0000	—	Reserved	1000	CALC	Current calibration command.
0001	—	Reserved.	1001	CALZ	Impedance calibration command.
0010	—	Reserved.	1010	CALE	End calibration command (CALC).
0011	—	Reserved.	1011	—	Reserved.
0100	—	Reserved.	1100	PDN	Enter Power Down power state.
0101	—	Reserved.	1101	—	Reserved
0110	—	Reserved	1110	—	Reserved
0111	—	Reserved	1111	—	Reserved.

Request Field Interactions

A summary of request packet interactions is shown in Table 7. Each case is limited to request packets with commands that perform memory operations (including refresh commands). This includes all commands in ROWA, ROWP, COL, and COLM packets. The commands in COLX packets are described in later sections. See "Maintenance Operations" on page 40.

Request packet/command "a" is followed by request packet/command "b". The minimum possible spacing between these two packet/commands is $0*t_{CYCLE}$. However, a larger time interval may be needed because of a resource interaction between the two packet/commands. If the minimum possible spacing is $0*t_{CYCLE}$, then an entry of "No limit" is shown in the table.

Note that the spacing values shown in the table are relative to the ***effective*** beginning of a packet/command. The use of the delay field with a command will delay the position of the effective packet/command from the position of the actual packet/command. See "Dynamic Request Scheduling" on page 20.

Any of the packet/command encoding under one of the four operation types is equivalent in terms of the resource constraints. Therefore, both the horizontal columns (packet "a") and vertical rows (packet "b") of the interaction table are divided into four major groups.

The four possible operation types for request packets a and b include:

- **: [A] Activate Row**
  - **ROWA/ACT**
  - **ROWP/REFA**
  - **ROWP/REFI**
- **; [R] Read Column**
  - **COL/RD**
- **; [W] Write Column**
  - **COL/WR**
  - **COLM/WRM**
- **; [P] Precharge Row**
  - **ROWP/PRE**
  - **ROWP/REFP**

Table 7. Packet Interaction Summary

First packet command to bank Ba		Second packet/command to bank Bb			
		Activate Row [A] ROWA – ACT Bb ROWP – REFA Bb ROWP – REFI Ba	*Read Column [R]* COL – RD Bb	*Write Column [W]* COL – WR Bb COLM – WRM Bb	*Precharge Row[P]* ROWP – PRE Bb ROWP – REFP Bb
Activate Row [A] ROWA – ACT Ba ROWP – REFA Ba ROWP – REFI Ba	Ba,Bb different	Case AAd: t_{RR}	Case ARd: No limit	Case AWd: No limit	Case APd: No limit
	Ba,Bb same	Case AAs: t_{RC}	Case ARs: t_{RCD-R}	Case AWs: t_{RCD-W}	Case APs: t_{RAS}
Read Column [R] COL – RD Ba	Ba,Bb different	Case RAd: No limit	Case RRd: t_{CC}	Case RWd[a]: $t_{\Delta RW}$	Case RPd: No limit
	Ba,Bb same	Case RAs[b]: $t_{RDP}+t_{RP}$	Case RRs: t_{CC}	Case RWs[a]: $t_{\Delta RW}$	Case RRs: t_{RDP}
Write Column [W] COL – WR Ba COLM – WRM Ba	Ba,Bb different	Case WAd: No limit	Case WRd[c]: $t_{\Delta WR}$	Case WWd: t_{CC}	Case WPd: No limit
	Ba,Bb same	Case WAs[b]: $t_{WRP}+t_{RP}$	Case WRs[c]: $t_{\Delta WR}$	Case WWs: t_{CC}	Case WPs: t_{WRP}
Precharge Row [P] ROWP – PRE Ba ROWP – REFP Ba	Ba,Bb different	Case PAd: No limit	Case PRd: No limit	Case PWd: No limit	Case PPd: t_{PP}
	Ba,Bb same	Case PAs: t_{RP}	Case PRs[d]: $t_{RP}+t_{RCD-R}$	Case PWs[d]: $t_{RP}+t_{RCD-W}$	Case PPs: t_{RC}
See Examples:		Figure 4	Figure 5	Figure 6	Figure 7

a. $t_{\Delta RW}$ is equal to $t_{CC} + t_{RW-BUB,XDR\ DRAM} + t_{CAC} - t_{CWD}$ and is defined in Table 17. This also depends upon propagation delay – See "Propagation Delay" on page 30.

b. A PRE command is needed between the RD and ACT/REFA commands or the WR/WRM and ACT/REFA commands.

c. $t_{\Delta WR}$ is defined in Table 17.

d. An ACT command is needed between the PRE/REFP and RD commands or the PRE/REFP and WR/WRM commands.

The first request is shown along the vertical axis on the left of the table. The second request is shown along the horizontal axis at the top of the table. Each request includes a bank specification “Ba” and “Bb”. The first and second banks may be the same, or they may be different. These two sub cases for each interaction are shown along the vertical axis on the left.

There are 32 possible interaction cases altogether. The table gives each case a label of the form “xyz”, where “x” and “y” are one of the four operation types (“A” for Activate, “R” for Read, “W” for Write, or “P” for Precharge) for the first and second request, respectively, and “z” indicates the same bank (“s”) or different bank (“d”).

Along the horizontal axis at the bottom of the table are cross-references to four figures (Figure 4 through Figure 7). Each figure illustrates the eight cases in the corresponding vertical column. Thus, Figure 4 shows the eight cases when the second request is an activate operation (“A”). In the following discussion of the cases, only those in which the interaction interval is greater than t_{CYCLE} will be described.

Request Interactions Cases

In Figure 4, the interaction interval for the AAd case is t_{RR}. This parameter is the row-to-row time and is the minimum interval between activate commands to different banks of a device.

The interaction interval for the AAs case is t_{RC}. This is the row cycle time parameter and is the minimum interval between activate commands to same banks of a device. A precharge operation must be inserted between the two activate operations.

The interaction interval for the RAs case is $t_{WRP} + t_{RP}$. A precharge operation must be inserted between the read and activate operation. The minimum interval between a read and a precharge operation to a bank is t_{RDP}. The minimum interval between a precharge and an activate operation to a bank is t_{RP}.

The interaction interval for the WAs case is $t_{WRP} + t_{RP}$. A precharge operation must be inserted between the read and the activate operation. The minimum interval between a write and a precharge operation to a bank is t_{WRP}. The minimum interval between a precharge and an activate operation to a bank is t_{RP}.

The interaction interval for the PAs case is t_{RP}. The minimum interval between a precharge and an activate operation to a bank is t_{RP}.

In Figure 5, the interaction interval for the ARs case is t_{RCD-R}. This is the row-to-column-read time parameter and represents the minimum interval between an activate operation and a read operation to a bank.

The interaction interval for the RRd and RRs cases is t_{CC}. This is the column-to-column time parameter and represents the minimum interval between two read operations.

The interaction interval for the WRd and WRs cases is $t\Delta_{WR}$. This is the write-to-read time parameter and represents the minimum interval between a write and a read operation to any banks. See “Read/Write Interaction” on page 28.

The interaction interval for the PRs case is $t_{RP} + t_{RCD-R}$. An activate operation must be inserted between the precharge and the read operation. The minimum interval between a precharge and an activate operation to a bank is t_{RP}. The minimum interval between an activate and read operation to a bank is t_{RCD-R}.

In Figure 6, the interaction interval for the AWs case is t_{RCD-W}. This is the row-to-column-write timing parameter and represents the minimum interval between an activate operation and a write operation to a bank.

The interaction interval for the RWd and RWs cases is $t\Delta_{RW}$. This is the read-to-write time parameter and represents the minimum interval between a read and a write operation to any banks. See “Read/Write Interaction” on page 28.

The interaction interval for the WWd and WWs cases is t_{CC}. This is the column-to-column time parameter and represents the minimum interval between two write operations.

The interaction interval for the PWs case is $t_{RP} + t_{RCD-W}$. An activate operation must be inserted between the precharge and the write operation. The minimum interval between a precharge and an activate operation to a bank is t_{RP}. The minimum interval between an activate and a write operation to a bank is t_{RCD-W}.

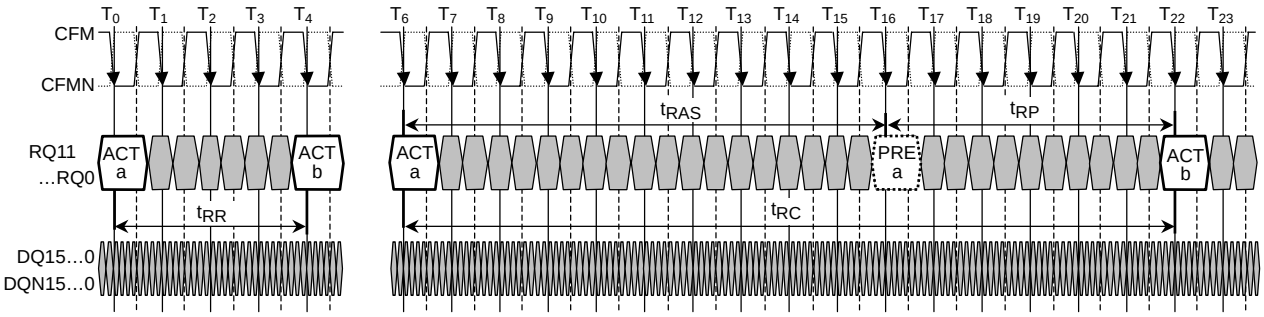
In Figure 7, the interaction interval for the APs case is t_{RAS}. This parameter is the minimum activate – to - precharge time to a bank.

The interaction intervals for the RPs and WPs cases are t_{RDP} and t_{WRD}, respectively. These are the read- or write-to-precharge time parameters to a bank.

The interaction interval for the PPd case is t_{PP}. This parameter is the precharge-to-precharge time and the minimum interval between precharge commands to different banks of a device.

The interaction interval for the PPs case is t_{RC}. This is the row cycle time parameter and the minimum interval between precharge commands to same banks of a device. An activate operation must be inserted between the two activate operations. This activate operation must be placed a time t_{RP} after the first, and a time t_{RAS} before the second precharge.

Figure 4. ACT-, RD-, WR-, PRE-to-ACT Packet Interactions

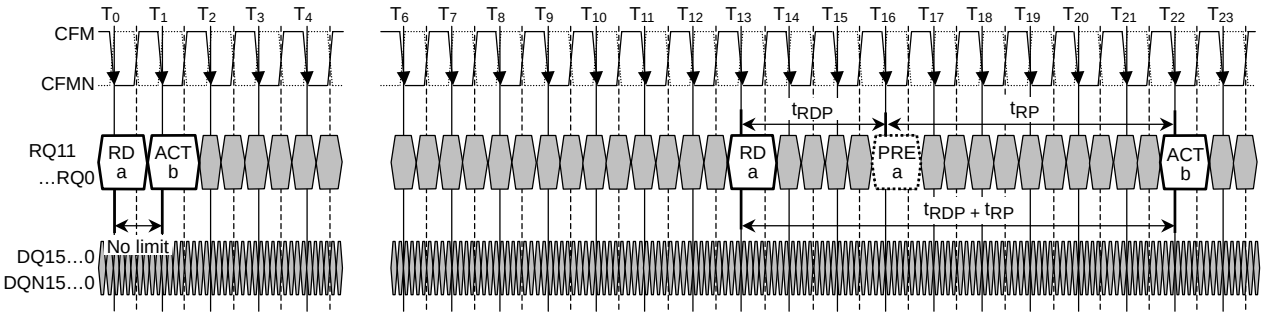


AAd Case (activate-activate-different bank)

a: ROWA Packet with ACT,Ba,Ra	Ba ≠ Bb
b: ROWA Packet with ACT,Bb,Rb	

AAs Case (activate-activate-same bank)

a: ROWA Packet with ACT,Ba,Ra	Ba = Bb
b: ROWA Packet with ACT,Bb,Rb	

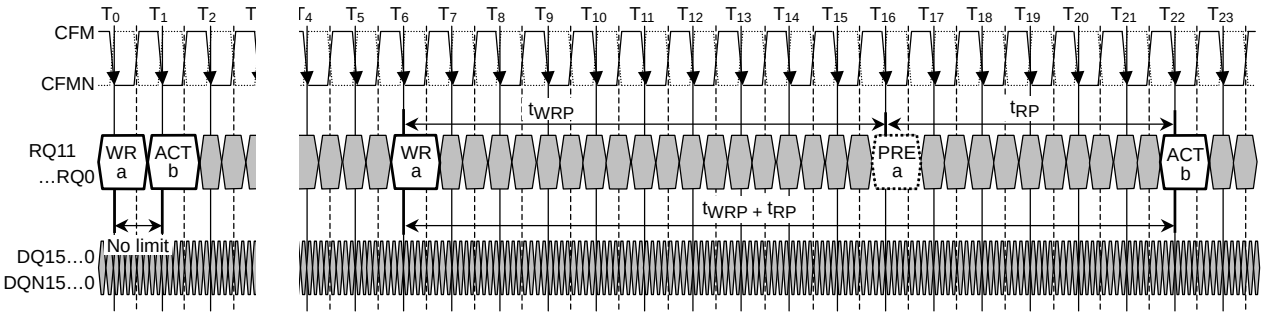


RAd Case (read-activate-different bank)

a: COL Packet with RD, Ba,Ra	Ba ≠ Bb
b: ROWA Packet with ACT,Bb,Rb	

RAs Case (read-activate-same bank)

a: COL Packet with RD,Ba,Ra	Ba = Bb
b: ROWA Packet with ACT,Bb,Rb	

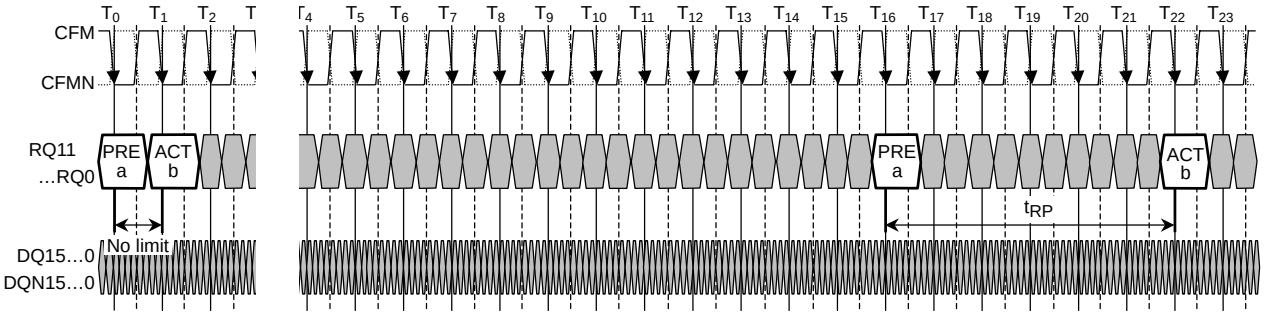


WAd Case (write-activate-different bank)

a: COL Packet with WR, Ba,Ra	Ba ≠ Bb
b: ROWA Packet with ACT,Bb,Rb	

WAs Case (write-activate-same bank)

a: COL Packet with WR,Ba,Ra	Ba = Bb
b: ROWA Packet with ACT,Bb,Rb	



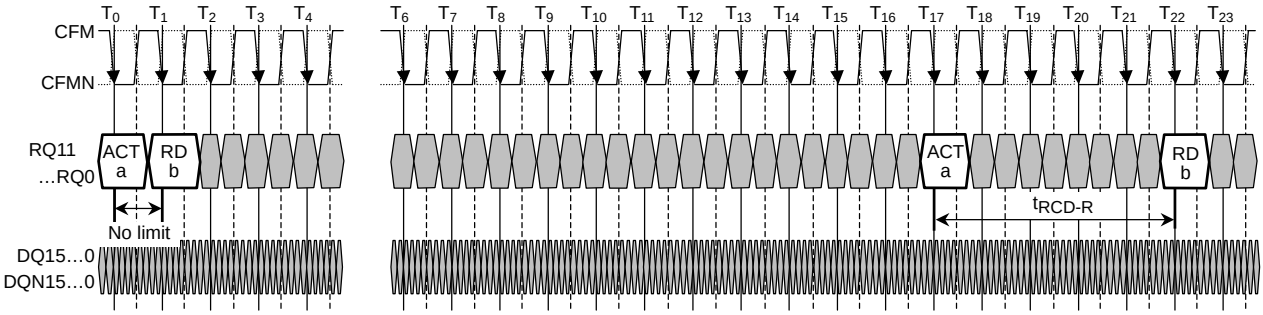
PAd Case (precharge-activate-different bank)

a: ROWP Packet with PRE, Ba	Ba ≠ Bb
b: ROWA Packet with ACT,Bb,Rb	

PAs Case (precharge-activate-same bank)

a: ROWP Packet with PRE,Ba	Ba = Bb
b: ROWA Packet with ACT,Bb,Rb	

Figure 5. ACT-, RD-, WR-, PRE-to-RD Packet Interactions

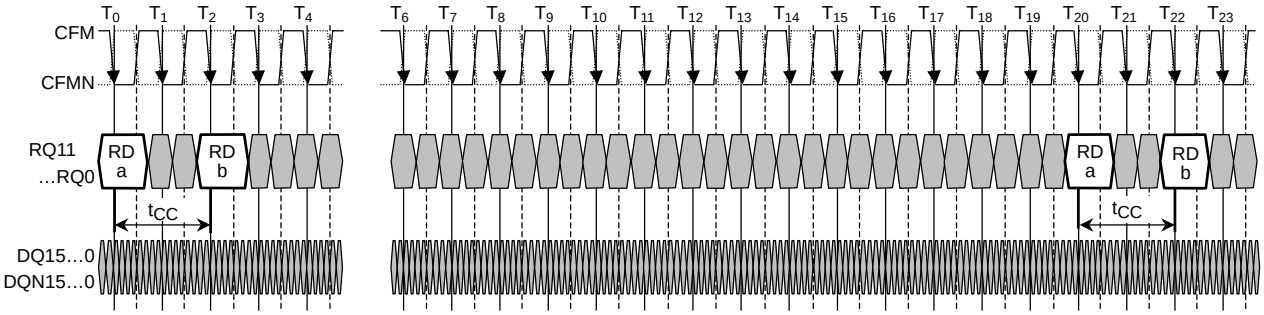


ARd Case (activate-read different bank)

a: ROWA Packet with ACT,Ba,Ra	Ba ≠ Bb
b: COL Packet with RD,Bb,Cb	

ARs Case (activate-read same bank)

a: ROWA Packet with ACT,Ba,Ra	Ba = Bb
b: COL Packet with RD,Bb,Cb	

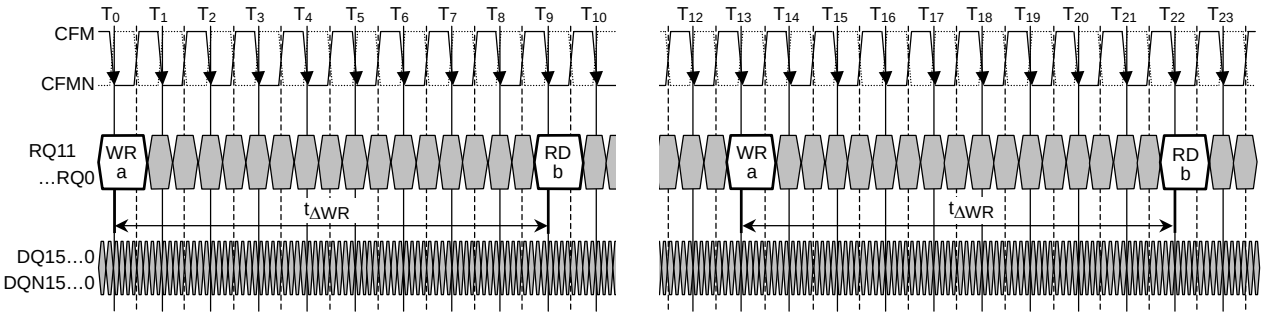


RRd Case (read-read different bank)

a: COL Packet with RD, Ba, Ca	Ba ≠ Bb
b: COL Packet with RD, Bb, Cb	

RRs Case (read-read same bank)

a: COL Packet with RD, Ba, Ca	Ba = Bb
b: COL Packet with RD, Bb, Cb	

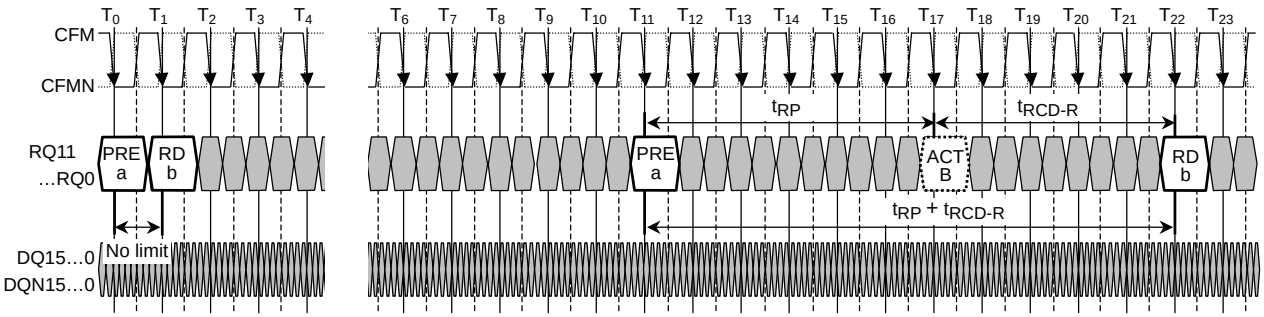


WRd Case (write-read different bank)

a: COL Packet with WR, Ba, Ca	Ba ≠ Bb
b: COL Packet with RD, Bb, Cb	

WRs Case (write-read same bank)

a: COL Packet with WR, Ba, Ca	Ba = Bb
b: COL Packet with RD, Bb, Cb	



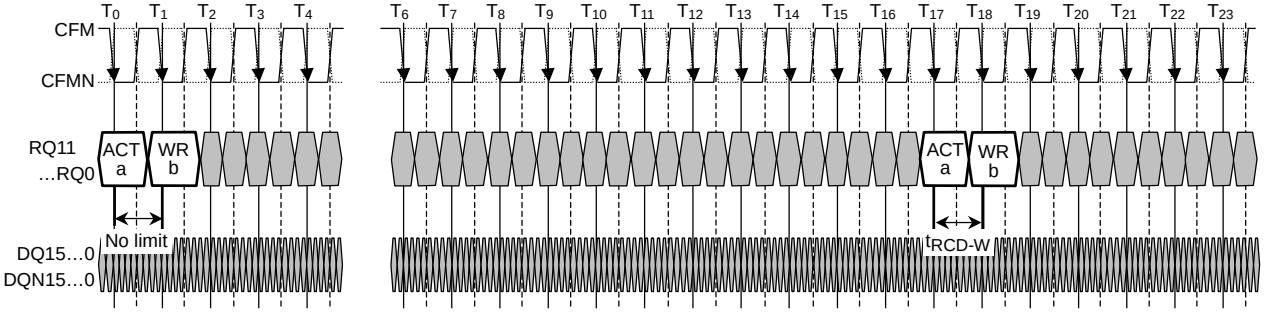
PRd Case (precharge-read different bank)

a: ROWP Packet with PRE, Ba	Ba ≠ Bb
b: COL Packet with RD, Bb, Cb	

PRs Case (precharge-read same bank)

a: ROWP Packet with PRE, Ba	Ba = Bb
b: COL Packet with RD, Bb, Cb	

Figure 6. ACT-, RD-, WR-, PRE-to-WR Packet Interactions

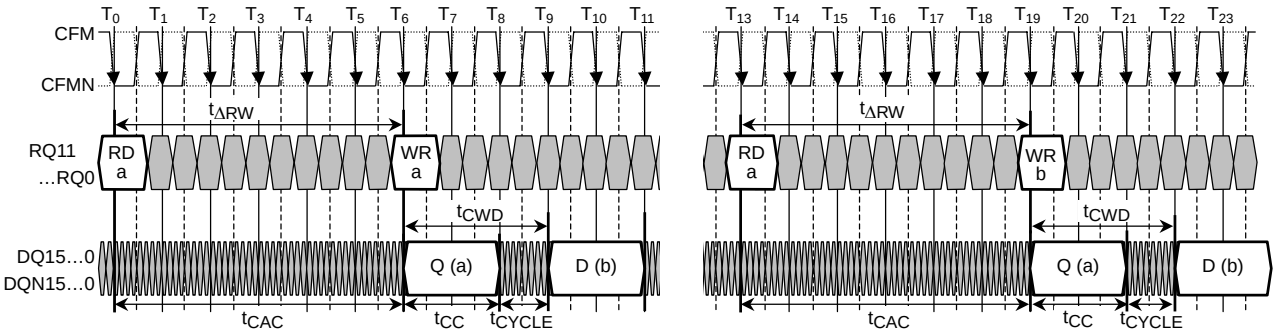


AWd Case (activate-write different bank)

a: ROWA Packet with ACT, Ba, Ra	Ba ≠ Bb
b: COL Packet with WR, Bb,Cb	

AWs Case (activate-write same bank)

a: ROWA Packet with ACT, Ba, Ra	Ba = Bb
b: COL Packet with WR, Bb, Cb	

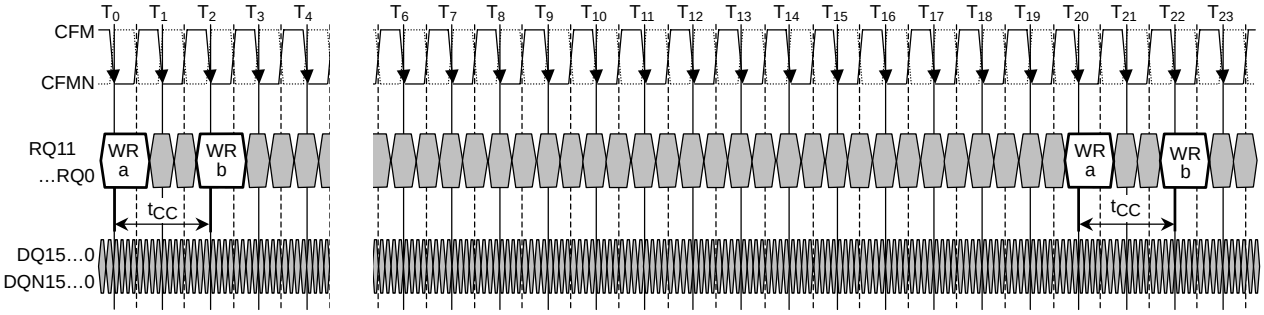


RWd Case (read-write different bank)

a: COL Packet with RD, Ba,Ra	Ba ≠ Bb
b: COL Packet with WR, Bb, Cb	

RWs Case (read-activate same bank)

a: COL Packet with RD, Ba, Ca	Ba = Bb
b: COL Packet with WR, Bb, Cb	

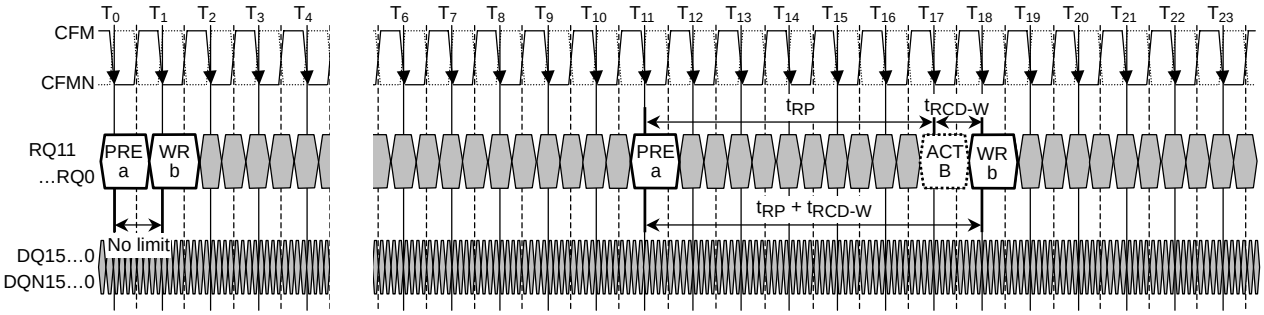


WWd Case (write-write different bank)

a: COL Packet with WR, Ba, Ca	Ba ≠ Bb
b: COL Packet with WR,Bb,Cb	

WWs Case (write-write same bank)

a: COP Packet with WR,Ba, Ca	Ba = Bb
b: COL Packet with WR, Bb, Cb	



PWd Case (precharge-write different bank)

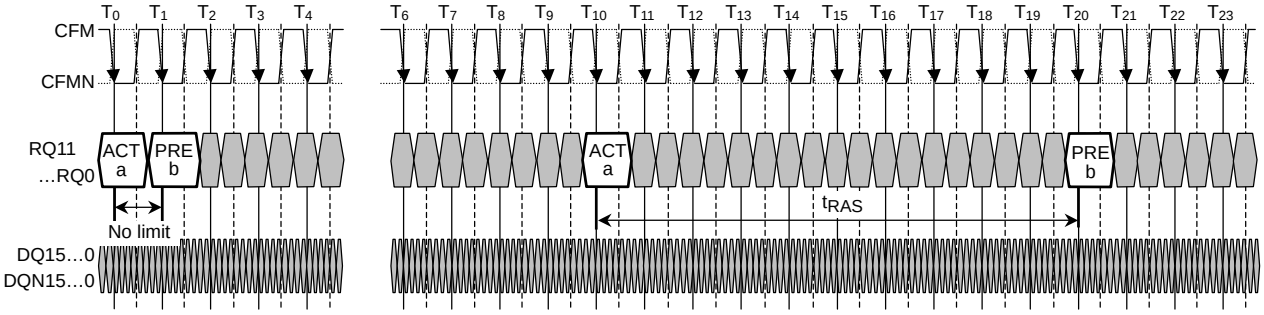
a: ROWP Packet with PRR, Ba	Ba ≠ Bb
b: COL Packet with WR, Bb,Cb	

PWs Case (precharge-write same bank)

a: ROWP Packet with PRE, Ba	Ba = Bb
b: COP Packet with WR, Bb, Cb	

Rev 0.85

Figure 7. ACT-, RD-, WR-, PRE-to-PRE Packet Interactions

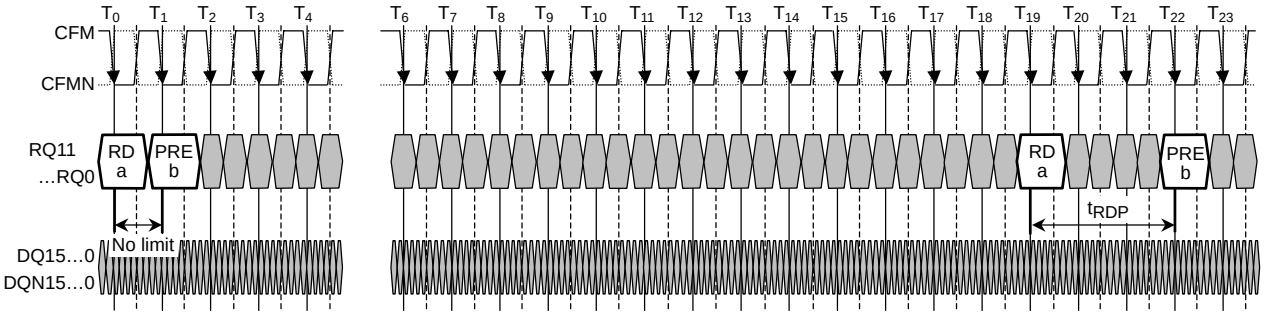


APd Case (activate-precharge different bank)

a: ROWA Packet with ACT, Ba, Ra	Ba ≠ Bb
b: ROWP Packet with PRE, Bb	

APs Case (activate-precharge same bank)

a: ROWA Packet with ACT, Ba, Ra	Ba = Bb
b: ROWP Packet with PRR, Bb	

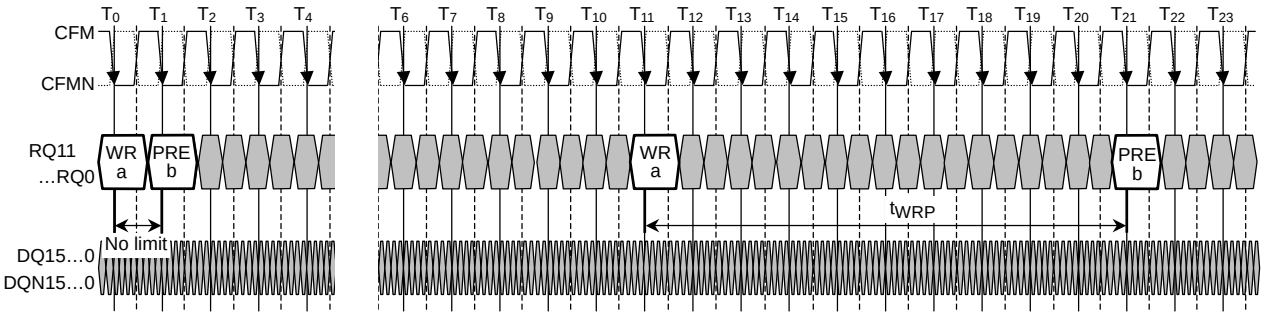


RPd Case (read-precharge different bank)

a: COL Packet with RD, Ba, Ca	Ba ≠ Bb
b: ROWP Packet with PRE, Bb	

RPs Case (read-precharge same bank)

a: COL Packet with RD, Ba, Ca	Ba = Bb
b: ROWP Packet with PRR, Bb	

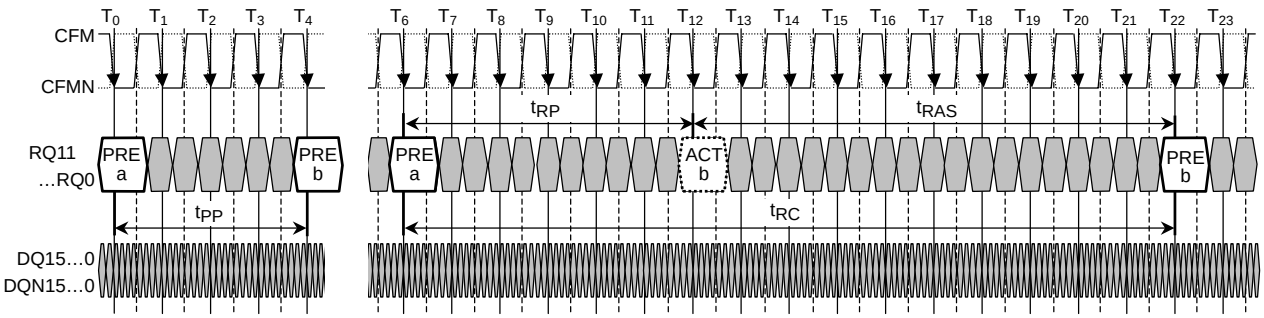


WPd Case (write-precharge different bank)

a: COL Packet with WR, Ba, Ca	Ba ≠ Bb
b: ROWP Packet with PRE, Bb	

WPs Case (write-precharge same bank)

a: COL Packet with WR, Ba, Ca	Ba = Bb
b: ROWP Packet with PRE, Bb	



PPd Case (precharge-precharge different bank)

a: ROWP Packet with PRE, Ba	Ba ≠ Bb
b: ROWP Packet with PRE, Bb	

PPs Case (precharge-precharge same bank)

a: ROWP Packet with PRE, Ba	Ba = Bb
b: ROWP Packet with PRE, Bb	

Dynamic Request Scheduling

Delay fields are present in the ROWA, COL, and ROWP packets. They permit the associated command to optionally wait for a time of one (or more) t_{CYCLE} before taking effect. This allows a memory controller more scheduling flexibility when issuing request packets. Figure 8 illustrates the use of the delay fields.

In the first timing diagram, a ROWA packet with an ACT command is present at cycle T_0. The DELA field is set to “1”. This request packet will be equivalent to a ROWA packet with an ACT command at cycle T_1 with the DELA field is set to “0”. This equivalence should be used when analyzing request packet interactions.

In the second timing diagram, a COL packet with a RD command is present at cycle T_0. The DELC field is set to “1”. This request packet will be equivalent to a COL packet with an RD command at cycle T_1 with the DELC field is set to “0”. This equivalence should be used when analyzing request packet interactions.

In a similar fashion, a COL packet with a WR command is present at cycle T_{12}. The DELC field is set to “1”. This request packet will be equivalent to a COL packet with a WR command at cycle T_{13} with the DELC field is set to “0”. This equivalence should be used when analyzing request packet interactions.

In the COL packet with a RD command example, the read data delay. t_{CAC} is measured between the Q read data packet and the virtual COL packet at cycle T_1.

Likewise, for the example with the COL packet with a WR command, the write data delay. t_{CWD} is measured between the D write data packet and the virtual COL packet at cycle T_{13}.

In the third timing diagram, a ROWP packet with a PRE command is present at cycle T_0. The DEL field (POP [1:0]) is set to “11”. This request packet will be equivalent to a ROWP packet with a PRE command at cycle T_1 with the DEL field is set to “10”, it will be equivalent to a ROWP packet with a PRE command at cycle T_2 with the DEL field is set to “01”, and it will be equivalent to a ROWP packet with a PRE command at cycle T_3 with the DEL field is set to “00”. This equivalence should be used when analyzing request packet interactions.

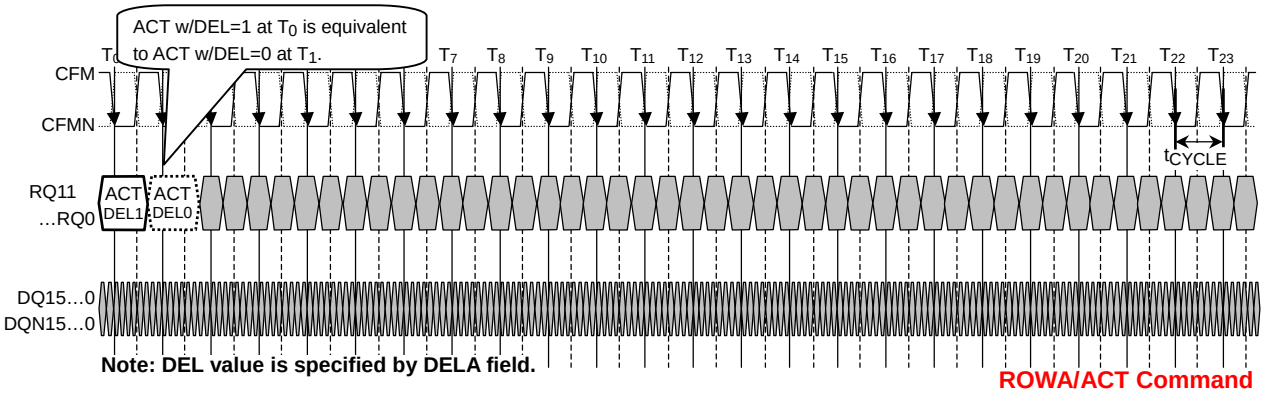
In the fourth timing diagram, a ROWP packet with a REFP command is present at cycle T_0. The DEL field (RA [7:6]) is set to “11”. This request packet will be equivalent to a ROWP packet with a REFP command at cycle T_1 with the DEL field is set to “10”, it will be equivalent to a ROWP packet with a REFP command at cycle T_2 with the DEL field is set to “01”, and it will be equivalent to a ROWP packet with a REFP command at cycle T_3 with the DEL field is set to “00”. This equivalence should be used when analyzing request packet interactions.

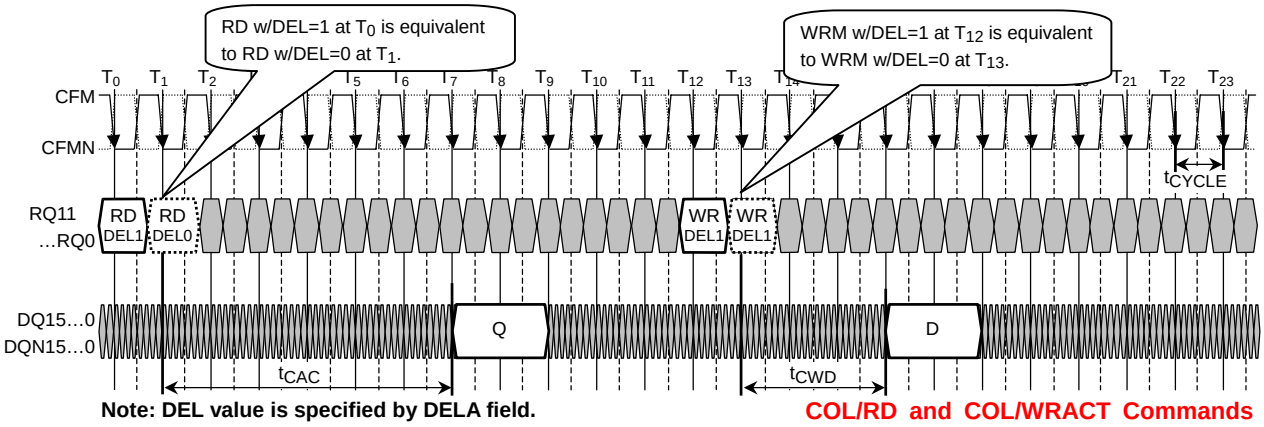
The two examples for the REFA and REFI commands are identical to the example just described for the REFP command.

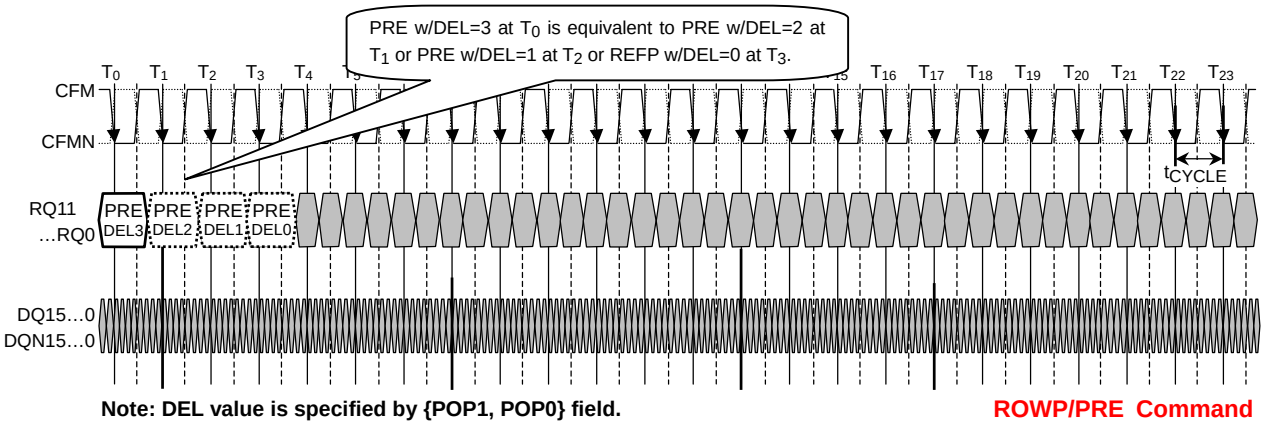
The ROWP packet allows two independent operations to be specified. A PRE precharge command uses the POP and BP fields, and the REFP, REFA, or REFI commands uses the ROP and RA fields. Both operations have an optional delay field (the POP field for the PRE command and the RA field with the REFP, REFA, or REFI commands). The two delay mechanisms are independent of one another. The POP field does not affect the timing of the REFP, REFA, or REFI commands, and the RA field does not affect the timing of the PRE command.

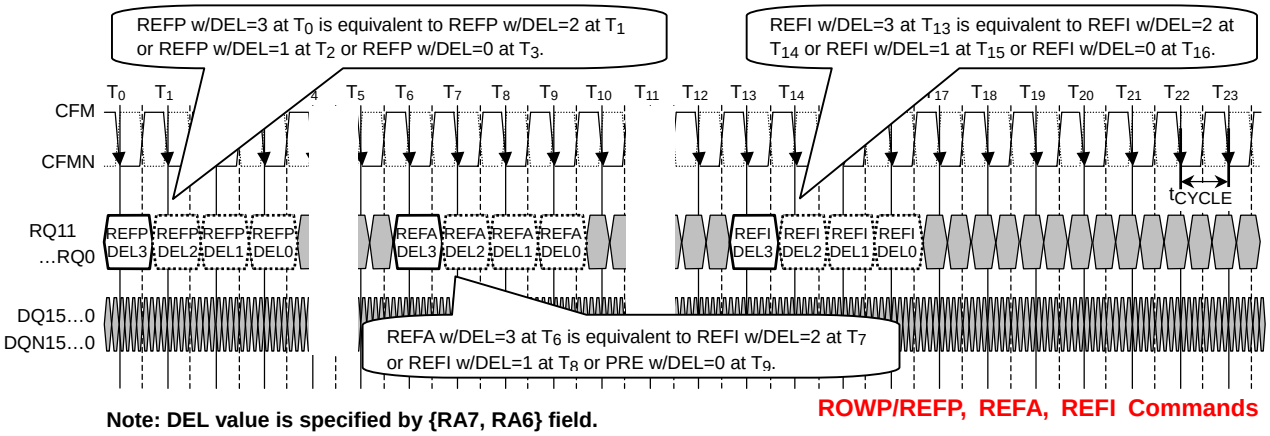
When the interactions of a ROWP packet are analyzed, it must be remembered that there are two independent commands specified, both of which may affect how soon the next request packet can be issued. The constraints from both commands in a ROWP packet must be considered, and the one that requires the longer time interval to the next request packet must be used by the memory controller. Furthermore, the two commands within a ROWP packet may not reference the same bank in the BP and RA fields.

Figure 8. Request Scheduling Examples









Memory Operations

Write Transactions

Figure 9 shows four examples of memory write transactions. A transaction is one or more request packets (and the associated data packets) needed to perform a memory access. The state of the memory core and the address of the memory access determine how many request packets are needed to perform the access.

The first timing diagram shows a page-hit write transaction. In this case, the selected bank is already open (a row is already present in the sense amp array for the bank). In addition, the selected row for the memory access matches the address of the row already sensed (a page hit). This comparison must be done in the memory controller. In this example, the access is made to row Ra of bank Ba.

In this case, write data may be directly written into the sense amp array for the bank, and row operations (activate or precharge) are not needed. A COL packet with WR command to column Ca1 of bank Ba is presented on edge T_0, and a second COL packet with WR command to column Ca2 of bank Ba is presented on edge T_2. Two write data packets D (a1) and D (a2) follow these COL packets after the write data delay t_{CWD}. The two COL packets are separated by the column-cycle time t_{CC}. This is also the length of each write data packet.

The second timing diagram shows an example of a page-miss write transaction. In this case, the selected bank is already open (a row is already present in the sense amp array for the bank). However, the selected row for the memory access does not match the address of the row already sensed (a page miss). This comparison must be done in the memory controller. In this example, the access is made to row Ra of bank Ba, and the bank contains a row other than Ra.

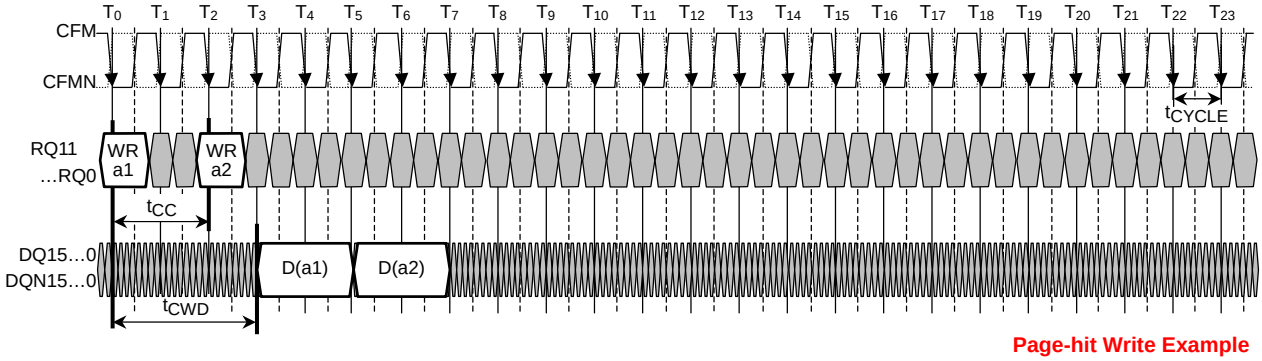
In this case, write data may not be directly written into the sense amp array for the bank. It is necessary to close the present row (precharge) and access the requested row (activate). A precharge command (PRE to bank Ba) is presented on edge T_0. An activate command (ACT to row Ra of bank Ba) is presented on edge T_6 a time t_{RP} later. A COL packet with WR command to column Ca1 of bank Ba is presented on edge T_7 a time $t_{RCD\text{-}W}$ later. A second COL packet with WR command to column Ca2 of bank Ba is presented on edge T_9. Two write data packets D (a1) and D (a2) follow these COL packets after the write data delay t_{CWD}. The two COL packets are separated by the column-cycle time t_{CC}. This is also the length of each write data packet.

The third timing diagram shows an example of a page-empty write transaction. In this case, the selected bank is already closed (no row is present in the sense amp array for the bank). No row comparison is necessary for this case; however, the memory controller must still remember that bank Ba has been left closed. In this example, the access is made to row Ra of bank Ba.

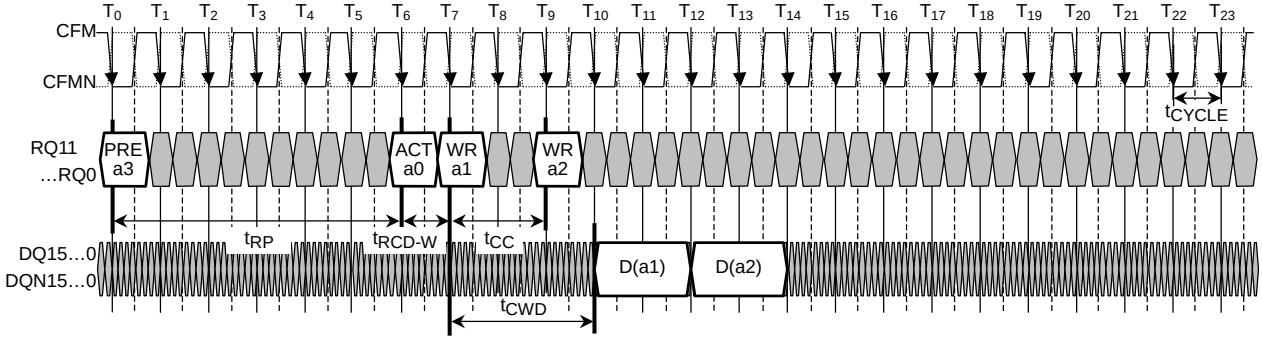
In this case, write data may not be directly written into the sense amp array for the bank. It is necessary to access the requested row (activate). An activate command (ACT to row Ra of bank Ba) is presented on edge T_0. A COL packet with WR command to column Ca1 of bank Ba is presented on edge T_1 a time t_{RCD-W} later. A second COL packet with WR command to column Ca2 of bank Ba is presented on edge T_3. Two write data packets D (a1) and D (a2) follow these COL packets after the write data delay t_{CWD}. The two COL packets are separated by the column-cycle time t_{CC}. This is also the length of each write data packet. After the final write command, it may be necessary to close the present row (precharge). A precharge command (PRE to bank Ba) is presented on edge T_{14} a time t_{WRP} after the last COL packet with a WR command. The decision whether to close the bank or leave it open is made by the memory controller and its page policy.

The fourth timing diagram shows another example of a page–empty write transaction. This is similar to the previous example except that only a single write command is presented, rather than two write commands. This example shows that even with a minimum length write transaction, the t_{RAS} parameter will not be a constraint. The t_{RAS} measures the minimum time between an activate command and a precharge command to a bank. This time interval is also constrained by the sum $t_{RCD-W} + t_{WRP}$ which will be larger for a write transaction. These two constraints (t_{RAS} and $t_{RCD-W} + t_{WRP}$) will be a function of the memory device's speed bin and the data transfer length (the number of write commands issued between the activate and precharge commands), and the t_{RAS} parameter could become a constraint for write transactions for future speed bins. In this example, the sum $t_{RCD-W} + t_{WRPs}$ is greater than t_{RAS} by the amount Δt_{RAS}.

Figure 9. Write Transactions

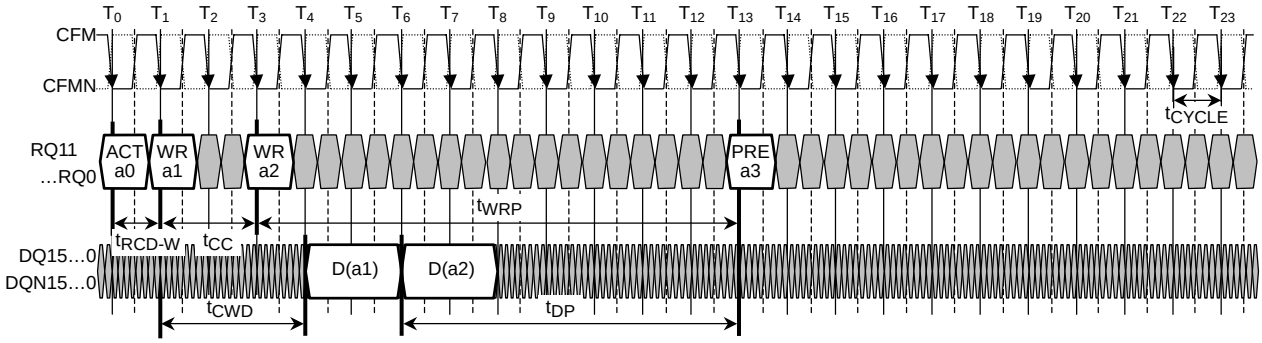


Page-hit Write Example



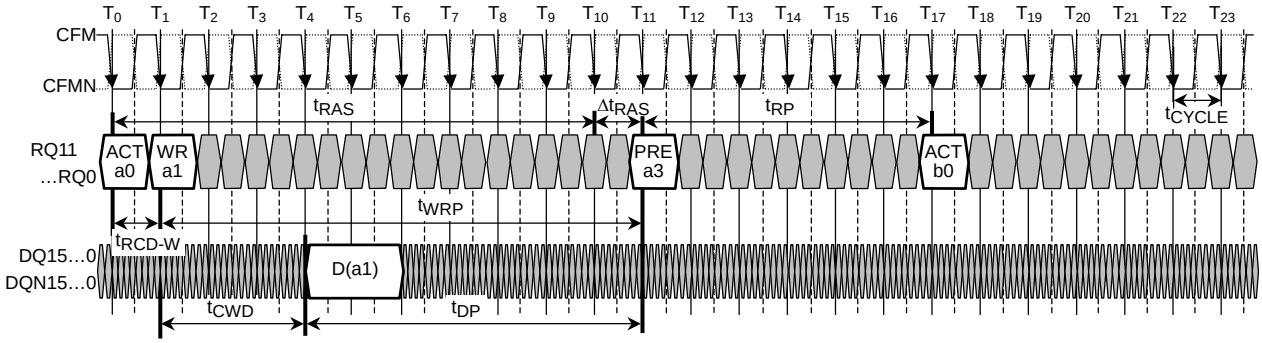
Transaction a: WR	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}

Page-miss Write Example



Transaction a: WR	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}

Page-empty Write Example



Transaction a: WR	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}
Bb=Ba → Transaction b: WR	b0 = {Bb, Rb}	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}	b3 = {Bb}

Page-empty Write Example – Core Limited

Read Transactions

Figure 10 shows four examples of memory read transactions. A transaction is one or more request packets (and the associated data packets) needed to perform a memory access. The state of the memory core and the address of the memory access determine how many request packets are needed to perform the access.

The first timing diagram shows a page-hit read transaction. In this case, the selected bank is already open (a row is already present in the sense amp array for the bank). In addition, the selected row for the memory access matches the address of the row already sensed (a page hit). This comparison must be done in the memory controller. In this example, the access is made to row Ra of bank Ba.

In this case, read data may be directly read from the sense amp array for the bank, and no row operations (activate or precharge) are needed. A COL packet with RD command to column Ca1 of bank Ba is presented on edge T_0, and a second COL packet with RD command to column Ca2 of bank Ba is presented on edge T_2. Two read data packets Q (a1) and Q (a2) follow these COL packets after the read data delay t_{CAC}. The two COL packets are separated by the column-cycle time t_{CC}. This is also the length of each read data packet.

The second timing diagram shows an example of a page-miss read transaction. In this case, the selected bank is already open (a row is already present in the sense amp array for the bank). However, the selected row for the memory access does not match the address of the row already sensed (a page miss). This comparison must be done in the memory controller. In this example, the access is made to row Ra of bank Ba, and the bank contains a row other than Ra.

In this case, read data may not be directly read from the sense amp array for the bank. It is necessary to close the present row (precharge) and access the requested row (activate). A precharge command (PRE to bank Ba) is presented on edge T_0. An activate command (ACT to row Ra of bank Ba) is presented on edge T_6 a time t_{RP} later. A COL packet with RD command to column Ca1 of bank Ba is presented on edge T_{11} a time t_{RCD-R} later. A second COL packet with RD command to column Ca2 of bank Ba is presented on edge T_{13}. Two read data packets Q(a1) and Q(a2) follow these COL packets after the read data delay t_{CAC}. The two COL packets are separated by the column-cycle time t_{CC}. This is also the length of each read data packet.

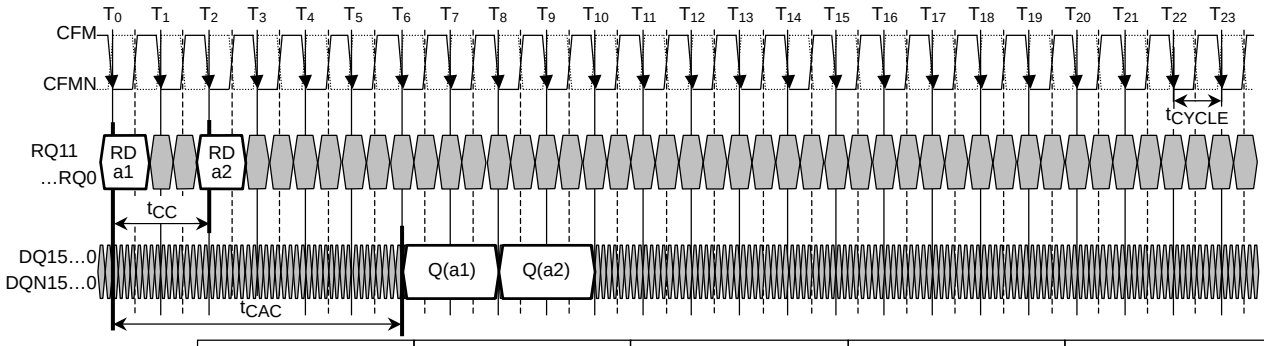
The third timing diagram shows an example of a page-empty write transaction. In this case, the selected bank is already closed (no row is present in the sense amp array for the bank). No row comparison is necessary for this case; however, the memory controller must still remember that bank Ba has been left closed. In this example, the access is made to row Ra of bank Ba.

In this case, read data may not be directly read from the sense amp array for the bank. It is necessary to access the requested row (activate). An activate command (ACT to row Ra of bank Ba) is presented on edge T_0. A COL packet with RD command to column Ca1 of bank Ba is presented on edge T_5 a time t_{RCD-R} later. A second COL packet with RD command to column Ca2 of bank Ba is presented on edge T_7. Two read data packets Q (a1) and Q (a2) follow these COL packets after the read data delay t_{CAC}. The two COL packets are separated by the column-cycle time t_{CC}. This is also the length of each read data packet. After the final read command, it may be necessary to close the present row (precharge). A precharge command — PRE to bank Ba — is presented on edge T_{10} a time t_{RDP} after the last COL packet with a RD command. Whether the bank is closed or left open depends on the memory controller and its page policy.

The fourth timing diagram shows another example of a page-empty read transaction. This is similar to the previous example except that it uses one read command instead of two read commands. In this case, the core parameter t_{RAS} may also be a constraint upon when the precharge command may be issued.

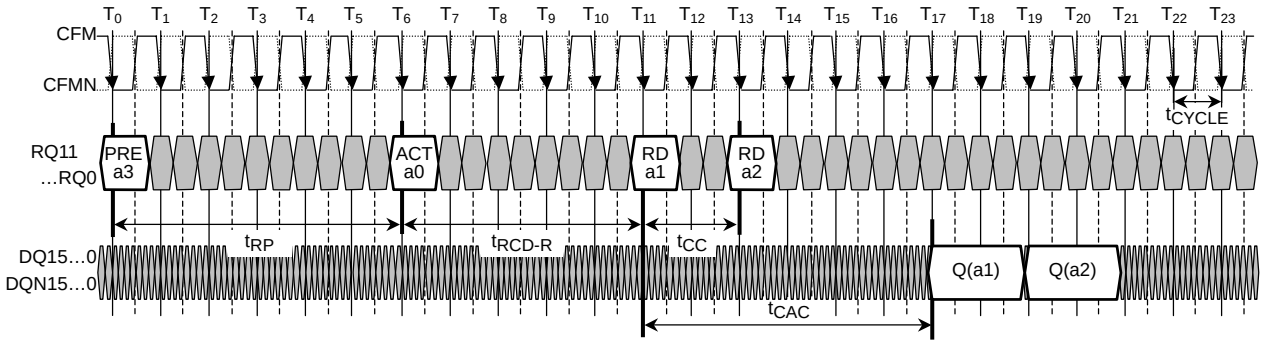
The t_{RAS} measures the minimum time between an activate command and a precharge command to a bank. This time interval is also constrained by the sum $t_{RCD-R} + t_{RDP}$ and must be set to whichever is larger. These two constraints (t_{RAS} and $t_{RCD-R} + t_{RDP}$) will be a function of the memory device's speed bin and the data transfer length (the number of read commands issued between the activate and precharge commands). In this example, the t_{RAS} is greater than the sum $t_{RCD-R} + t_{RDP}$ by the amount Δt_{RDP}.

Figure 10. Read Transactions



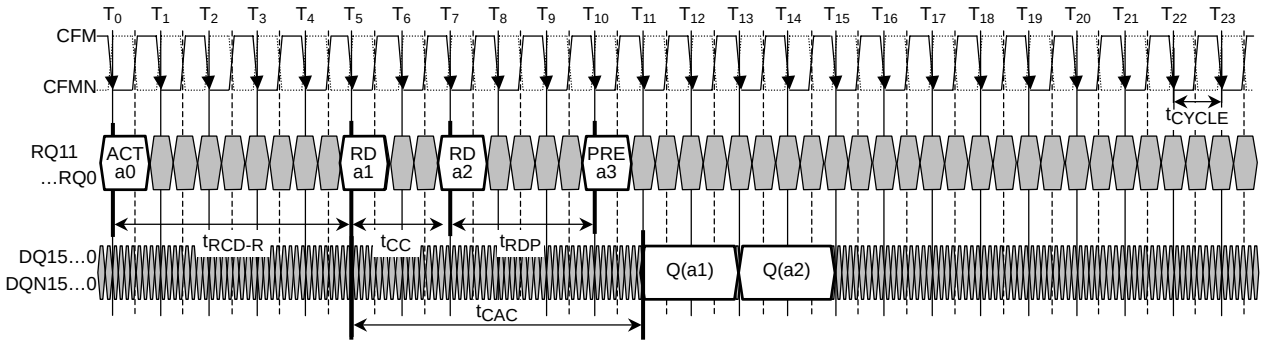
Transaction a: RD	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}

Page-hit Read Example



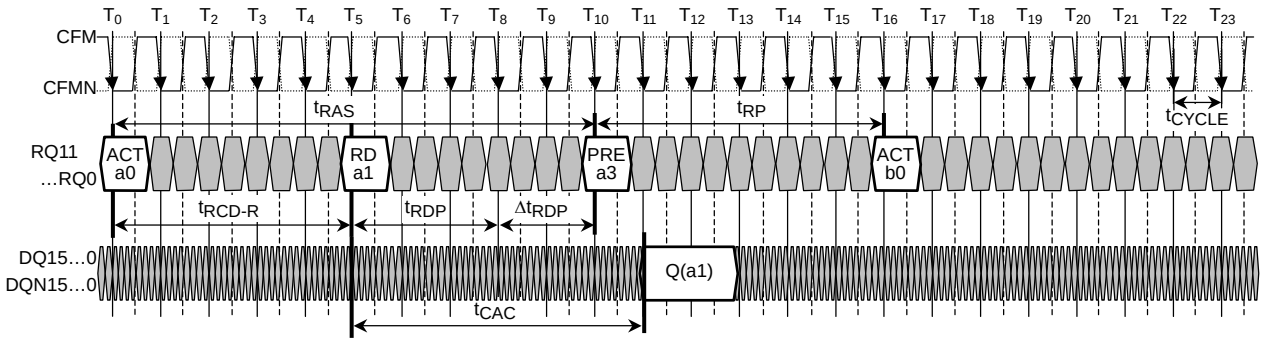
Transaction a: RD	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}

Page-miss Read Example



Transaction a: RD	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}

Page-empty Read Example



Transaction a: RD	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}
Bb=Ba → Transaction b: RD	b0 = {Bb, Rb}	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}	b3 = {Bb}

Page-empty Read Example – Core Limited

Interleaved Transactions

Figure 11 shows two examples of interleaved transactions. Interleaved transactions are overlapped with one another; a transaction is started before an earlier one is completed.

The timing diagram at the top of the figure shows interleaved write transactions. Each transaction assumes a page-empty access; that is, a bank is in a closed state prior to an access, and is precharged after the access. With this assumption, each transaction requires the same number of request packets at the same relative positions. If banks were allowed to be in an open state, then each transaction would require a different number of request packets depending upon whether the transaction was page-empty, page-hit, or page-miss. This situation is more complicated for the memory controller, and will not be analyzed in this document.

In the interleaved page–empty write example, there are four sets of request pins RQ11…RQ0 shown along the left side of the timing diagram. The first three show the timing slots used by each of the three requests packet types (ACT, COL and PRE), and the fourth set (ALL) shows the previous three merged together. This allows the pattern used for allocating request slots for the different packets to be seen more clearly.

The slots at {T_0, T_4, T_8, T_{12}, ...} are used for ROWA packets with ACT commands. This spacing is determined by the t_{RR} parameter. There should not be interference between the interleaved transactions due to resource conflicts because each bank address — Ba, Bb, Bc, Bd, and Be — is assumed to be different from another. If two of the bank addresses are the same, the later transaction would need to wait until the earlier transaction had completed its precharge operation. Five different banks are needed because the effective t_{RC} ($t_{RC} + \Delta t_{RC}$) is $20 \times t_{CYCLE}$.

The slots at {T_1, T_3, T_5, T_7, T_9, T_{11}, ...} are used for COL packets with WR commands. This frequency of the COL packet spacing is determined by the t_{CC} parameter and by the fact that there are two column accesses per row access. The phasing of the COL packet spacing is determined by the $t_{RCD\text{-}W}$ parameter. If the value of $t_{RCD\text{-}W}$ required the COL packets to occupy the same request slots as the ROWA packets (this case is not shown), the DELC field in the COL packet could be used to place the COL packet one t_{CYCLE} earlier.

The DQ bus slots at {T_7, T_9, T_{11}, T_{13}, ...} carry the write data packets { D (a1), D (a2), D (b1), D (b2), ... }. Two write data packets are written to a bank in each transaction. The DQ bus is completely filled with write data; no idle cycles need to be introduced because there are no resource conflicts in this example.

The slots at {T_{14}, T_{18}, T_{22}, ...} are used for ROWP packets with PRE commands. This frequency of ROWP packet spacing is determined by the t_{PP} parameter. The phasing of the ROWP packet spacing is determined by the t_{WRP} parameter. If the value of t_{WRP} required the ROWP packets to occupy the same request slots as the ROWA or COL packets already assigned (this case is not shown), the delay field in the ROWP packet could be used to place the ROWP packet one or more t_{CYCLE} earlier.

There is an example of an interleaved page-empty read at the bottom of the figure. As before, there are four sets of request pins RQ11…RQ0 shown along the left side of the timing diagram, allowing the pattern used for allocating request slots for the different packets to be seen more clearly.

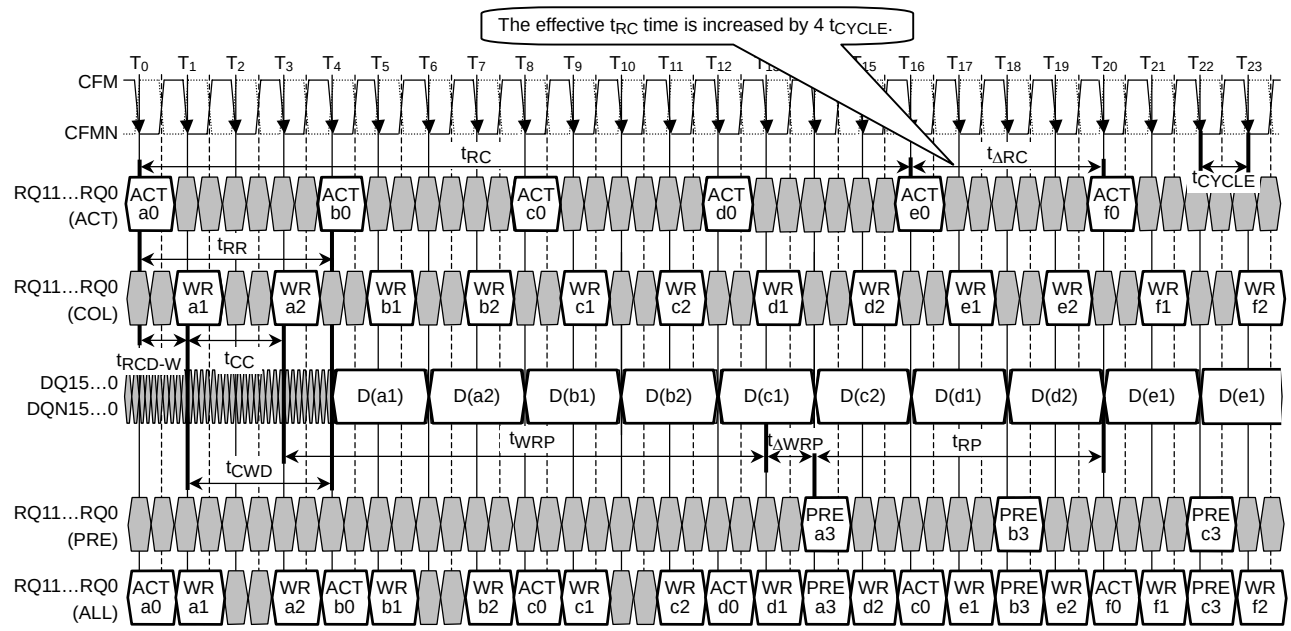
The slots at {T_0, T_4, T_8, T_{12}, ...} are used for ROWA packets with ACT commands. This spacing is determined by the t_{RR} parameter. There should not be interference between the interleaved transactions due to resource conflicts because each bank address — Ba, Bb, Bc, and Bd — is assumed to be different from another. Four different banks are needed because the effective t_{RC} is $16 \times t_{CYCLE}$.

The slots at {T_5, T_7, T_9, T_{11}, ...} are used for COL packets with RD commands. This frequency of the COL packet spacing is determined by the t_{CC} parameter and by the fact that there are two column accesses per row access. The phasing of the COL packet spacing is determined by the t_{RCD-R} parameter. If the value of t_{RCD-R} required the COL packets to occupy the same request slots as the ROWA packets (this case is not shown), the DELC field in the COL packet could be used to place the packet one t_{CYCLE} earlier.

The DQ bus slots at {T_{11}, T_{13}, T_{15}, T_{17}, ...} carry the read data packets { Q (a1), Q (a2), Q (b1), Q (b2), ...}. Two read data packets are read from a bank in each transaction. The DQ bus is completely filled with read data — that is, no idle cycles need to be introduced because there are no resource conflicts in this example.

The slots at {T_{10}, T_{14}, T_{18}, T_{22}, ...} are used for ROWP packets with PRE commands. This frequency of the ROWP packet spacing is determined by the t_{PP} parameter. The phasing of the ROWP packet spacing is determined by the t_{RDP} parameter. If the value of t_{RDP} required the ROWP packets to occupy the same request slots as the ROWA or COL packets already assigned (this case is not shown), the delay field in the ROWP packet could be used to place the ROWP packet one or more t_{CYCLE} earlier.

Figure 11. Interleaved Transactions

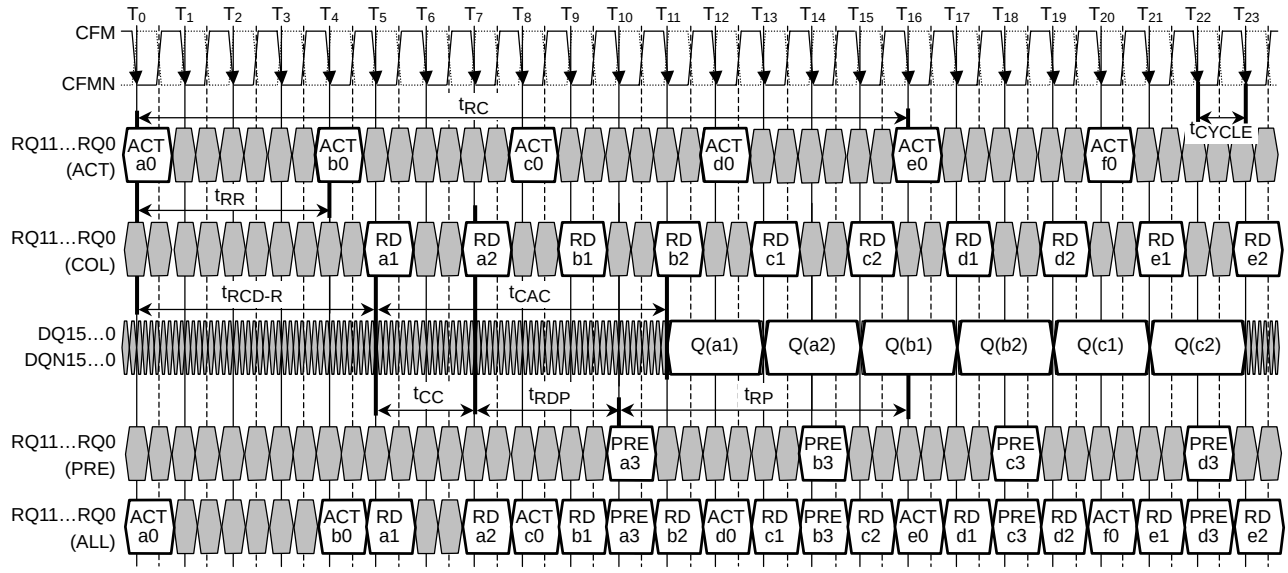


Ba,Bb,Bc,Bd,Be are different banks.

Bf = Ba

Transaction a: WR	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}
Transaction b: WR	b0 = {Bb, Rb}	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}	b3 = {Bb}
Transaction c: WR	c0 = {Bc, Rc}	c1 = {Bc, Cc1}	c2 = {Bc, Cc2}	c3 = {Bc}
Transaction d: WR	d0 = {Bd, Rd}	d1 = {Bd, Cd1}	d2 = {Bd, Cd2}	d3 = {Bd}
Transaction e: WR	e0 = {Be, Re}	e1 = {Be, Ce1}	e2 = {Be, Ce2}	e3 = {Be}
Transaction f: WR	f0 = {Bf, Rf}	f1 = {Bf, Cf1}	f2 = {Bf, Cf2}	f3 = {Bf}

Interleaved Page-empty Write Example



Ba,Bb,Bc,Bd are different banks.

Be = Ba

Transaction a: RD	a0 = {Ba, Ra}	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	a3 = {Ba}
Transaction b: RD	b0 = {Bb, Rb}	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}	b3 = {Bb}
Transaction c: RD	c0 = {Bc, Rc}	c1 = {Bc, Cc1}	c2 = {Bc, Cc2}	c3 = {Bc}
Transaction d: RD	d0 = {Bd, Rd}	d1 = {Bd, Cd1}	d2 = {Bd, Cd2}	d3 = {Bd}
Transaction e: RD	e0 = {Be, Re}	e1 = {Be, Ce1}	e2 = {Be, Ce2}	e3 = {Be}

Interleaved Page-empty Read Example

Read/Write Interaction

The previous section described overlapped read transactions and overlapped write transactions in isolation. This section will describe the interaction of read and write transactions and the spacing required to avoid channel and core resource conflicts.

Figure 12 shows a timing diagram (top) for the first case, a write transaction followed by a read transaction. Two COL packets with WR commands are presented on cycles T_0 and T_2. The write data packets are presented a time t_{CWD} later on cycles T_4 and T_6. The device requires a time $t_{\Delta WR}$ after the second COL packet with a WR command before a COL packet with a RD command may be presented. Two COL packets with RD commands are presented on cycles T_{11} and T_{13}. The read data packets are returned a time t_{CAC} later on cycles T_{17} and T_{19}. The time $t_{\Delta WR}$ is required for turning around internal bi-directional interconnections (inside the device). This time must be observed regardless of whether the write and read commands are directed to the same bank or different banks. A gap $t_{WR-BUB,\ XDR\ DRAM}$ will appear on the DQ bus between the end of the D (a2) packet and the beginning of the Q (b1) packet (measured at the appropriate packet reference points). The size of this gap can be evaluated by calculating the difference between cycles T_2 and T_{17} using the two timing paths:

$$t_{WR-BUB,XDRDRAM} = t_{\Delta WR} + t_{CAC} - t_{CWD} - t_{CC}$$

In this example, the value of $t_{WR-BUB,\ XDR\ DRAM}$ is greater than its minimum value of $t_{WR-BUB,\ XDR\ DRAM,\ MIN}$. The values of $t_{\Delta WR}$ and t_{CAC} are equal to their minimum values.

In the second case, the timing diagram displayed at the bottom of Figure 12 illustrates a read transaction followed by a write transaction. Two COL packets with RD commands are presented on cycles T_0 and T_2. The read data packets are returned a time t_{CAC} later on cycles T_6 and T_8. The device requires a time $t_{\Delta RW}$ after the second COL packet with a RD command before a COL packet with a WR command may be presented. Two COL packets with WR commands are presented on cycles T_{10} and T_{12}. The write data packets are presented a time t_{CWD} later on cycles T_{13} and T_{15}. The time $t_{\Delta RW}$ is required for turning around the external DQ bi-directional interconnections (outside the device). This time must be observed regardless whether the read and write commands are directed to the same bank or different banks. The time $t_{\Delta RW}$ depends upon four timing parameters, and may be evaluated by calculating the difference between cycles T_2 and T_{13} using the two timing paths:

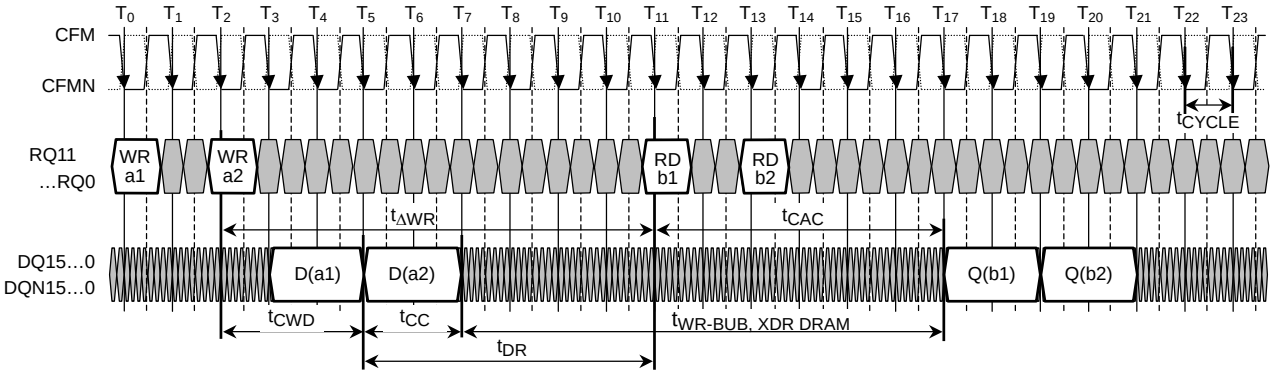
$$t_{\Delta RW} + t_{CWD} = t_{CAC} + t_{CC} + t_{RW-BUB,XDRDRAM}$$

or

$$t_{\Delta RW} = (t_{CAC} - t_{CWD}) + t_{CC} + t_{RW-BUB,XDRDRAM}$$

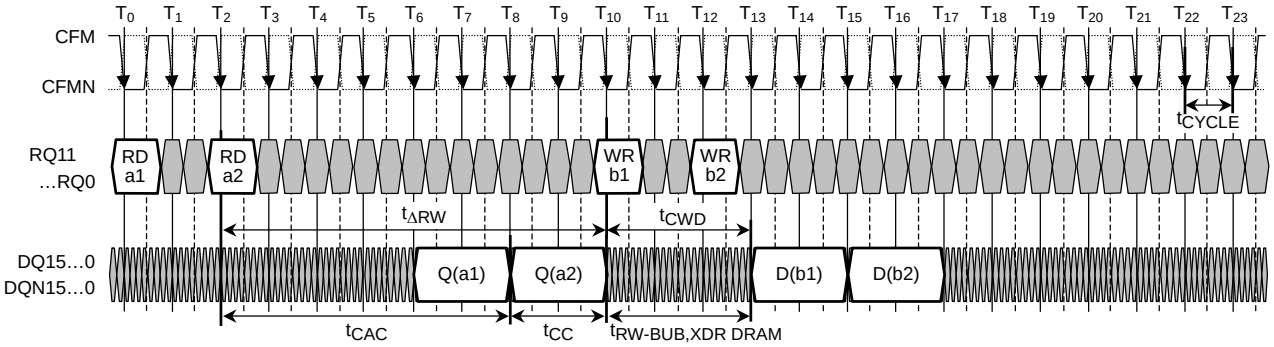
In this example, the values of $t_{\Delta RW}$, t_{CAC}, t_{CWD}, t_{CC}, and $t_{RW-BUB,XDR\ DRAM}$ are equal to their minimum values.

Figure 12. Write/Read Interaction



Transaction a: WR	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}
Transaction b: RD	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}

Write/Read Turnaround Example



Transaction a: WR	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}
Transaction b: RD	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}

Read/Write Turnaround Example

Propagation Delay

Figure 13 shows two timing diagrams that display the system-level timing relationships between the memory component and the memory controller.

The timing diagram at the top of the figure shows the case of a write-read-write command and data at the memory component. In this case, the timing will be identical to what has already been shown in the previous sections; i.e. with all timing measured at the pins of the memory component. This timing diagram was produced by merging portions of the top and bottom timing diagrams in Figure 12.

The example shown is that of a single COL packet with a write command, followed by a single COL packet with a read command, followed by a second COL packet with a write command. These accesses all assume a page-hit to an open bank.

A timing interval $t_{\Delta WR}$ is required between the first WR command and the RD command, and a timing interval $t_{\Delta RW}$ is required between the RD command and the second WR command. There is a write data delay t_{CWD} between each WR command and the associated write data packet D. There is a read data delay t_{CAC} between the RD command and the associated read data packet Q. In this example, all timing parameters have assumed their minimum values except $t_{WR-BUB,XDR\ DRAM}$.

The lower timing diagram in the figure shows the case where timing skew is present between the memory controller and the memory component. This skew is the result of the propagation delay of signal wavefronts on the wires carrying the signals.

The example in the lower diagram assumes that there is a propagation delay of t_{PD-RQ} along both the RQ wires and the CFM/CFMN clock wires between the memory controller and the memory component (the value of t_{PD-RQ} used here is $1*t_{CYCLE}$). Note that in an actual system the t_{PD-RQ} value will be different for each memory component connected to the RQ wires.

In addition, it is assumed that there is a propagation delay t_{PD-D} along the DQ/DQN wires between the memory controller and the memory component (the direction in which write data travels, and it is assumed that there is the same propagation delay t_{PD-Q} along the DQ/DQN wires between the memory component and the memory controller (the direction in which read data travels). The sum of these two propagation delays is also denoted by the timing parameter $t_{PD,CYC} = t_{PD-D} + t_{PD-Q}$.

As a result of these propagation delays, the position of packets will have timing skews that depend upon whether they are measured at the pins of the memory controller or the pins of the memory component. For example, the CFM/CFMN signals at the pins of the memory component are t_{PD-RQ} later than at the pins of the memory controller. This is shown by the cycle numbering of the CFM/CFMN signals at the two locations — in this example cycle T_1 at the memory controller aligns with cycle T_0 at the memory component.

All the request packets on the RQ wires will have a t_{PD-RQ} skew at the memory component relative to the memory controller in this example. Because the t_{PD-D} propagation delay of write data matches the t_{PD-RQ} propagation delay of the write command, the controller may issue the write data packet D(a0) relative to the COL packet with the first write command “WR a0” with the normal write data delay t_{CWD}. If the propagation delays between the memory controller and memory component were different for the RQ and DQ buses (not shown in this example), the write data delay at the memory controller would need to be adjusted.

A propagation delay is seen by the read command — that is, the read command will be delayed by a t_{PD-RQ} skew at the memory component relative to the memory controller. The memory component will return the read data packet Q(b0) relative to this read command with the normal read data delay t_{CAC} (at the pins of the memory component).

The read data packet will be skewed by an additional propagation delay of t_{PD-Q} as it travels from the memory component back to the memory controller. The effective read data delay measured between the read command and the read data at the memory controller will be $t_{CAC} + t_{PD-RQ} + t_{PD-Q}$.

The t_{PD-RQ} factor is caused by the propagation delay of the request packets as they travel from memory controller to memory component. The t_{PD-Q} factor is caused by the propagation delay of the read data packets as they travel from memory component to memory controller.

Propagation Delay (continue)

All timing parameters will be equal to their minimum values except $t_{WR-BUB,XDR\ DRAM}$ (as in the top diagram), and the timing parameters $t_{RW-BUB,XDR\ DRAM}$ and $t_{\Delta RW}$. These will be larger than their minimum values by the amount ($t_{PD,CYC}-t_{PD,CYC,MIN}$), where $t_{PD,CYC} = t_{PD-D} + t_{PD-Q}$. This may be seen by evaluating the two timing paths between cycle T_9 at the Controller and cycle T_{21} at the XDR DRAM:

$t_{\Delta RW} + t_{PD-RQ} + t_{CWD} = t_{PD-RQ} + t_{CAC} + t_{CC} + t_{RW-BUB,XDR\ DRAM}$

or

$t_{\Delta RW} = (t_{CAC}-t_{CWD}) + t_{CC} + t_{RW-BUB,XDR\ DRAM}$

The following relationship was shown for Figure 12.

$t_{\Delta RW,MIN} = (t_{CAC}-t_{CWD}) + t_{CC} + t_{RW-BUB,XDRDRAM,MIN}$

or

$(t_{\Delta RW}-t_{\Delta RW,MIN}) = (t_{RW-BUB,XDR\ DRAM}-t_{RW-BUB,XDRDRAM,MIN})$

In other words, the two timing parameters $t_{RW-BUB,XDR\ DRAM}$ and $t_{\Delta RW}$ will change together. The relationship of this change to the propagation delay $t_{PD,CYC}$ (= $t_{PD-D} + t_{PD-Q}$) can be derived by looking at the two timing paths from T_{15} to T_{21} at the XDR DRAM:

$t_{PD-Q} + t_{CC} + t_{RW-BUB,} \times 10 + t_{PD-D} = t_{CC} + t_{RW-BUB,XDR\ DRAM}$

or

$t_{RW-BUB,XDR\ DRAM} = t_{RW-BUB,} \times 10 + t_{PD-D} + t_{PD-Q}$

or

$t_{RW-BUB,\ XDR\ DRAM} = t_{RW-BUB,} \times 10 + t_{PD,CYC}$

in a system with minimum propagation delays:

$t_{RW-BUB,XDRDRAM,MIN} = t_{RW-BUB,} \times 10 + t_{PD,CYC,MIN}$

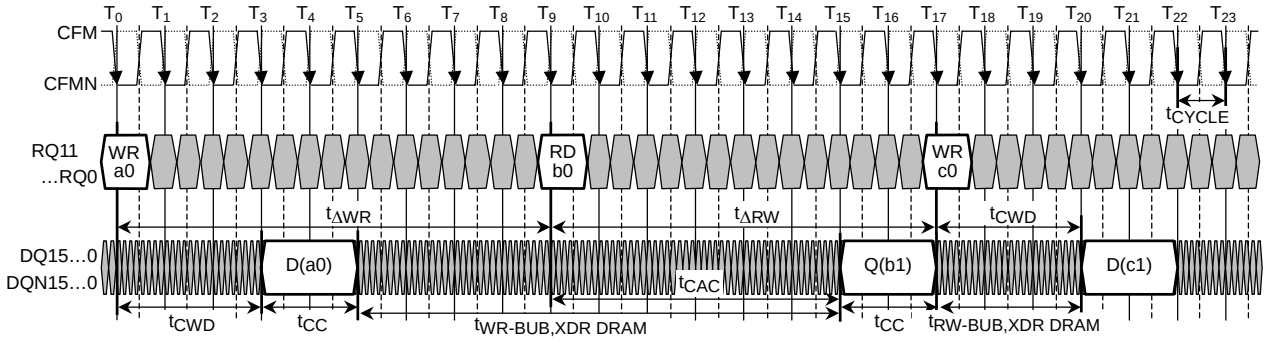
and since $t_{RW-BUB,} \times 10$ is equal to $t_{RW-BUB,YRAC,MIN}$ in both cases, the following is true:

$(t_{PD,CYC}-t_{PD,CYC,MIN}) = (t_{RW-BUB,XDR\ DRAM}-t_{RW-BUB,XDRDRAM,MIN}) = (t_{\Delta RW}-t_{\Delta RW,MIN})$

In other words, the values of the $t_{RW-BUB,XDR\ DRAM,MIN}$ and $t_{\Delta RW,MIN}$ timing parameters correspond to the value of $t_{PD,CYC,MIN}$ for the system (this is equal to one t_{CYCLE}). As $t_{PD,CYC}$ is increased from this minimum value, $t_{RW-BUB,XDR\ DRAM}$ and $t_{\Delta RW}$ increase from their minimum values by an equivalent amount.

Figure 13. Propagation Delay

XDR DRAM

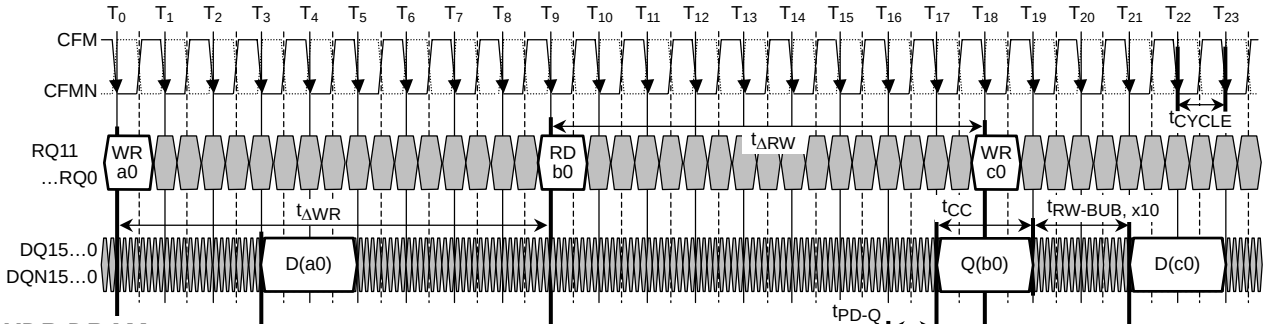


Transaction a: WR	a0 = {Ba, Ca0}
Transaction b: RD	b0 = {Bb, Cb0}
Transaction c: WR	c0 = {Bc, Cc0}

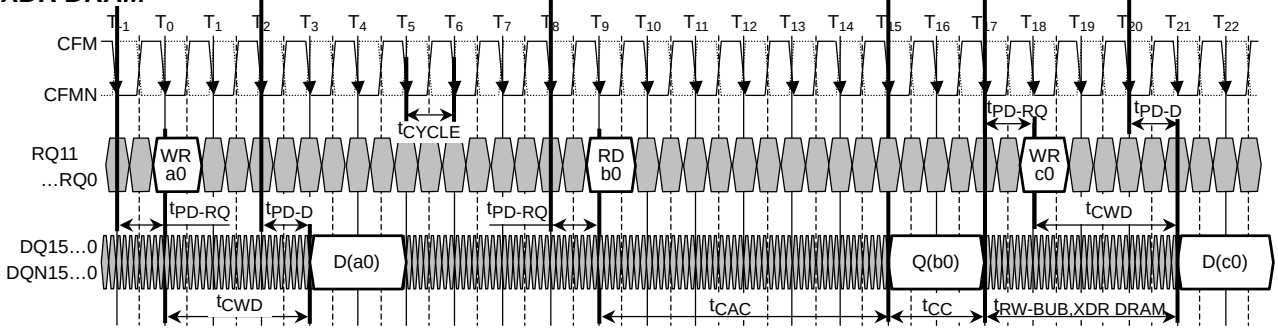
Write-Read-Write at XDR DRAM

(portions of top and bottom timing diagrams of Figure 12 merged)

Controller



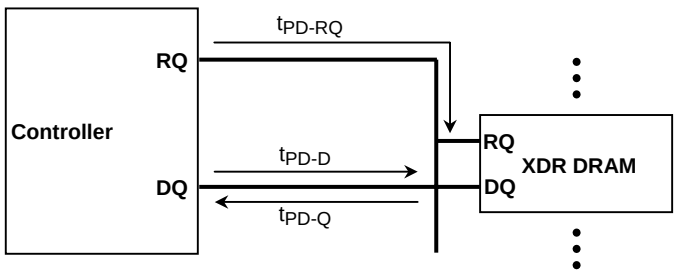
XDR DRAM



Transaction a: WR	a0 = {Ba, Ca0}
Transaction b: RD	b0 = {Bb, Cb0}
Transaction c: WR	c0 = {Bc, Cc0}

Write-Read-Write at Controller and XDR DRAM

w/t_{PD-RQ} = t_{PD-Q} = t_{PD-D} = 1*t_{CYCLE}



Register Operations

Serial Transactions

The serial interface consists of five pins. This includes RST, SCK, CMD, SDI, and SDO. SDO uses CMOS signaling levels. The other four pins use RSL signaling levels. RST, CMD, SDI, and SDO use a timing window, which surrounds the falling edge of SCK). The RST pin is used for initialization.

Figure 14 and Figure 15 show examples of a serial write transaction and a serial read transaction. Each transaction starts on cycle S_4 and requires 32 SCK edges. The next serial transaction can begin on cycle S_{36}. SCK does not need to be asserted if there is no transaction.

Serial Write Transaction

The serial device write transaction in Figure 14 begins with the Start [3:0] field. This consists of bits “1100” on the CMD pin. This indicates to the XDR DRAM that the remaining 28 bits constitute a serial transaction.

The next two bits are the SCMD [1:0] field. This field contains the serial command, the bits 00 in the case of a serial device write transaction.

The next eight bits are “00” and the SID [5:0] field. This field contains the serial identification of the device being accessed.

The next eight bits are the SADR [7:0] field. This field contains the serial address of the control register being accessed.

A single bit “0” follows next. This bit allows one cycle for the access time to the control register.

The next eight bits on the CMD pin is the SWD [7:0] field. This is the write data that is placed into the selected control register.

A final bit “0” is driven on the CMD pin to finish the serial write transaction.

A serial broadcast write is identical except that the contents of the SID [5:0] field in the transaction is ignored and all devices perform the register write. The SDI and SDO pins are not used during either serial write transaction.

Serial Read Transaction

The serial device read transaction in Figure 15 begins with the Start [3:0] field. This consists of bits “1100” on the CMD pin. This indicates that the remaining 28 bits constitute a serial transaction.

The next two bits are the SCMD [1:0] field. This field contains the serial command, and the bits “10” in the case of a serial device read transaction.

The next eight bits are “00” and the SID [5:0] field. This field contains the serial identification of the device being accessed.

The next eight bits are the SADR [7:0] field and contain the serial address of the control register being accessed.

A single bit “0” follows next. This bit allows one cycle for the access time to the control register and time to turn on the SDO output driver.

The next eight bits on the CMD pin are the sequence “00000000”. At the same time, the eight bits on the SDO pin are the SRD [7:0] field. This is the read data that is accessed from the selected control register. Note the output timing convention here: bit SRD [7] is driven from a time $t_{Q,\,SI,\,MAX}$ after edge S_{26} to a time $t_{Q,\,SI,\,MIN}$ after edge S_{27}. The bit is sampled in the controller by the edge S_{27}.

A final bit “0” is driven on the CMD pin to finish the serial read transaction.

A serial forced read is identical except that the contents of the SID [5:0] field in the transaction is ignored and all devices perform the register read. This is used for device testing.

Figure 16 shows the response of a DRAM to a serial device read transaction when its internal SID [5:0] register field doesn’t match the SID [5:0] field of the transaction. Instead of driving read data from an internal register for cycle edges S_{27} through S_{34} on the SDO output pin, it passes the input data from the SDI input pin to the SDO output pin during this same period.

Table 8. SCMD Field Encoding Summary

SCMD [1:0]	Command	DESCRIPTION
00	SDW	Serial device write-one device is written, the one whose SID[5:0] register matches the SID [5:0] filed of the transaction.
01	SBW	Serial broadcast write – all devices are written, regardless of the contents of the SID [5:0] register and the SID [5:0] transaction field.
10	SDR	Serial device read – one device is read, the one whose SID[5:0] register matches the SID [5:0] field of the transaction.
11	SFR	Serial forced read – all devices are read, regardless of the contents of the SID [5:0] register and the SID [5:0] transaction field.

Figure 14. Serial Write Transaction

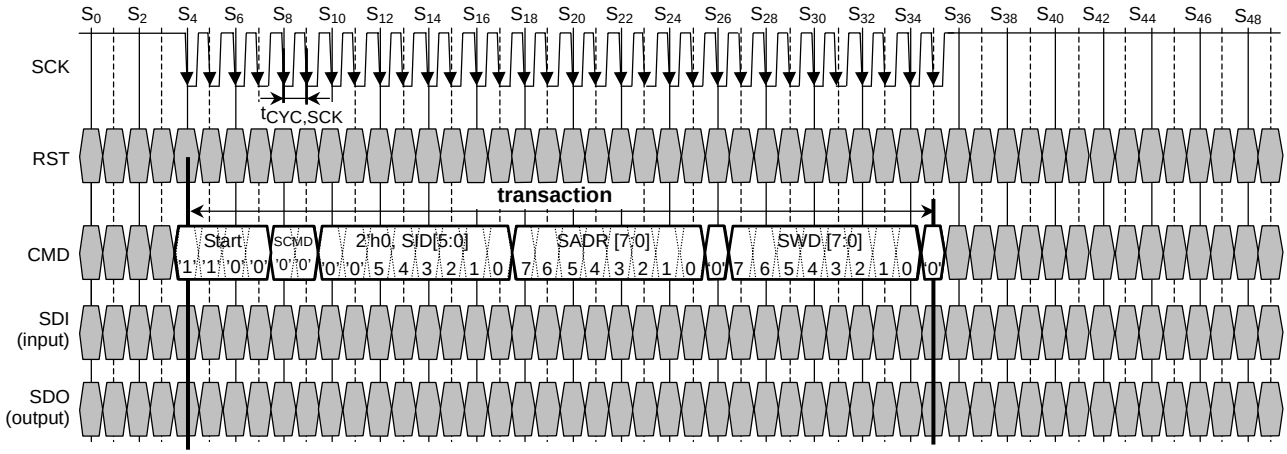


Figure 15. Serial Read Transaction – Selected DRAM

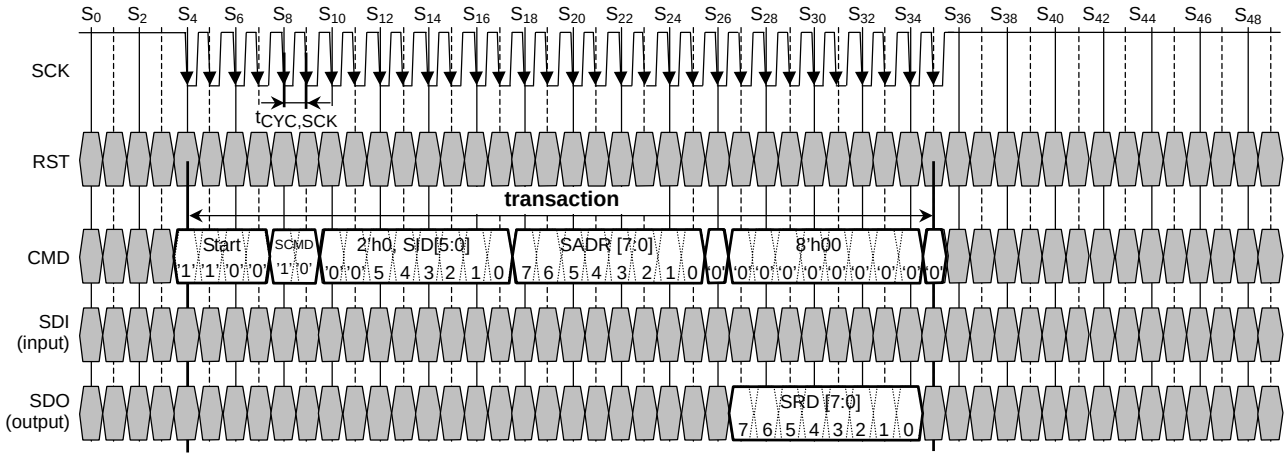
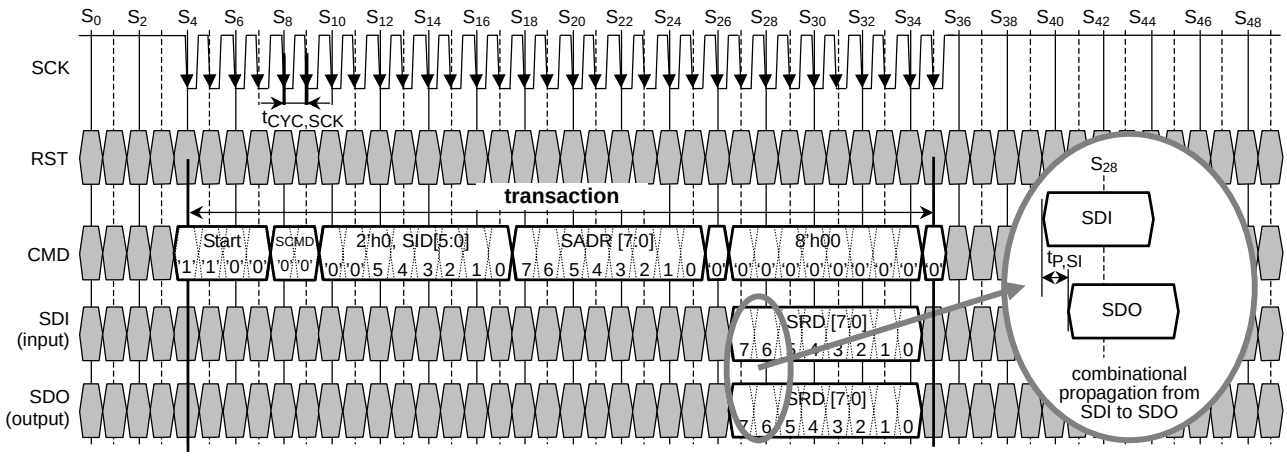


Figure 16. Serial Read Transaction – Non-Selected DRAM



Register Summary

Figure 17 through Figure 40 show the control registers in the memory component. The control registers are responsible for configuring the component's operating mode, for managing power state transitions, for managing refresh, and for managing calibration operations.

A control register may contain up to eight bits. Each figure shows defined bits in white and reserved bits in gray. Reserved bits must be written as 0 and must be ignored when read. Write-only fields must be ignored when read

Each figure displays the following register information:

1. Register name
2. Register mnemonic
3. Register address (SADR [7:0] value needed to access it)
4. Read-only, write-only or read-write
5. Initialization state
6. Description of each defined register field

Figure 17 shows the Serial Identification register. This register contains the SID [5:0] (serial identification field). This field contains the serial identification value for the device. The value is compared to the SID [5:0] field of a serial transaction to determine if the serial transaction is directed to this device. The serial identification value is set during the initialization sequence.

Figure 18 shows the Configuration Register. It contains three fields. The first is the WIDTH field. This field allows the number of DQ/DQN pins used for memory read and write accesses to be adjusted. The SLE field enables data to be written into the memory through the serial interface using the WDSL register.

Figure 19 shows the Power Management Register. It contains two fields. The first is the PX field. When this field is written with a 1, the memory component transitions from Power Down to active state. It is usually unnecessary to write a 0 into this field; this is done automatically by the PDN command in a COLX packet. The PST field indicates the current power state of the memory component.

Figure 20 shows the Write Data Serial Load Register. It permits data to be written into memory via the Serial Interface.

Figure 23 shows the Refresh Bank Control Register. It contains two fields: BANK and MBR. The BANK field is read-write and contains the bank address used by self-refresh during the powerdown state. The MBR field controls how many banks are refreshed during each refresh operation. Figure 24, Figure 25 and Figure 26 show different fields of the Refresh Row Register (high, middle, and low). This read-write field contains the row address used by self- and auto-refresh. See “Refresh Transactions” on page 40 for more details.

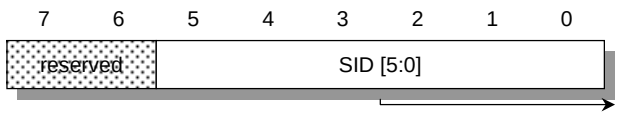
Figure 27 and Figure 28 show the Current Calibration 0 and 1 registers. They contain the CCVALUE0 and CCVALUE1 fields, respectively. These are read-write fields which control the amount of IOL current driven by the DQ and DQN pins during a read transaction. The Current Calibration 0 Register controls the even-numbered DQ and DQN pins, and the Current Calibration 1 controls the odd-numbered DQ and DQN pins.

Figure 29 and Figure 30 shows the Impedance Calibration 0 and 1 registers. They contain the ZCVALUE0 and ZCVALUE1 fields, respectively. These are read-write fields that control the impedance of the on-chip termination components in the DQ and DQN pins. The Impedance Calibration 0 Register controls the even-numbered DQ and DQN pins, and the Impedance Calibration 1 controls the odd-numbered DQ and DQN pins.

Figure 33 through Figure 39 shows the test registers. This includes the TEST, DLL, PLL0, PLL1, IFT, DA, and PART registers. These are used during device testing. They are not to be read or written during normal operation.

Figure 40 shows the DLY register. This is used to set the value of t_{CAC} and t_{CWD} used by the component. See “Timing Parameters” on page 60.

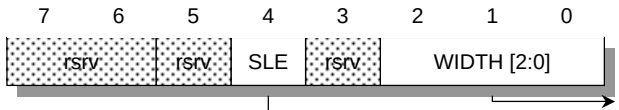
Figure 17. Serial Identification (SID) Register



Serial Identification Register — Read-only register
SADR [7:0]: 00000001_2 — SID [7:0] resets to 00000000_2

SID [5:0] – Serial Identification field.
This field contains the serial identification value for the device. The value is compared to the SID [5:0] field of a serial transaction to determine if the serial transaction is directed to this device. The serial identification value is set during the initialization sequence.

Figure 18. Configuration (CFG) Register

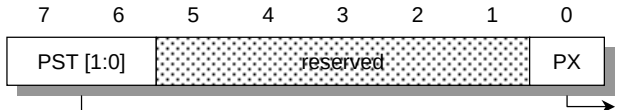


Configuration Register — Read/write register
SADR [7:0]: 00000010_2 — CFG [7:0] resets to 00000100_2

WIDTH [2:0] – Device interface width field.
- 000_2 – Reserved
- 001_2 – Reserved
- 010_2 – ×4 device width
- 011_2 – ×8 device width
- 100_2 – ×16 device width
- 101_2, 110_2, 111_2 – Reserved

SLE - Serial Load enable field.
- 0_2 – WDSL-path-to-memory disabled
- 1_2 – WDSL-path-to-memory enabled

Figure 19. Power Management (PM) Register

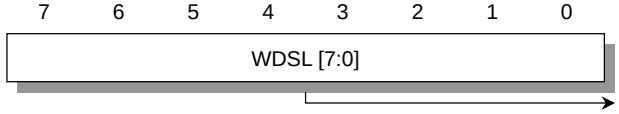


Power Management Register — Read/write register
SADR [7:0]: 00000011_2 — PM [7:0] resets to 00000000_2

PX – Power down exit field. (write-one-only read = zero)
- 0_2 – Power down entry do not write zero – use PDN command.
- 1_2 – Power down exit – write one to exit

PST [1:0] – Power state field (read-only).
- 00_2 – Power down (with self-refresh).
- 01_2 – Active/active-idle
- 10_2 – Reserved
- 11_2 – Reserved

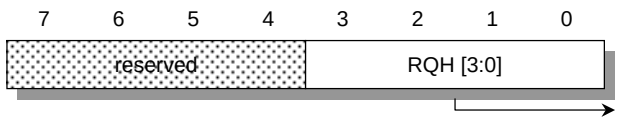
Figure 20. Write Data Serial Load (WDSL) Control Register



Write Data Serial Load Control Register — Read/write register
SADR [7:0]: 00000100_2 — WDSL [7:0] resets to 00000000_2

WDSL [7:0] – Writing to this register places eight bits of data into the serial-to-parallel conversion logic (the “Demux” block of Figure 2). Writing to this register “2x16” times accumulates a full “t_{CC}” worth of write data. A subsequent WR command (with SLE = 1 in CFG register in Figure 18) will write this data (rather than DQ data) to the sense amps of a memory bank. The shifting order of the write data is shown in Table 10.

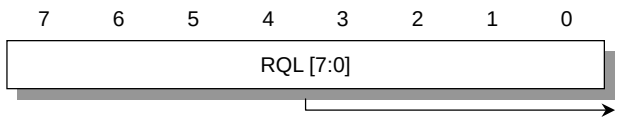
Figure 21. RQ Scan High (RQH) Register



RQ Scan High Register — Read/write register
SADR [7:0]: 00000110_2 — RQH [7:0] resets to 00000000_2

RQH [3:0] – Latched value of RQ [11:8] in RQ wire test mode.

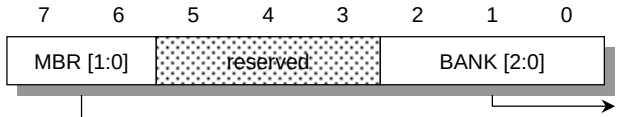
Figure 22. RQ Scan Low (RQL) Register



RQ Scan Low Register — Read/write register
SADR [7:0]: 00000111_2 — RQL [7:0] resets to 00000000_2

RQH [7:0] – Latched value of RQ [7:0] in RQ wire test mode.

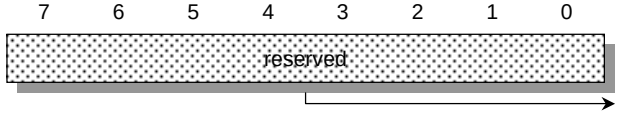
Figure 23. Refresh Bank (REFB) Control Register



Refresh Bank Control Register — Read/write register
SADR [7:0]: 00001000_2 — REFB [7:0] resets to 00000000_2

BANK [2:0] – Refresh bank field.
This field returns the bank address for the next self-refresh operation when in powerdown power state.
MBR [1:0] – Multiple-bank and multi-row refresh control field.
00_2 – Single-bank refresh 10_2 – Reserved
01_2 – Reserved 11_2 – Reserved

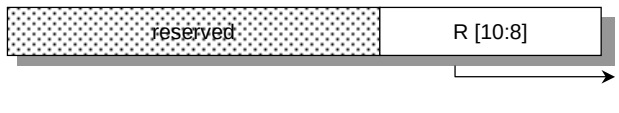
Figure 24. Refresh High (REFH) Row Register



Refresh High Row Register — Read/write register
SADR [7:0]: 00001001_2 — REFH [7:0] resets to 00000000_2

Reserved – Refresh row field.
This field contains the high-order bits of the row address that will be refreshed during the next refresh interval. This row address will be incremented after a REFI command for auto-refresh, or when the BANK [2:0] field for the REFB register equals the maximum bank address for self-refresh.

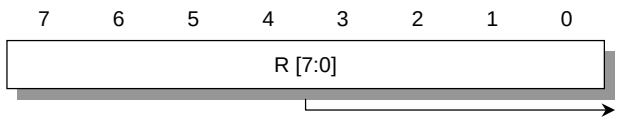
Figure 25. Refresh Middle (REFM) Row Register



Refresh Middle Row Register — Read/write register
SADR [7:0]: 00001010_2 — REFM [7:0] resets to 00000000_2

R [10:8] – Refresh row field.
This field contains the middle-order bits of the row address that will be refreshed during the next refresh interval. This row address will be incremented after a REFI command for auto-refresh, or when the BANK [2:0] field for the REFB register equals the maximum bank address for self-refresh.

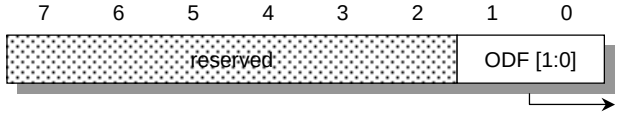
Figure 26. Refresh Low (REFL) Row Register



Refresh Low Row Register — Read/write register
SADR [7:0]: 00001011_2 — REFL [7:0] resets to 00000000_2

R [7:0] – Refresh row field.
This field contains the low-order bits of the row address that will be refreshed during the next refresh interval. This row address will be incremented after a REFI command for auto-refresh, or when the BANK [2:0] field for the REFB register equals the maximum bank address for self-refresh.

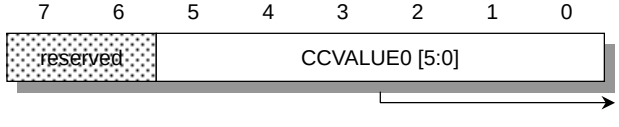
Figure 27. IO Configuration (IOCFG) Register



Current Calibration 0 Register — Read/write register
SADR [7:0]: 00010000_2 — CC0 [7:0] resets to 00001111_2

ODF [1:0] – Overdrive Function field.
00 – Nominal VOSW,DQ range
01 – reserved
10 – reserved
11 – reserved

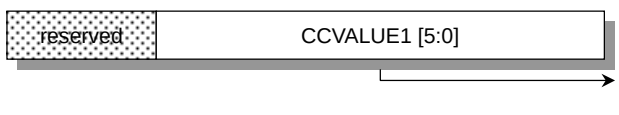
Figure 28. Current Calibration 0 (CC0) Register



Current Calibration 0 Register — Read/write register
SADR [7:0]: 00010000_2 — CC0 [7:0] resets to 00001111_2

CCVALUE0 [5:0] – Current calibration value field.
This field controls the amount of current drive for the even-numbered DQ and DQN pins.

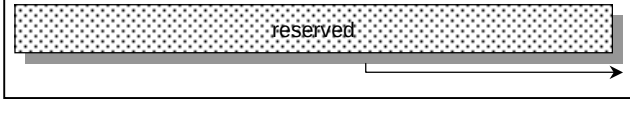
Figure 29. Current Calibration 1 (CC1) Register



Current Calibration 1 Register — Read/write register
SADR [7:0]: 00010001_2 — CC1 [7:0] resets to 00001111_2

CCVALUE1 [5:0] – Current calibration value field.
This field controls the amount of current drive for the odd-numbered DQ and DQN pins.

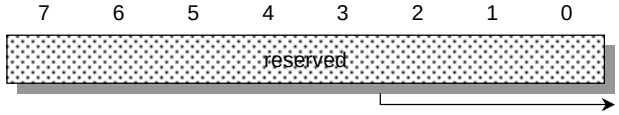
Figure 30. Impedance Calibration 0 (ZC0) Register



Impedance Calibration 0 Register — Read/write register
SADR [7:0]: 00010010_2 — ZC0 [7:0] resets to 00000000_2

reserved

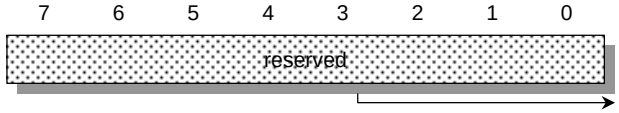
Figure 31. Impedance Calibration 1 (ZC1) Register



Impedance Calibration 1 Register — Read/write register
SADR [7:0]: 00010011_2 — ZC1 [7:0] resets to 00000000_2

resereved.

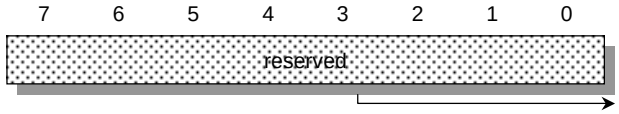
Figure 32. Current Fuse Setting 0 (FZC0) Register



Current Fuse Setting Register — Read-only register
SADR [7:0]: 00010100_2 — FZC0 [7:0] resets to vvvvvvvv (vendor-dependent reset value)

reserved

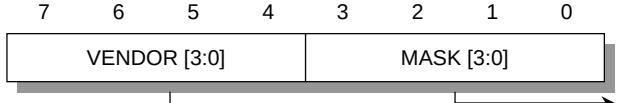
Figure 33. Current Fuse Setting 1 (FZC1) Register



Current Fuse Setting Register — Read-only register
SADR [7:0]: 00010101_2 — FZC1 [7:0] resets to vvvvvvvv (vendor-dependent reset value)

reserved

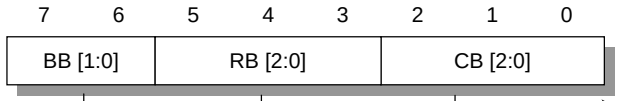
Figure 34. Read Only Memory 0 (ROM0) Register



Read Only Memory 0 Register — Read-only register
SADR [7:0]: 00010110_2 — ROM0 [7:0] resets to vvvvmmmm

MASK [3:0] – Version number of mask (0001_2 is first version).
VENDOR [3:0] – Vendor number for component:
0000 – Reserved
0001 – Toshiba
0010 – Elpida
0011 – SEC
0100 – 1111 – Reserved

Figure 35. Read Only Memory 1 (ROM1) Register



Read Only Memory 1 Register — Read-only register
SADR [7:0]: 00010111_2 — ROM0 [7:0] resets to bbrrrccc

CB [2:0] – Column address bits: #bits = 6 + CB [2:0]
RB [2:0] – Row address bits: #bits = 10 + RB [2:0]
BB [2:0] – Bank address bits: #bits = 2 + BB [2:0]
There three fields indicate how many column, row, and bank address bits are present. An offset of {6,10,2} is added to the field value to give the number of address bits.

Figure 36. Test Register

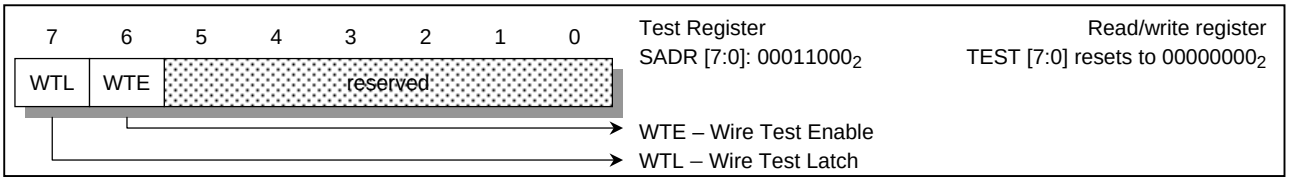


Figure 37. DLL Register

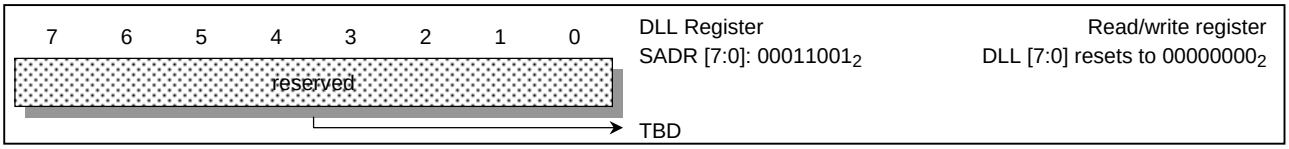


Figure 38. PLL0 Register

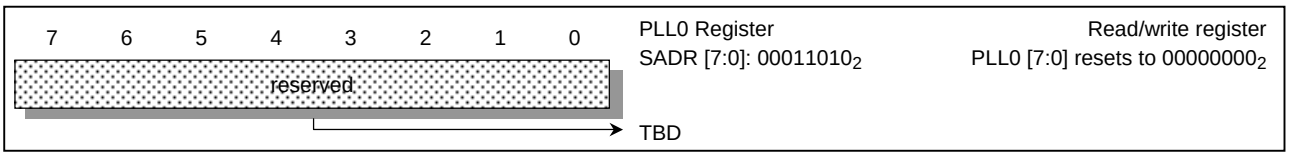


Figure 39. PLL1 Register

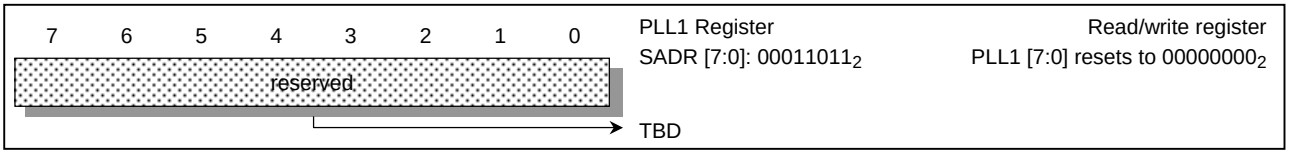


Figure 40. IFT Register

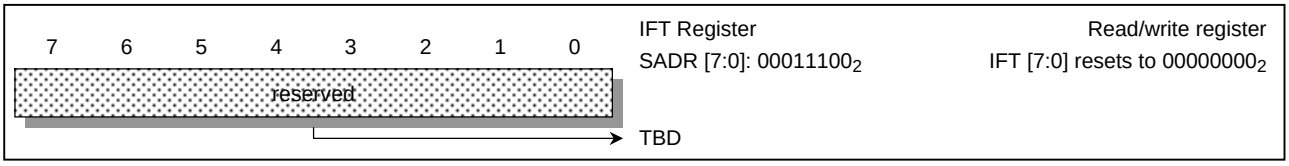


Figure 41. DA Register

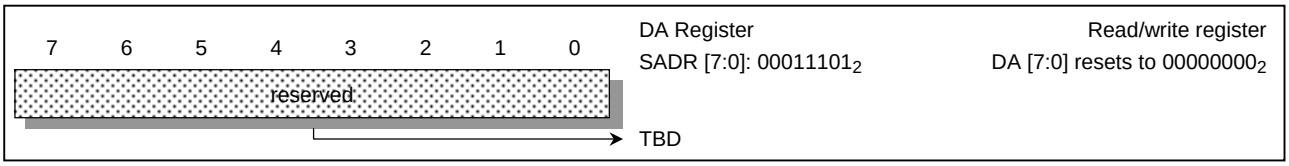
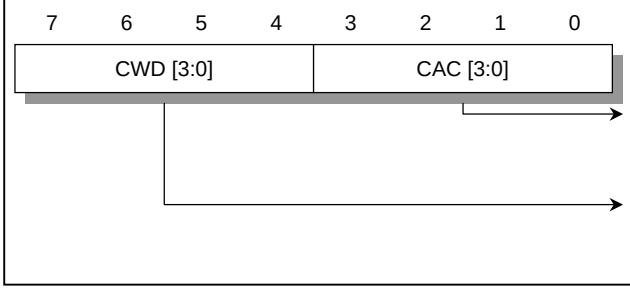


Figure 42. Delay (DLY) Control Register



DLY Register — Read/write register

SADR [7:0]: 00011111_2 — DLY [7:0] resets to 01000110_2

CAC [3:0] - Programmed value of t_{CAC} timing parameter:

0110_2 – $t_{CAC} = 6*t_{CYCLE}$ 1000_2 – $t_{CAC} = 8*t_{CYCLE}$

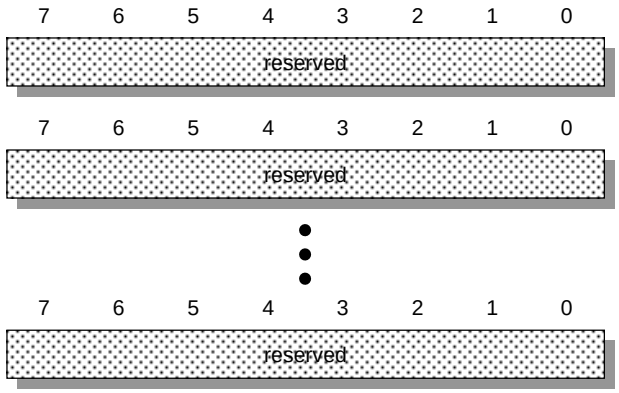
0111_2 – $t_{CAC} = 7*t_{CYCLE}$ others – Reserved

CWD [3:0] – Programmed value of t_{CWD} timing parameter:

0011_2 – $t_{CWD} = 3*t_{CYCLE}$

0100_2 – $t_{CWD} = 4*t_{CYCLE}$ others – Reserved

Figure 43. Partner-Definable (PART0-PARTF) Registers



PART0 Register — Read/write register

SADR [7:0]: 10000000_2 — PART0 [7:0] resets to 00000000_2

PART1 Register — Read/write register

SADR [7:0]: 10000001_2 — PART1 [7:0] resets to 00000000_2

PARTF Register — Read/write register

SADR [7:0]: 10001111_2 — PARTF [7:0] resets to 00000000_2

Note: The partner-definable registers should not be written or read; doing so will produce undefined results.

Maintenance Operations

Refresh Transactions

Figure 44 contains two timing diagrams showing examples of refresh transactions. The top timing diagram shows a single refresh operation. Bank Ba is assumed to be closed (in a precharged state) when a REFA command is received in a ROWP packet on clock edge T_0. The REFA command causes the row addressed by the REFr register (REFH/REFM/REFL) to be opened (sensed) and placed in the sense amp array for the bank.

Note that the REFA and REFI commands are similar to the ACT command functionally; both specify a bank address and delay value, and both cause the selected bank to open (to become sensed.) The difference is that the ACT command is accompanied by a row address in the ROWA packet, while the REFA and REFI commands use a row address in the REFr register (REFH/REFM/REFL).

After a time t_{RAS}, a ROWP packet with REFP command to bank Ba is presented. This causes the bank to be closed (precharged), leaving the bank in the same state as when the refresh transaction began.

Note that the REFP command is equivalent to the PRE command functionally; both specify a bank address and delay value, and both cause the selected bank to close (to become precharged).

After a time t_{RP}, another ROWP packet with REFA command to bank Bb is presented (banks Ba and Bb are the same in this example). This starts a second refresh cycle. Each refresh transaction requires a total time $t_{RC} = t_{RAS} + t_{RP}$, but refresh transactions to different banks may be interleaved like normal read and write transactions.

Each row of each bank must be refreshed once in every t_{REF} interval. This is shown with the fourth ROWP packet with a REFA command in the top timing diagram.

Interleaved Refresh Transactions

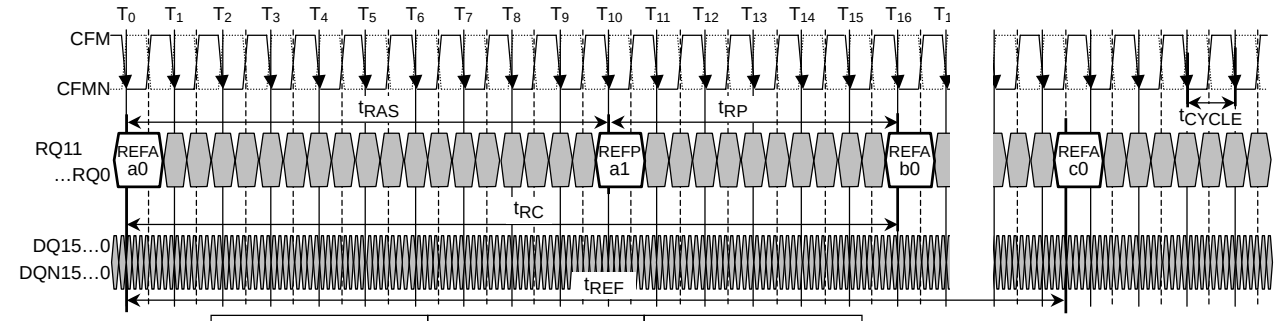
The lower timing diagram in Figure 44 represents one way a memory controller might handle refresh maintenance in a real system.

A series of eight ROWP packets with REFA commands (except for the last which is a REFI command) are presented starting at edge T_0. The packets are spaced with intervals of t_{RR}. Each REFA or REFI command is addressed to a different bank (Ba through Bh) but uses the same row address from the REFr (REFH/REFM/REFL) register. The eighth REFI command uses this address and then increments it so the next set of eight REFA/REFI commands will refresh the next set of rows in each bank.

A series of eight ROWP packets with REFP commands are presented effectively at edge T_{10} (a time t_{RAS} after the first ROWP packet with a REFA command). The packets are spaced with intervals of t_{PP}. Like the REFA/REFI commands, each REFP command is addressed to a different bank (Ba through Bh).

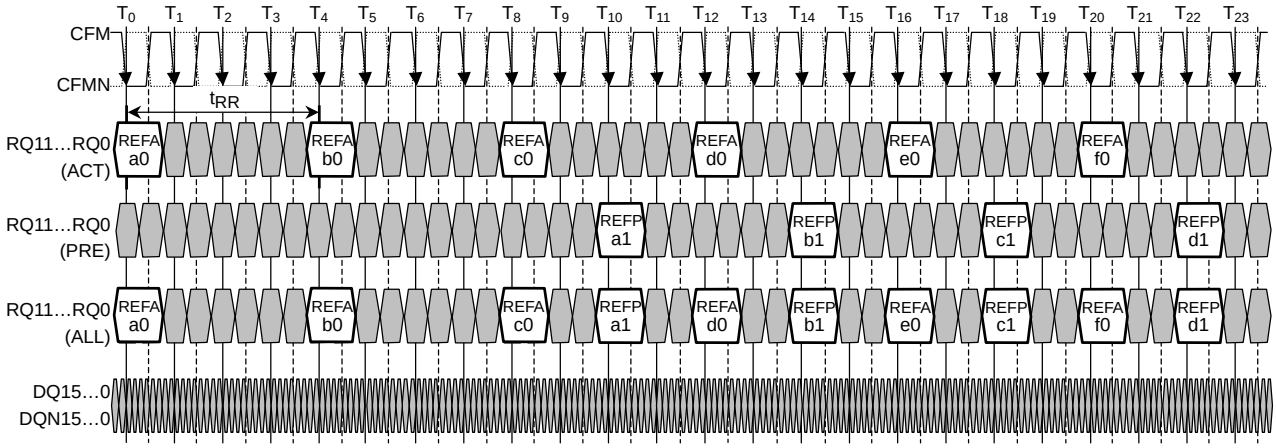
This burst of eight refresh transactions fully utilizes the memory component. However, other read and writes transactions may be interleaved with the refresh transactions before and after the burst to prevent any loss of bus efficiency. In other words, a ROWA packet with ACT command for a read or write could have been presented at edge T_{-4} (a time t_{RR} before the first refresh transaction starts at edge T_0). Also, a ROWA packet with ACT command for a read or write could have been presented at edge T_{36} (a time t_{RR} after the last refresh transaction starts at edge T_{32}). In both cases, the other request packets for the interleaved read or write accesses (the precharge commands and the read or write commands) could be slotted in among the request packets for the refresh transactions.

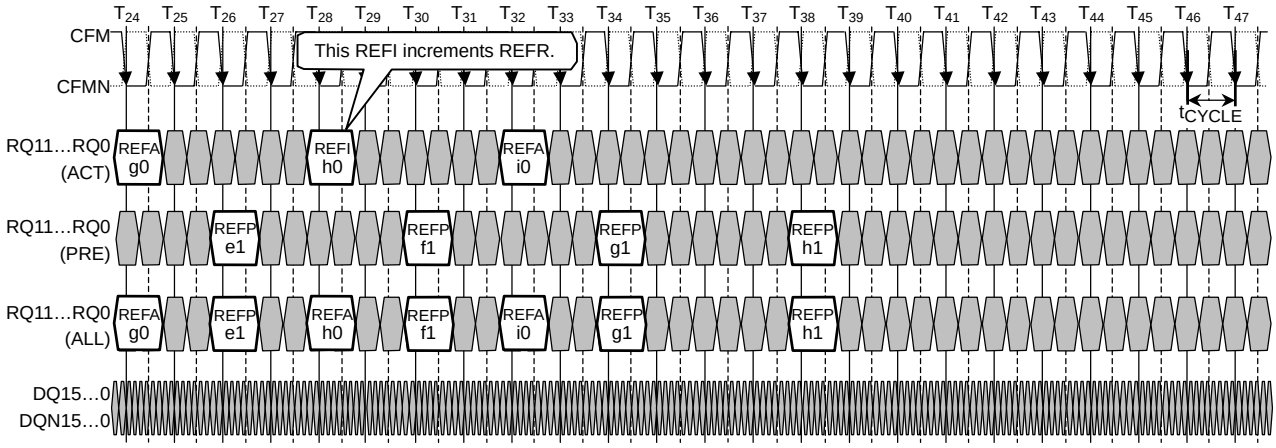
Figure 44. Refresh Transactions



	Transaction a: REF	a0 = {Ba, REFR}	a1 = {Ba }
Bb = Ba →	Transaction b: REF	b0 = {Bb, REFR}	b1 = {Bb }
Bc/Rc = Ba/Ra →	Transaction c: REF	c0 = {Bc, REFR}	c1 = {Bc }

Refresh Transaction





Ba,Bb,Bc,Bd,Be, Bf,Bg and Bh are different banks.

	Transaction a: REF	a0 = {Ba, REFR}	a1 = {Ba}
	Transaction b: REF	b0 = {Bb, REFR}	b1 = {Bb}
	Transaction c: REF	c0 = {Bc, REFR}	c1 = {Bc}
	Transaction d: REF	d0 = {Bd, REFR}	d1 = {Bd}
	Transaction e: REF	e0 = {Be, REFR}	e1 = {Be}
	Transaction f: REF	f0 = {Bc, REFR}	f1 = {Bf}
	Transaction g: REF	g0 = {Bg, REFR}	g1 = {Bg}
	Transaction h: REF	h0 = {Bh, REFR}	h1 = {Bh}
Bi = Ba →	Transaction i: REF	i0 = {Bi, REFR + 1}	i1 = {Bi}

Interleaved Refresh Example

Calibration Transactions

Figure 45 shows the calibration transaction diagrams for the XDR DRAM device. There is one calibration operation supported: calibration of the output current level I_{OL} each DQi and DQNi pin.

The output current calibration sequence is shown in the upper diagram. It begins when a period of $t_{CMD-CALC}$ is observed after the last RQ packet (with command "CMD a" in this example). No request packets should be issued in this period.

A COLX packet with a "CALC b" command is then issued to start the current calibration sequence. A period of t_{CALCE} is observed after this packet. No request packets should be issued during this period.

A COLX packet with a "CALE c" command is then issued to end the current calibration sequence. A period of $t_{CALE-CMD}$ is observed after this packet. No request packets should be issued during this period. The first request packet may then be issued (with command "CMD d" in this example).

A second current calibration sequence must be started within an interval of t_{CALC}. In this example, the next COLX packet with a "CALC e" command starts a subsequent sequence.

Note that the labels for the CFM clock edges (of the form Ti) are not to scale, and are used to identify events in the diagram.

The dynamic termination calibration sequence is shown in the lower diagram. Note that this memory component does not use this sequence; termination calibration is performed during the manufacturing process. However, the termination sequence shown will be issued by the controller for those memory components which do use a periodic calibration mechanism.

It begins when a period of $t_{CMD-CALZC}$ is observed after the packet at edge T_0 (with command CMDa in the example). No request packets should be issued during this period.

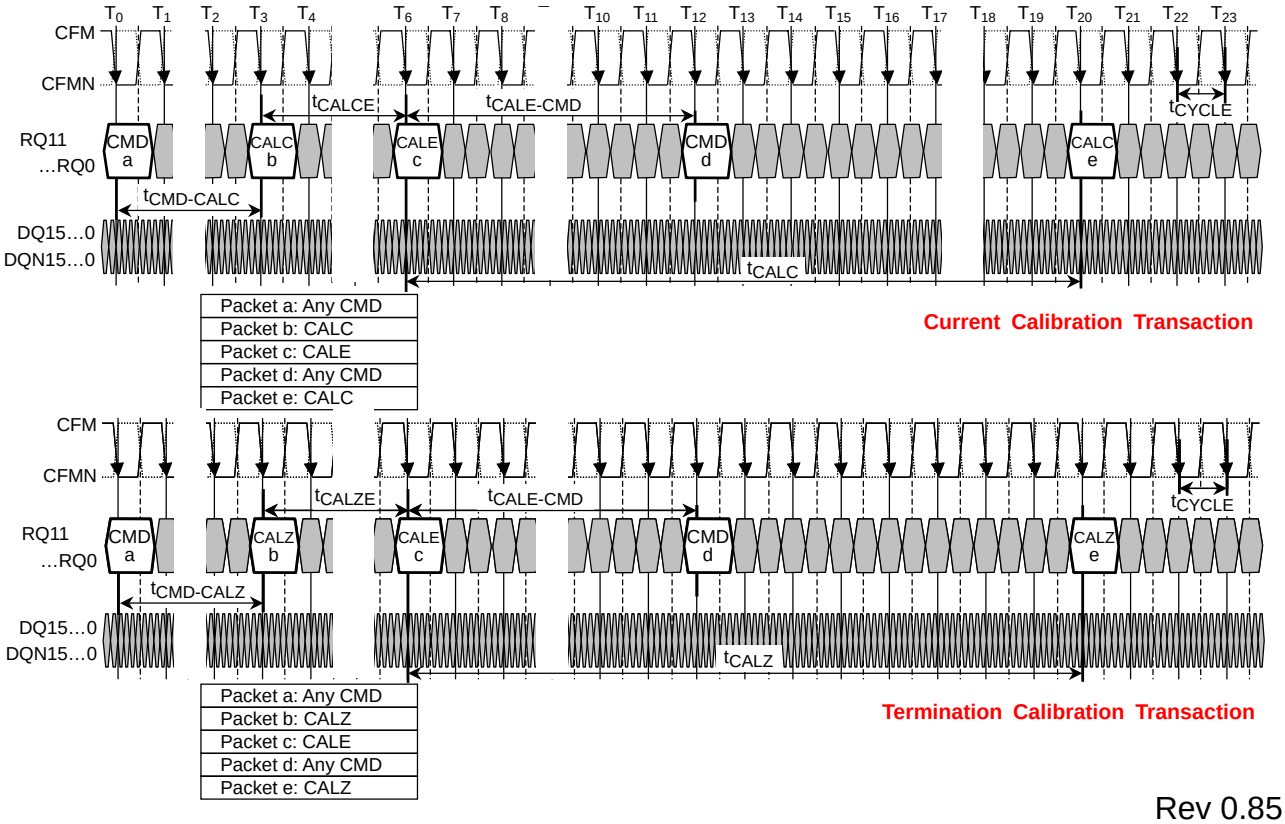
A COLX packet with a CALZ command is then issued at edge T_3 to start the current calibration sequence. A second period of t_{CALZE} is observed after this packet. No request packets should be issued during this period.

A COLX packet with a CALE command is then issued at edge T_6 to end the current calibration sequence. A third period of $t_{CALE-CMD}$ is observed after this packet. No request packets should be issued during this period. The first request packet may be issued at edge T_{12} (with command CMDd in this example).

A second current calibration sequence must be started within an interval of t_{CALZ}. In this example, the next COLX packet with a CALZ command occurs at edge T_{20}.

Note that the labels for the CFM clock edges (of the form Ti) are not to scale, and are used to identify events in the diagrams.

Figure 45. Calibration Transactions



Power State Management

Figure 46 shows power state transition diagrams for the XDR DRAM device. There are two power states in the XDR DRAM: Powerdown and Active. Powerdown state is to be used in applications in which it is necessary to shut down the CFM/CFMN clock signals. In this state, the contents of the storage cells of the XDR DRAM will be retained by an internal state machine which performs periodic refresh operations using the REFB and REFr control registers.

The upper diagram shows the sequence needed for Powerdown entry. Prior to starting the sequence, all banks of the XDR DRAM must be precharged so they are left in a closed state. Also, all 2^3 banks must be refreshed using the current value of the REFr registers, ***and the REFr registers must NOT be incremented with the REFI command at the end of this special set of refresh transactions***. This ensures that no matter what value has been left in the REFB register, no row of any bank will be skipped when automatic refresh is first started in Powerdown. There may be some banks at the current row value in the REFr registers that are refreshed twice during the Powerdown entry process.

After the last request packet (with the command CMDa in the upper diagram of the figure), an interval of $t_{CMD-PDN}$ is observed. No request packets should be issued during this period.

A COLX packet with the PDN command is issued after this interval, causing the XDR DRAM to enter Powerdown state after an interval of $t_{PDN-ENTRY}$ has elapsed (this is the parameter that should be used for calculating the power dissipation of the XDR DRAM). The CFM/CFMN clock signals may be removed a time $t_{PDN-CFM}$ after the COLX packet with the PDN command. Also, the termination voltage supply may be removed (set to the ground reference) from the VTERM pins a time $t_{PDN-CFM}$ after the COLX packet with the PDN command. The voltage on the DQ/DQN pins will follow the voltage on the VTERM pins during Power Down entry.

When the XDR DRAM is in Powerdown, an internal frequency source and state machine will automatically generate internal refresh transactions. It will cycle through all 2^3 state combinations of the REFB register. When the largest value is reached and the REFB value wraps around, the REFr register is incremented to the next value. The REFB and REFr values select which bank and which row are refreshed during the next automatic refresh transaction.

The lower diagram shows the sequence needed for Powerdown exit. The sequence is started with a serial broadcast write (SBW command) tranasaction using the serial bus of the XDR DRAM. This transaction writes the value “00000001” to the Power Management (PM) register (SADR = “00000011”) of all XDR DRAMs connected to the serial bus. This sets the PX bit of the PM register, causing the XDR DRAMs to return to Active power state.

The CFM/CFMN clock signals must be stable a time $t_{CFM-PDN}$ before the end of the SBW transaction. Also, the termination voltage supply must be restored to its normal operating point ($V_{TERM, DRSL}$) on the VTERM pins a time tCFM-PDN before the end of the SBW transaction. The voltage on the DQ/DQN pins will follow the voltage on the VTERM pins during Power Down exit.

The XDR DRAM will enter Active state after an interval of $t_{PDN-EXIT}$ has elapsed from the end of the SBW transaction (this is the parameter that should be used for calculating the power dissipation of the XDR DRAM).

The first request packet may be issued after an interval of $t_{PDN–CMD}$ has elapsed from the end of the SBW transaction, ***and must contain a “REFA” command in a ROWP packet***. In this example, this packet is denoted with the command “REFA 1”. No other request packets should be issued during this $t_{PDN-CMD}$ interval.

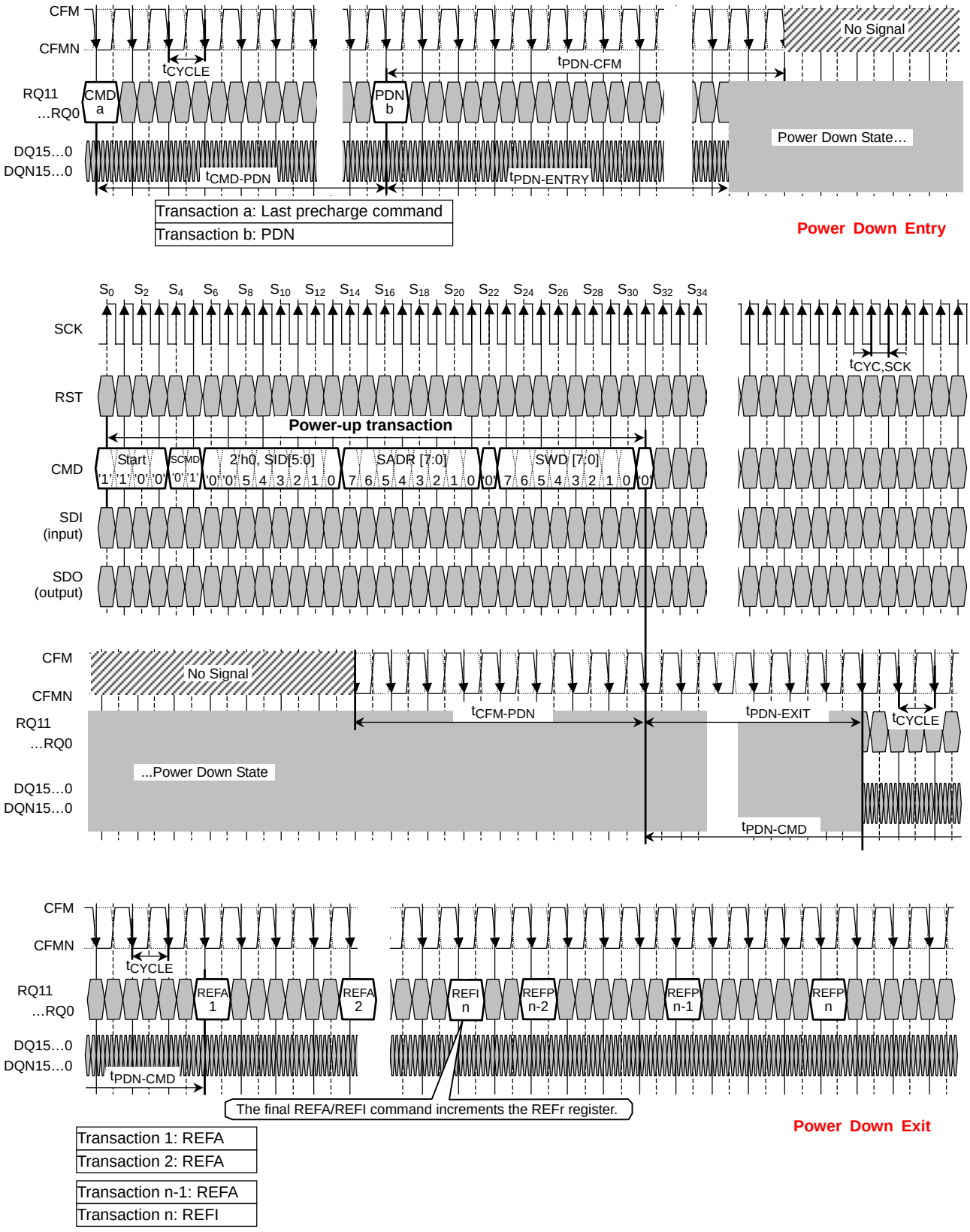
All “n” banks (in the example, $n = 2^3$) must be refreshed using the current value of the REFr registers. The “nth” refresh transaction will use a “REFI” command to increment the REFr register (instead of a “REFR” command). This ensures that no matter what value has been left in the REFB register, no row of any bank will be skipped when normal refresh is restarted in Active state. There may be some banks at the current row value in the REFr registers that are refreshed twice during the Powerdown exit process.

Note that during the Powerdown state an internal time source keeps the device refreshed. However, during the $t_{PDN-CMD}$ interval, no internal refresh operations are performed. As a result, an additional burst of refresh transactions must be issued after the burst of “n” transactions described above. This second burst consists of “m” refresh transactions:

$m = \text{ceiling}\ [2^3 * 2^{11} * t_{PDN-CMD}/t_{REF}]$

Where “2^{11}” is the number of rows per bank, and “2^3” is the number of banks. Every “nth” refresh transaction (where $n = 2^3$) will use a “REFI” command (to increment the REFr register) instead of a “REFA” command.

Figure 46. Power State Management



Initialization

Figure 47 shows the topology of the serial interface signals of a XDR DRAM system. The three signals RST, CMD, and SCK are transmitted by the controller and are received by each XDR DRAM device along the bus. The signals are terminated to the VTERM supply through termination components at the end farthest from the controller. The SDI input of the XDR DRAM device furthest from the controller is also terminated to VTERM. The SDO output of each XDR DRAM device is transmitted to the SDI input of the next XDR DRAM device (in the direction of the controller). This SDO/SDI daisy-chain topology continues to the controller, where it ends at the SRD input of the controller. All the serial interface signals are low-true. All the signals use RSL signaling circuits, except for the SDO output which uses CMOS signaling circuits.

Figure 47. Serial Interface Systems Topology

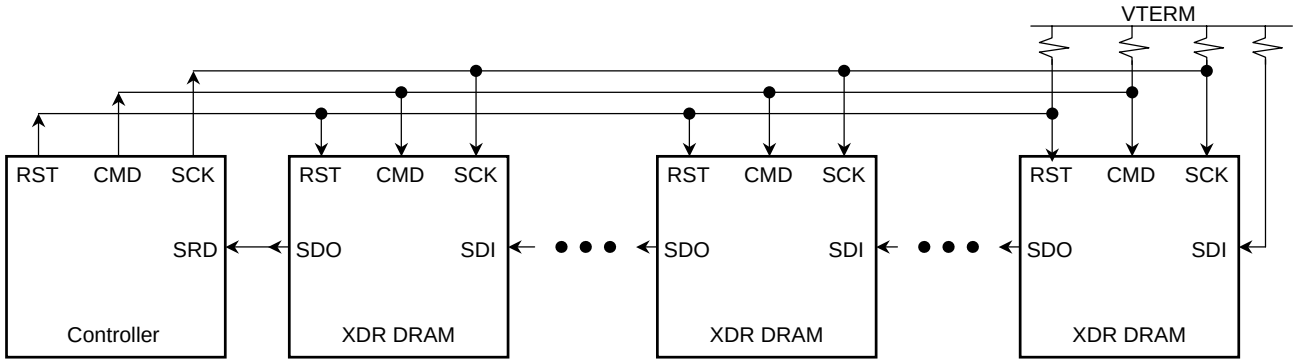
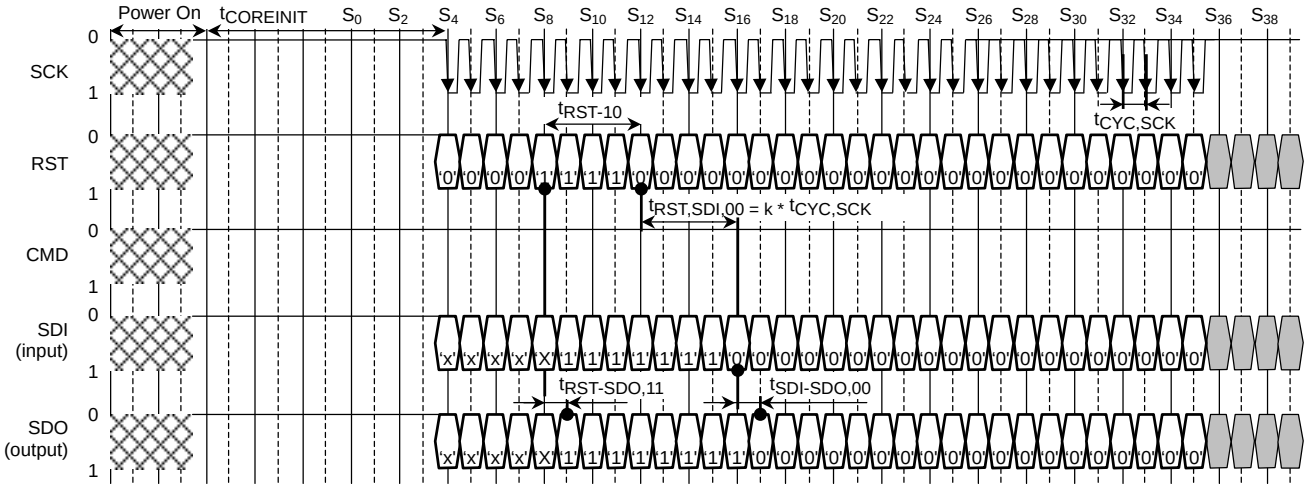


Figure 48 shows the initialization timing of the serial interface for the XDR DRAM [k] device in the system shown above. Prior to initialization, the RST is held at zero. The CMD input is not used here, and should also be held at zero. Note that the inputs are all sampled by the negative edge of the SCK clock input. The SDI input for the XDR DRAM [0] device is zero, and is unknown for the remaining devices.

On negative SCK edge S_8 the RST input is sampled one. It is sampled one on the next four edges, and is sampled zero on edge S_{12} a time t_{RST-10} after it was first sampled one. The state of the control registers in the XDR DRAM device are set to their reset values after the first edge (S_8) in which RST is sampled one.

Figure 48. Initialization Timing for XDR DRAM [k] Device



The SDI inputs will be sampled one within a time $t_{RST-SDO, 11}$ after RST is first sampled one in all the XDR DRAMs except for XDR DRAM [0]. XDR DRAM [0]'s SDI input will always be sampled zero.

XDR DRAM [k] will see its RST input sampled zero at S_{12}, and will then see its SDI input sampled zero at S_{16} (after SDI had previously been sampled one). This interval (measured in $t_{CYC, SCK}$ units) will be equal to the index [k] of the XDR DRAM device along the serial interface bus. In this example, k is equal to 4.

This is because each XDR DRAM device will drive its SDO output zero around the SCK edge a time $t_{SDI-SDO,00}$ after its SDI input is sampled zero.

In other words, the XDR DRAM [0] device will see RST and SDI both sampled zero on the same edge S_{12} ($t_{RST-SDI, 00}$ will be $0 \times t_{CYC, SCK}$ units), and will drive its SDO to zero around the subsequent edge (S_{13}).

The XDR DRAM [1] device will see SDI sampled zero on edge S_{13} ($t_{RST-SDI, 00}$ will be $1 \times t_{CYC, SCK}$ units), and will drive its SDO to zero around the subsequent edge (S_{14}).

The XDR DRAM [2] device will see SDI sampled zero on edge S_{14} ($t_{RST-SDI, 00}$ will be $2 \times t_{CYC, SCK}$ units), and will drive its SDO to zero around the subsequent edge (S_{15}).

This continues until the last XDR DRAM device drives the SRD input of the controller. Each XDR DRAM device contains a state machine which measures the interval $t_{RST–SDI, 00}$ between the edges in which RST and SDI are both sampled zero, and uses this value to set the SID [5:0] field of the SID (Serial Identification) register. This value allows directed read and write transactions to be made to the individual XDR DRAM devices.

Table 9 summarizes the range of the timing parameters used for initialization by the serial interface bus.

Table 9. Initialization Timing Parameters

Symbol	Parameter	Min	Max	Unit	Figure (s)
$t_{RST, 10}$	Number of cycles between RST being sampled one and RST being sampled zero.	2	—	$t_{CYC,SCK}$	—
$t_{RST-SDO, 11}$	Number of cycles between RST being sampled one and SDO being driven to one.	1	1	$t_{CYC,SCK}$	—
$t_{RST, SDI, 00}$	Number of cycles between RST being sampled zero (after being sampled one for $t_{RST, 10, MIN}$ or more cycles) and SDI being sampled zero. This will be equal to the index [k] of the XDR DRAM device along the serial interface bus.	0	63	$t_{CYC,SCK}$	—
$t_{SDI-SDO, 00}$	Number of cycles between SDI being sampled one (after RST has been sampled one for $t_{RST, 10, MIN}$ or more cycles and is then sampled zero) and SDO being driven to zero.	1	1	$t_{CYC,SCK}$	—
$t_{RST-SCK}$	The number of SCK falling edges after the first SCK falling edge in which RST is sampled one.	20	—	$t_{CYC,SCK}$	—

XDR DRAM Initialization Overview

[1] Apply voltage to VDD, VTERM and VREF pins. VTERM and VREF coltages must be less or equal to VDD voltage at all times. Wait a time interval $t_{COREINIT}$. Power-on reset circuit in XDR DRAM places XDR DRAM into low-power state.

[2] Assert RST, SCK, SDI and CMD to logical zero. Then:

- Pulse SCK to logical one, then to logical zero four times.
- Assert RST to logical one. Reset circuit places XDR DRAM into low-power state (identical to power-on reset).
- Perform remaining initialization sequence in Figure 48.

[3] XDR DRAM has valid Serial ID and all registers have default values that are defined in Figure 17 through Figure 42.

[4] Perform broadcast or directed register writes to adjust registers which need a value different from their default value.

[5] Perform Powerdown Exit sequence shown in Figure 46. This includes the activity from SCK cycle S_0 through the final REFP command.

[6] Perform termination current calibration. The CALZ/CALE sequence shown in Figure 45 is issued 128 times, then the CALC/CALE sequence is issued 128 times. After this, each sequence is issued once every t_{CALZ} or t_{CALC} interval.

[7] Condition the XDR DRAM banks by performing a REFA/REFI activate and REFP precharge operation to each bank eight times. This can be interleaved to save time. The row address for the activate operation will step through eight successive values of the REFr registers. The sequence between cycles T_0 and T_{32} in the Interleaved Refresh Example in Figure 44 could be performed eight times to satisfy this conditioning requirement.

XDR DRAM Pattern Load with WDSL Register

The XDR memory system requires a method of deterministically loading pattern data to XDR DRAMs before beginning Receive Timing Calibration (RX TCAL). The method employed by the XDR DRAMs to achieve this is called Write Data Serial Load (WDSL). A WDSL packet sends one-byte of serial data which is serially shifted into a holding register within the XDR DRAM. Initialization software sends a sequence of WDSL packets, each of which shifts the new byte in and advances the shifter by 8 positions. In this way, XDR DRAMs of varying widths can be loaded with a single command type.

Each sequence of WDSL packets will load one full column of data to the internal holding register of the target XDR DRAM. Depending upon the ratio of native device width to programmed width, there may be more than one sub-column per column. After loading a full column, a series of WR commands will be issued to sequentially transfer each sub-column of the column to the XDR DRAM core (s), based upon the SC [3:0] bits.

Table 10. WDSL-to-Core/DQ/SC Map (First Generation ×16/×8/×4 XDR DRAM, BL = 16)

LOGICAL VIEW OF XDR DRAM					Word Written (1 = Written, 0 = Not Written)						
DQ Pin Used			Core Word	WDSL Core Word Load Order	×16	×8		×4			
×4	×8	×16		WD [n] [15:0]	SC[3:2]=xx	SC[3:2]=0x	SC[3:2]=1x	SC[3:2]=00	SC[3:2]=01	SC[3:2]=10	SC[3:2]=11
DQ0	DQ0	DQ0	WD [0] [15:0]	WDSL Word 8	1	1	0	1	0	0	0
DQ1	DQ1	DQ1	WD [1] [15:0]	WDSL Word 7	1	1	0	1	0	0	0
DQ2	DQ2	DQ2	WD [2] [15:0]	WDSL Word 12	1	1	0	1	0	0	0
DQ3	DQ3	DQ3	WD [3] [15:0]	WDSL Word 3	1	1	0	1	0	0	0
DQ0	DQ4	DQ4	WD [4] [15:0]	WDSL Word 10	1	1	0	0	1	0	0
DQ1	DQ5	DQ5	WD [5] [15:0]	WDSL Word 5	1	1	0	0	1	0	0
DQ2	DQ6	DQ6	WD [6] [15:0]	WDSL Word 14	1	1	0	0	1	0	0
DQ3	DQ7	DQ7	WD [7] [15:0]	WDSL Word 1	1	1	0	0	1	0	0
DQ0	DQ0	DQ8	WD [8] [15:0]	WDSL Word 9	1	0	1	0	0	1	0
DQ1	DQ1	DQ9	WD [9] [15:0]	WDSL Word 6	1	0	1	0	0	1	0
DQ2	DQ2	DQ10	WD [10] [15:0]	WDSL Word 13	1	0	1	0	0	1	0
DQ3	DQ3	DQ11	WD [11] [15:0]	WDSL Word 2	1	0	1	0	0	1	0
DQ0	DQ4	DQ12	WD [12] [15:0]	WDSL Word 11	1	0	1	0	0	0	1
DQ1	DQ5	DQ13	WD [13] [15:0]	WDSL Word 4	1	0	1	0	0	0	1
DQ2	DQ6	DQ14	WD [14] [15:0]	WDSL Word 15	1	0	1	0	0	0	1
DQ3	DQ7	DQ15	WD [15] [15:0]	WDSL Word 0	1	0	1	0	0	0	1
PHYSICAL VIEW OF XDR DRAM					Word Written (1 = Written, 0 = Not Written)						
DQ2	DQ6	DQ14	WD [14] [15:0]	WDSL Word 15	1	0	1	0	0	0	1
		DQ6	WD [6] [15:0]	WDSL Word 14	1	1	0	0	1	0	0
	DQ2	DQ10	WD [10] [15:0]	WDSL Word 13	1	0	1	0	0	1	0
		DQ2	WD [2] [15:0]	WDSL Word 12	1	1	0	1	0	0	0
DQ0	DQ4	DQ12	WD [12] [15:0]	WDSL Word 11	1	0	1	0	0	0	1
		DQ4	WD [4] [15:0]	WDSL Word 10	1	1	0	0	1	0	0
	DQ0	DQ8	WD [8] [15:0]	WDSL Word 9	1	0	1	0	0	1	0
		DQ0	WD [0] [15:0]	WDSL Word 8	1	1	0	1	0	0	0
DQ1	DQ3	DQ3	WD [1] [15:0]	WDSL Word 7	1	1	0	1	0	0	0
		DQ11	WD [9] [15:0]	WDSL Word 6	1	0	1	0	0	1	0
	DQ7	DQ7	WD [5] [15:0]	WDSL Word 5	1	1	0	0	1	0	0
		DQ15	WD [13] [15:0]	WDSL Word 4	1	0	1	0	0	0	1
DQ3	DQ3	DQ3	WD [3] [15:0]	WDSL Word 3	1	1	0	1	0	0	0
		DQ11	WD [11] [15:0]	WDSL Word 2	1	0	1	0	0	1	0
	DQ7	DQ7	WD [7] [15:0]	WDSL Word 1	1	1	0	0	1	0	0
		DQ15	WD [15] [15:0]	WDSL Word 0	1	0	1	0	0	0	1

Table 11. Core Data Word-to WDSL Format[a]

DQ Serialization Order																
CFM/PCLK Cycle	Cycle 0								Cycle 1							
Symbol (Bit) Time	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15
Bit Transmitted on DQ pins	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
WDSL Byte/Bit Transfer Order																
Core Word	Core Word WD [n] [15:0]															
WDSL Word	WDSL Word n															
WDSL Byte Order	WDSL Byte 0								WDSL Byte 1							
SWD Field of Serial Packet	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit Transmitted on CMD pin	D15	D11	D7	D3	D14	D10	D6	D2	D13	D9	D5	D1	D12	D8	D4	D0

Special Feature Description

Dynamic Width Control

This XDR DRAM device includes a feature called dynamic width control. This permits the device to be configured so that read and write data can be accessed through differing widths of DQ pins. Figure 46 shows a diagram of the logic in the path of the read data (Q) and write data (D) that accomplishes this.

The read path is on the right of the figure. There are 16 sets of S signals (the internal data bus connecting to the sense amps of the memory core), with 16 signals in each set. When the XDR DRAM device is configured for maximum width operation (using the WIDTH [2:0] field in the CFG register), each set of 16 S signals goes to one of the 16 DQ pins (via the Q [15:0] [15:0] read bus) and are driven out in the 16 time slots for a read data packet.

When the XDR DRAM device is configured for a width that is less than the maximum, some of the DQ pins are used and the rest are not used. The SC [3:0] field of the COL request packets selects which S [15:0] [15:0] signals are passed to the Q [15:0] [15:0] read bus and driven as read data.

Figure 50 shows the mapping from the S bus to the Q bus as a function of the WIDTH [2:0] register field and the SC [3:0] field of the COL request packet. There is a separate table for each valid value of WIDTH [2:0]. In each table, there is an entry in the left column for each valid value of SC [3:0]. This field should be treated as an extension of the C [9:4] column address field. The right hand column shows which sets of S [15:0] [15:0] signals are mapped to the Q read data bus for a particular value of SC [3:0].

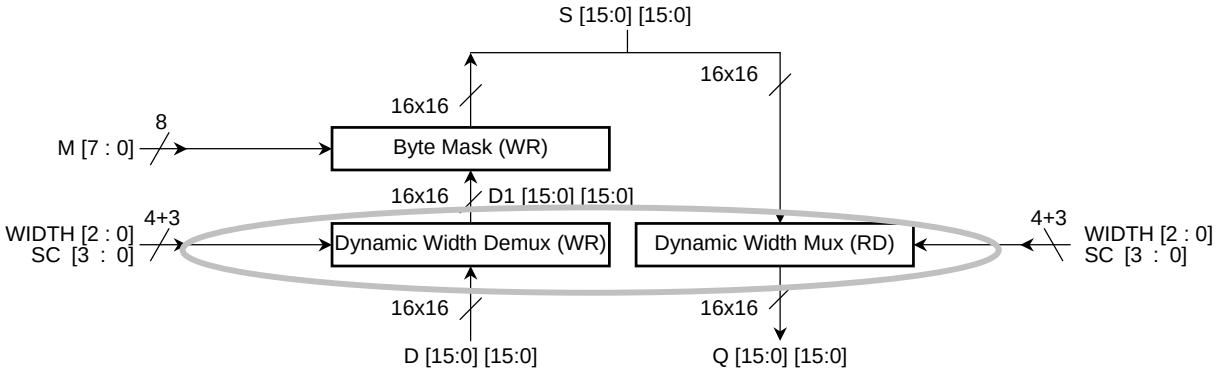
For example, assume that the WIDTH [2:0] value is “010”, indicating a device width of x4. Looking at the appropriate table in Figure 47, it may be seen that in the SC [3:0] field, the SC [1:0] sub-column address bits are not used. The remaining SC [3:0] address bit(s) selects one of the 64-bit blocks of S bus signals, causing them to be driven onto the Q [3:0] [15:0] read data bus, which in turn is driven to the DQ3…DQ0/DQN3…DQN0 data pins. The Q [15:4] [15:0] signals and DQ15…DQ4/DQN15...DQN4 data pins are not used for a device width of x4.

The write path is shown on the left side of Figure 46. As before, there are 16 sets of S signals (the internal data bus connecting to the sense amps of the memory core), with 16 signals in each set. When the XDR DRAM device is configured for maximum width operation (using the WIDTH [2:0] field in the CFG register), each set of 16 S signals is driven from one of the 16 DQ pins (via the D [15:0] [15:0] write bus) from each of the 16 time slots for a write data packet.

Figure 50 also shows the mapping from the D bus to the S bus as a function of the WIDTH [2:0] register field and the SC [3:0] field of the COL request packet. There is a separate table for each valid value of WIDTH [2:0]. In each table, there is an entry in the left column for each valid value of SC [3:0]. This field should be treated as an extension of the C [9:4] column address field. The right hand column shows which set of S [15:0] [15:0] signals are mapped from the D read data bus for a particular value of SC [3:0].

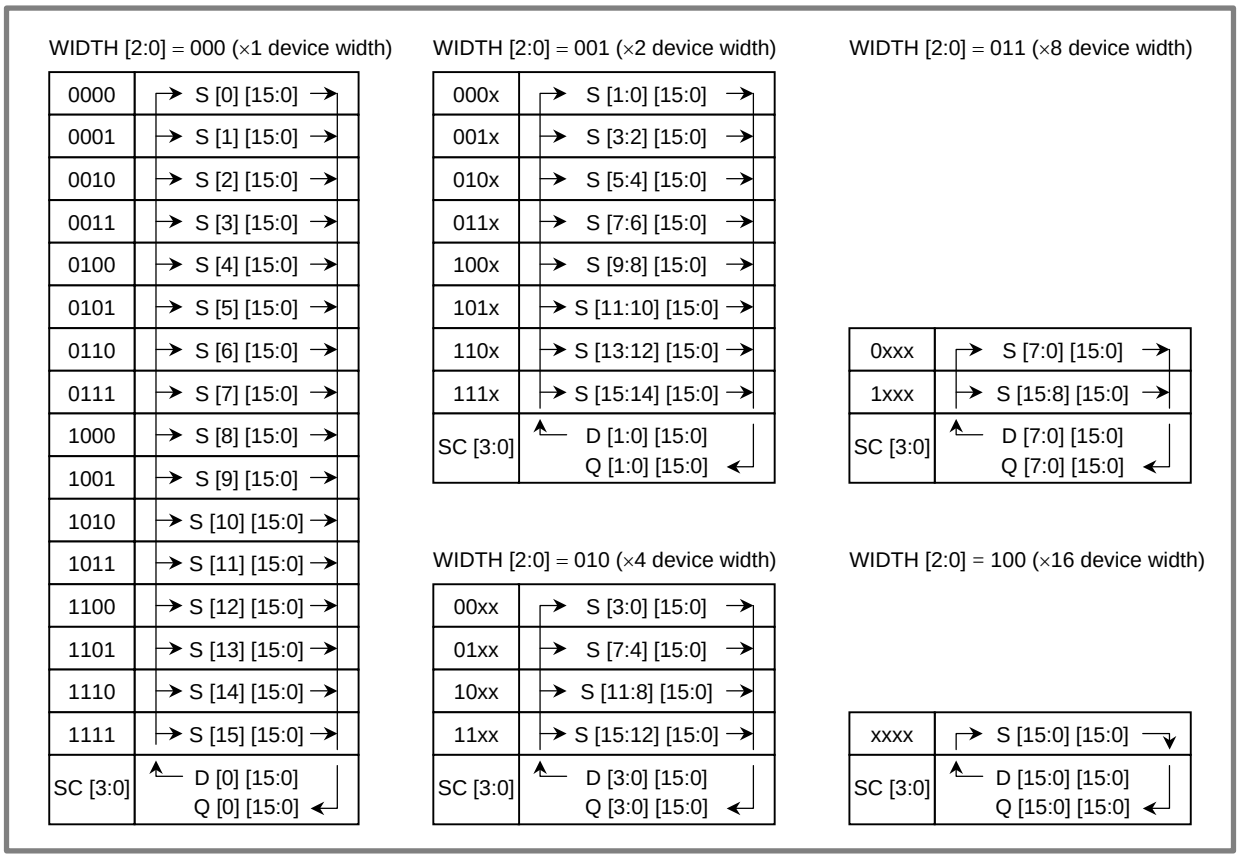
For example, assume that the WIDTH [2:0] value is “001”, indicating a device width of x2. Looking at the appropriate table in Figure 47, it may be seen that in the SC [3:0] field, the SC [0] sub-column address bit is not used. The remaining SC [3:0] address bit(s) selects one of the 32-bit blocks of S bus signals, causing them to be driven from the D [1:0] [15:0] write data bus, which in turn is driven from the DQ1…DQ0/DQN1...DQN0 data pins. The D [15:2] [15:0] signals and DQ15…DQ2/DQN15…DQN2 data pins are not used for a device width of x2.

Figure 49. Multiplexes for Dynamic Width Control



The block diagram in Figure 49 indicates that the Dynamic Width logic is positioned after the serial-to-parallel conversion (demux block) in the data receiver block and before the parallel-to-serial conversion (mux block) in the data transmitter block (see the block diagram in Figure 2 also). The block diagram is shown in this manner so the functionality of the logic can be made as clear as possible. Some implementations may place this logic in the data receiver and transmitter blocks, performing the mapping in Figure 50 on the serial data rather than the parallel data. However, this design choice will not affect the functionality of the Dynamic Width logic; it is strictly an implementation decision.

Figure 50. D-to-S and S-to-Q Mapping for Dynamic Width Control



Write Masking

Figure 51 shows the logic used by the XDR DRAM device when a write-masked command (WRM) is specified in a COLM packet. This masking logic permits individual bytes of a write data packet to be written or not written according to the value of an eight bit write mask M [7:0].

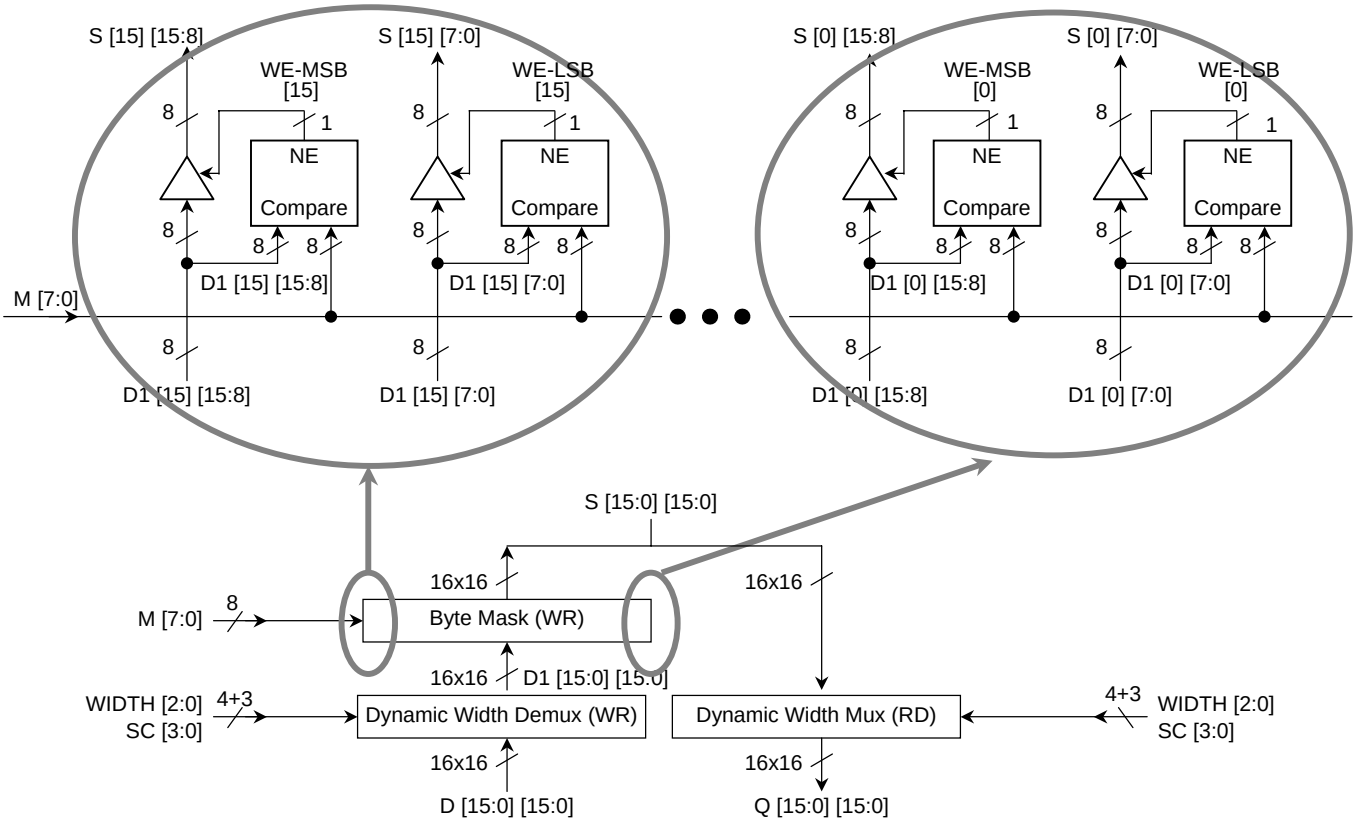
In Figure 51, there are 16 sets of 16 bit signals forming the D1 [15:0] [15:0] input bus for the Byte Mask block. These are treated as 2x16 8-bit bytes:

D1 [15] [15:8]
D1 [15] [7:0]
...
D1 [1] [15:8]
D1 [1] [7:0]
D1 [0] [15:8]
D1 [0] [7:0]

The eight bits of each byte is compared to the value in the byte mask field (M [7:0]). If they are not equal (NE), then the corresponding write enable signal (WE) is asserted and the byte is written into the sense amplifier. If they are equal, then the corresponding write enable signal (WE) is deserted and the byte is not written into the sense amplifier.

In the example of Figure 52, a WRM command performs a masked write of a 64-byte data packet to all the memory devices connected to the RQ bus (and receiving the command). It is the job of the memory controller to search the 64-bytes to find an eight bit data value that is not used and place it into the M [7:0] field. This will always be possible because there are 256 possible 8-bit values and there are only 64 possible values used in the bytes in the data packet.

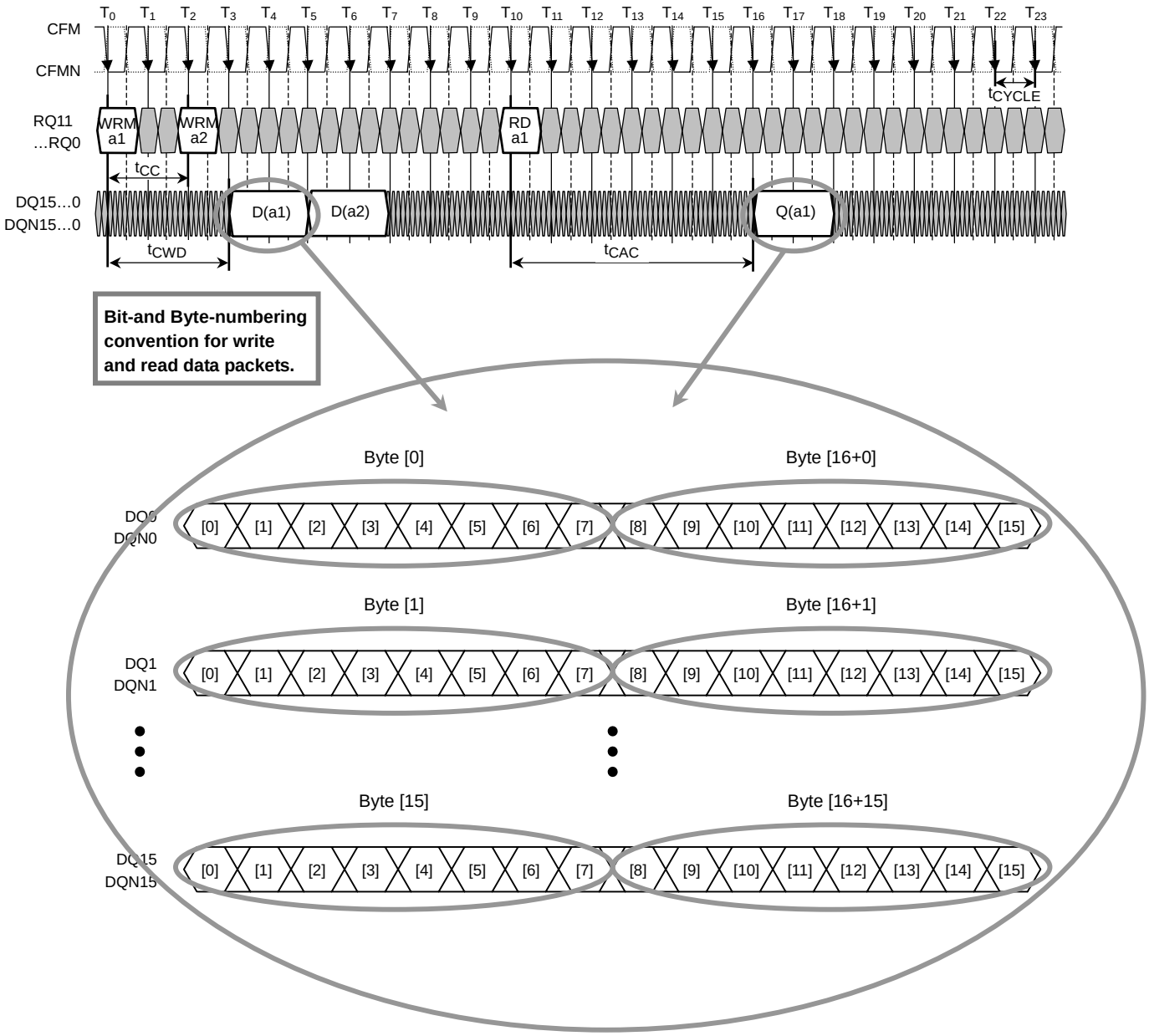
Figure 51. Byte Mask Logic



Note that other systems might use a data transfer size that is different than the 64 bytes per t_{CC} interval per RQ bus that is used in the example in Figure 51.

Figure 52 shows the timing of two successive WRM commands in COLM packets. The timing is identical to that of two successive WR commands in COL packets. The one difference is that the COLM packet includes a M [7:0] field that indicates the reserved bit pattern (for the eight bits of each byte) that indicates that the byte is not to be written. This requires that the alignment of bytes within the data packet be defined, and also that the bit numbering within each byte be defined (note that this was not necessary for the unmasked WR command). In the figure, bytes are contained within a single DQ/DQN pin pair - this is necessary so the dynamic width feature can be supported. Thus, each pin pair carries two bytes of each data packet. Byte [0] is transferred earlier than byte [1], and bit [0] of each byte (corresponding to M [0]) is transferred first, followed by the remaining bits in succession).

Figure 52. Write-Masked (WRM) Transaction Example



Multiple Bank Sets and the ERAW Feature

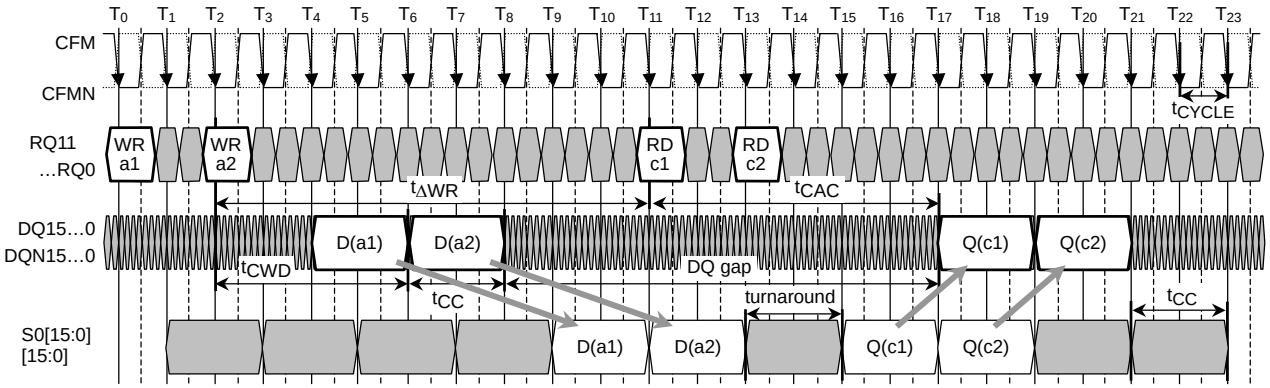
Figure 55 shows a block diagram of a XDR DRAM in which the banks are divided into two sets (called the even bank set and the odd bank set) according to the least significant bit of the bank address field. This XDR DRAM supports a feature called “Early Read After Write” (hereafter called “ERAW”).

The logic that accepts commands on the RQ11…RQ0 signals is capable of operating these two bank sets independently. In addition, each bank set connects to its own internal “S” data bus (called S0 and S1). The receive interface is able to drive write data onto either of these internal data buses, and the transmit interface is able to sample read data from either of these internal data buses. These capabilities will permit the delay between a write column operation and a read column operation to be reduced, thereby improving performance.

Figure 50 shows the timing previously presented in Figure 12, but with the activity on the internal S data bus included. The write-to-read parameter $t_{\Delta WR}$ ensures that there is adequate turnaround time on the S bus between D (a2) and Q (c1).

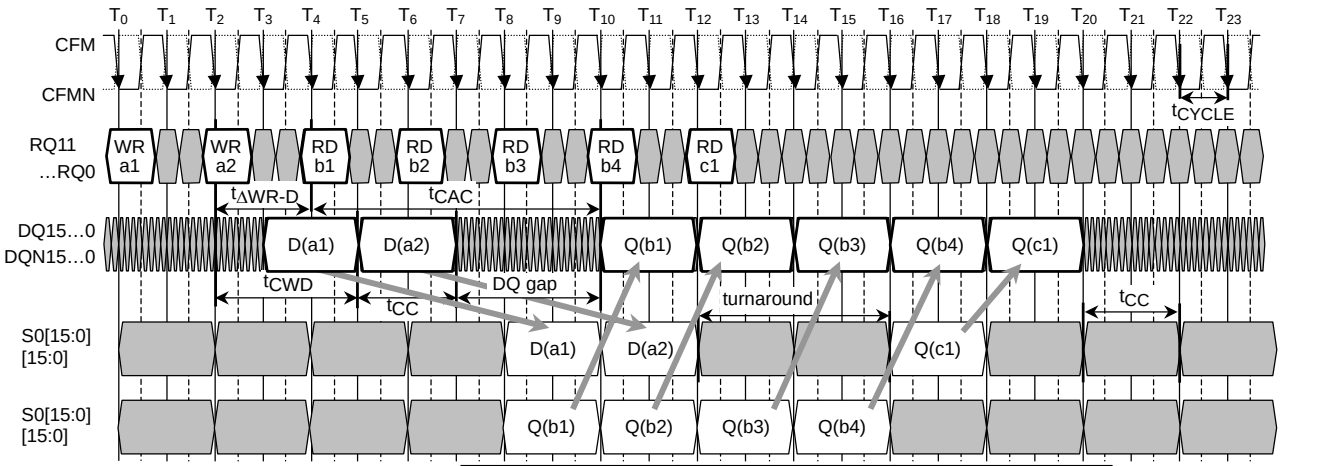
When ERAW is supported with odd and even bank sets, the $t_{\Delta WR, MIN}$ parameter must be obeyed when the write and read column operations are to the same bank set, but a second parameter $t_{\Delta WR-D}$ permits earlier column operations to the opposite bank set. Figure 51 shows how this is possible because there are two internal data buses S0 and S1. In this example, the four columns read operations are made to the same bank Bb, but they could use different banks as long as they all belonged to the bank set that was different from the bank set containing Ba (for the column write operations).

Figure 53. Write/Read Interaction – No ERAW Feature



Transaction a: WR	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}
Transaction c: RD	c1 = {Bc, Cc1}	c2 = {Bc, Cc2}

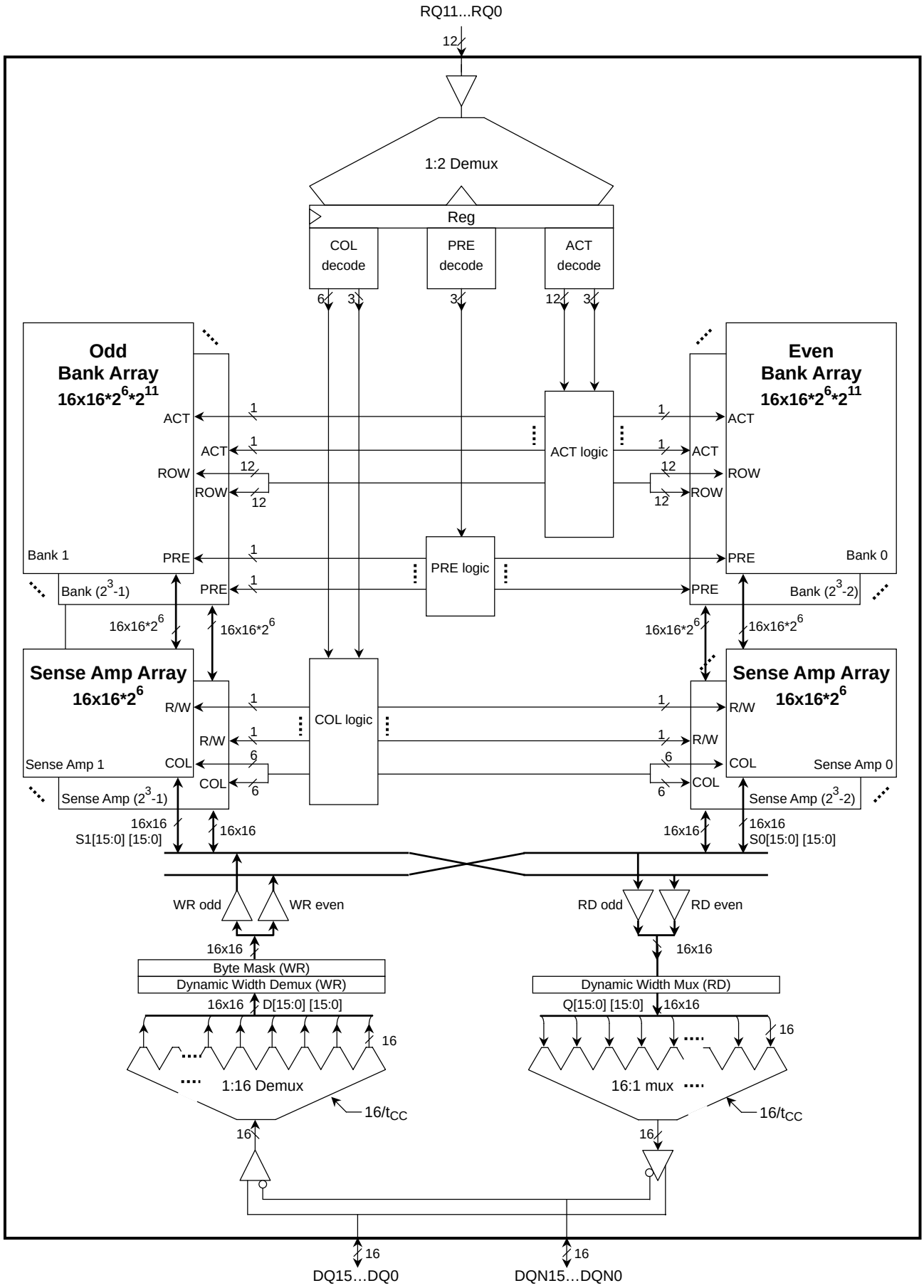
Figure 54. Write/Read Interaction – ERAW Feature



Bank Restrictions
Bb is in different bank set than Ba
Be is in same bank set as Ba

Transaction a: WR	a1 = {Ba, Ca1}	a2 = {Ba, Ca2}	
Transaction b: RD	b1 = {Bb, Cb1}	b2 = {Bb, Cb2}	B3 = {Bb, Cb3}
Transaction c: RD	c1 = {Bc, Cc1}	c2 = {Bc, Cc2}	

Figure 55. XDR DRAM Block Diagram with Bank Sets



Simultaneous Precharge

When the XDR DRAM supports multiple bank sets as in Figure 55, another feature may be supported, in addition to ERAW. This feature is simultaneous precharge, and the timing of several cases is shown in Figure 56.

The t_{PP} parameter specifies the minimum spacing between packets with precharge commands in XDR DRAMs with a single bank set, or between packets to the same bank set in a XDR DRAM with multiple bank sets. The t_{PP-D} parameter specifies the minimum spacing between packets with precharge commands to different bank sets in a XDR DRAM with multiple bank sets.

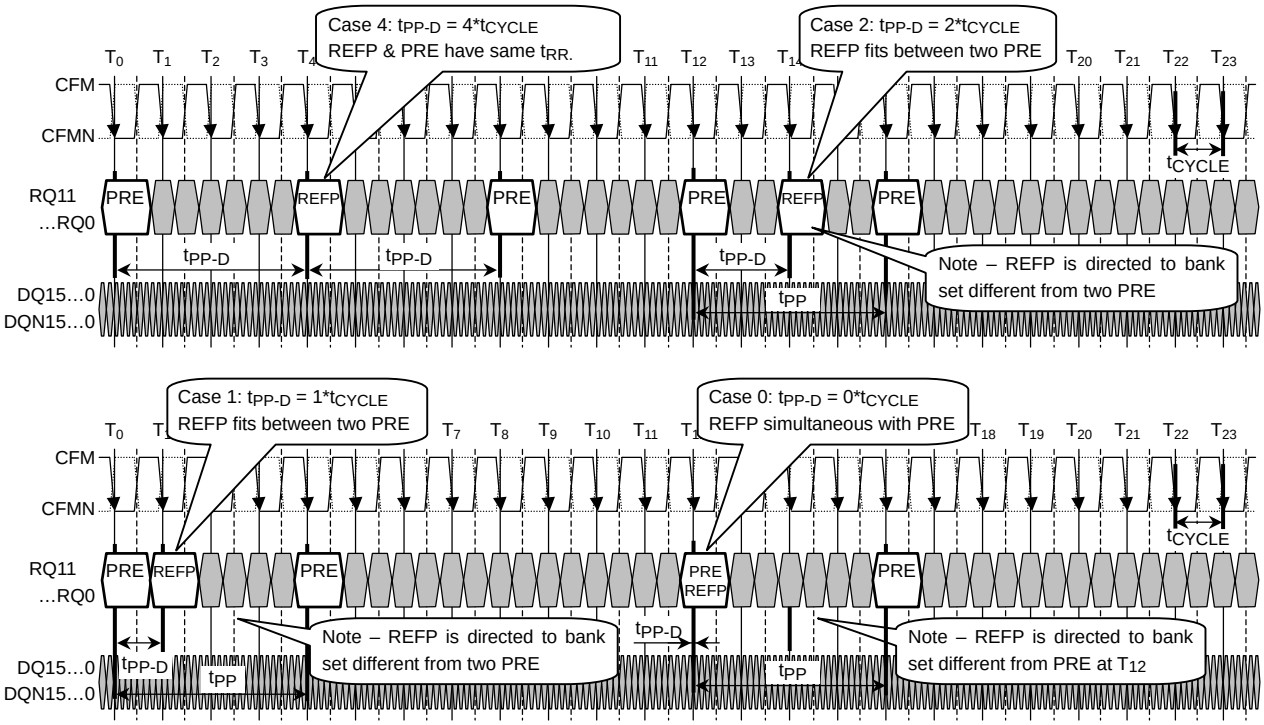
In Figure 56, Case 4 shows an example when both t_{PP} and t_{PP-D} must be at least $4 \times t_{CYCLE}$. In such a case, precharge commands to different bank sets satisfy the same constraint as precharge commands to the same bank set.

In Figure 56, Case 2 shows an example when t_{PP} must be at least $4 \times t_{CYCLE}$ and t_{PP-D} must be at least $2 \times t_{CYCLE}$. In such a case, a precharge command to one bank set may be inserted between two precharge commands to a different bank set.

In Figure 56, Case 1 shows an example when t_{PP} must be at least $4 \times t_{CYCLE}$ and t_{PP-D} must be at least $1 \times t_{CYCLE}$. As in the previous case, a precharge command to one bank set may be inserted between two precharge commands to a different bank set. In this case, the middle precharge command will not be symmetrically placed relative to the two outer precharge commands.

In Figure 56, Case 0 shows an example when t_{PP} must be at least $4 \times t_{CYCLE}$ and t_{PP-D} must be at least $0 \times t_{CYCLE}$. This means that two precharge commands may be issued on the same CFM clock edge. This is only possible by using the delay mechanism in one of the two commands. See “Dynamic Request Scheduling” on page 20. It is also possible by taking advantage of the fact that two independent precharge commands may be encoded within a single ROWP packet. In the example shown, the ROWP packet contains both a REFP command and a PRE command. Both precharge commands will be issued internally to different bank sets on the same CFM clock edge.

Figure 56. Simultaneous Precharge – t_{PP-D} Cases



Operating Conditions

Electrical Conditions

Table 12 summarizes all electrical conditions (temperature and voltage conditions) that may be applied to the memory component. The first section of parameters is concerned with absolute voltages, storage and operating temperatures, and the power supply, reference, and termination voltages.

The second section of parameters determines the input voltage levels for the RSL RQ signals. The high and low voltages must satisfy a symmetry parameter with respect to the $V_{REF, RSL}$.

The third section of parameters determines the input voltage levels for the RSL SI (serial interface) signals. The high and low voltages must satisfy a symmetry parameter with respect to the $V_{REF, RSL}$.

The fourth section of parameters determines the input voltage levels for the CFM clock signals. The high and low voltages are specified by a common-mode value and a swing value.

The fifth section of parameters determines the input voltage levels for the write data signals on the DRSL DQ pins. The high and low voltages are specified by a common-mode value and a swing value.

Table 12. Electrical Conditions

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{IN, ABS}$	Voltage applied to any pin (except V_{DD}) with respect to GND	–0.300	1.500	V
$V_{DD, ABS}$	Voltage on V_{DD} with respect to GND	–0.500	2.300	V
T_{STORE}	Storage temperature	–50	100	°C
T_J	Junction temperature under bias during normal operation	Manufacturer-specific values		°C
V_{DD}	Supply voltage applied to V_{DD} pins during normal operation	1.80 – 0.060	1.80 + 0.060	V
$V_{REF,RSL}$	RSL – Reference voltage applied to V_{REF} pin [a]	$V_{TERM,RSL}$ – 0.450 – 0.025	$V_{TERM,RSL}$ – 0.450 + 0.025	V
$V_{TERM,DRSL}$	RSL – Termination voltage applied to V_{TERM} pins	1.200 – 0.060	1.200 + 0.060	V
$V_{IL,RQ}$	RSL RQ inputs – low voltage	$V_{REF,RSL}$ – 0.450	$V_{REF,RSL}$ – 0.150	V
$V_{IH,RQ}$ [b]	RSL RQ inputs – high voltage	$V_{REF,RSL}$ + 0.150	$V_{REF,RSL}$ + 0.450	V
$R_{A,RQ}$	RSL RQ inputs – data asymmetry: $R_{A,RQ} = (V_{IH,RQ} - V_{REF,RSL}) / (V_{REF,RSL} - V_{IL,RQ})$	TBD	TBD	V
$V_{IL,SI}$	RSL Serial Interface inputs – low voltage	$V_{REF,RSL}$ – 0.450	$V_{REF,RSL}$ – 0.200	V
$V_{IH,SI}$ [b]	RSL Serial Interface inputs – high voltage	$V_{REF,RSL}$ + 0.200	$V_{REF,RSL}$ + 0.450	V
$R_{A,SI}$	RSL RQ inputs – data asymmetry: $R_{A, SI} = (V_{IH,RQ} - V_{REF,RSL}) / (V_{REF,RSL} - V_{IL,RQ})$	TBD	TBD	V
$V_{ICM,CFM}$	CFM/CFMN input – common mode: $V_{ICM, CFM} = (V_{IH,CFM}^{b} + V_{IL,CFM})/2$	$V_{TERM,DRSL}$ – 0.150	$V_{TERM,DRSL}$ – 0.075	V
$V_{ISW,CFM}$	CFM/CFMN input – high-low swing: $V_{ISW, CFM} = (V_{IH,CFM}^{b} - V_{IL,CFM})$	0.150	0.300	V
$V_{ICM,DQ}$	DRSL DQ inputs – common mode: $V_{ICM, DQ} = (V_{IH,DQ}^{b} + V_{IL,DQ})/2$	$V_{TERM,DRSL}$– 0.150	$V_{TERM,DRSL}$– 0.025	V
$V_{ISW,DQ}$	DRSL DQ inputs – high-low swing: $V_{ISW, DQ} = (V_{IH,DQ}^{b} - V_{IL,DQ})$	0.050	0.300	V

a $V_{TERM,DRSL}$ is typically 1.2000 V ± 0.060 V. It connects to the RSL termination components, not to this DRAM component.

b V_{IH} is typically equal to $V_{TERM,RSL}$ or $V_{TERM,DRSL}$ (whichever is appropriate) under DC conditions in an system.

Timing Conditions

Table 13 summarizes all timing conditions that may be applied to the memory component. The first section of parameters is concerned with parameters for the clock signals. The second section of parameters is concerned with parameters for the request signals. The third section of parameters is concerned with parameters for the write data signals. The fourth section of parameters is concerned with parameters for the serial interface signals. The fifth section is concerned with all other parameters, including those for refresh, calibration, power state transitions, and initialization

Table 13. Timing Conditions

SYMBOL	PARAMETER		MIN	MAX	UNITS	Figure (s)
t_{CYCLE} or $t_{CYC,CFM}$	CFM RSL clock - cycle time	–4000	2.000	3.830	ns	Figure 58
		–3200	2.500	3.830		
		–2400	3.333	3.830		
$t_{R, CFM}$, $t_{F, CFM}$	CFM/CFMN input - rise and fall time - use minimum for test.		0.080	0.200	t_{CYCLE}	Figure 58
$t_{H, CFM}$, $t_{L, CFM}$	CFM/CFMN input – high and low times		40%	60%	t_{CYCLE}	Figure 58
$t_{R, RQ}$, $t_{F, RQ}$	RSL RQ input - rise/fall times (20% - 80%) - use minimum for test.		0.080	0.260	t_{CYCLE}	Figure 59
$t_{S, RQ}$, $t_{H, RQ}$	RSL RQ input to sample points (set/hold)	@ 2.500 ns > t_{CYCLE} ≥ 2.000 ns	TBD	—	ns	Figure 59
		@ 3.333 ns > t_{CYCLE} ≥ 2.500 ns	0.200	—		
		@ 3.830 ns ≥ t_{CYCLE} ≥ 3.333 ns	TBD	—		
$t_{IR, DQ}$, $t_{IF, DQ}$	DRSL DQ input - rise/fall times (20% - 80%) - use minimum for test.		0.020	0.074	t_{CYCLE}	Figure 60
$t_{S, DQ}$, $t_{H, DQ}$	DRSL DQ input to sample points (set/hold)	@ 2.500 ns > t_{CYCLE} ≥ 2.000 ns	TBD	—	ns	Figure 60
		@ 3.333 ns > t_{CYCLE} ≥ 2.500 ns	0.0625	—		
		@ 3.830 ns ≥ t_{CYCLE} ≥ 3.333 ns	TBD	—		
$t_{DOFF, DQ}$	DRSL DQ input delay offset (fixed) to sample points		–0.080	0.080	t_{CYCLE}	Figure 60
$t_{CYC, SCK}$	Serial Interface SCK input - cycle time		20	—	ns	Figure 62
$t_{R, SCK}$, $t_{F, SCK}$	Serial Interface SCK input - rise and fall times		—	5.0	ns	Figure 62
$t_{H, SCK}$, $t_{L, SCK}$	Serial Interface SCK input - high and low times		40%	60%	t_{CYCLE}	Figure 62
$t_{IR, SI}$, $t_{IF, SI}$	Serial Interface CMD, RST, SDI input - rise and fall times		—	5.0	ns	Figure 62
$t_{S, SI}$, $t_{H, SI}$	Serial Interface CMD, SDI input to SCK clock edge - set/hold time		5	—	ns	Figure 62
$t_{DLY, SI-RQ}$	Delay from last SCK clock edge for register write to first CFM edge with RQ packet. Also, delay from first CFM edge with RQ packet to the first SCK clock edge for register operation.		10	—	t_{CYCLE}	—
t_{REF}	Refresh interval. Every row of every bank must be accessed at least once in this interval with a ROW-ACT, ROWP-REF or ROWP-REFI command.		—	16	ms	Figure 44
$t_{REFA-REFA,AVG}$	Average refresh command interval. ROWP-REFA or ROWP-REFI commands must be issued at this average rate. This depends upon t_{REF} and the number of banks and the number of rows: t_{REFI} = $t_{REF}/(N_B*N_R)$ = $t_{REF}/(2^3*2^{11})$.		$t_{REFA-REFA,AVG}$=977		ns	—
$N_{REFA, BURST}$	Refresh burst limit. The number of ROWP-REFA or ROWP-REFI commands which can be issued consecutively at the minimum command spacing.		—	128	commands	—
$t_{BURST-REFA}$	Refresh burst interval. The interval between a burst of $N_{REFA,BURST,MAX}$ ROWP-REFA or ROWP-REFI commands and the next ROWP-REFA or ROWP-REFI command.		—	TBD	t_{CYCLE}	—
$t_{COREINIT}$	Interval needed for one initialization after power is applied.		—	1.500	ms	—
t_{CALC}	Current calibration interval		—	100	ms	Figure 45
$t_{CMD-CALC}$ $t_{CMD-CALZ}$	Delay between packet with any command and CALC packet	w/ PRE or REFP command	4	—	t_{CYCLE}	Figure 45
		w/ any other command	16	—		
t_{CALCE}, t_{CALZE}	Delay between CALC/CALZ packet and CALE packet		12	—	t_{CYCLE}	Figure 45
$t_{CALE-CMD}$	Delay between CALE packet and packet with any command		24	—	t_{CYCLE}	Figure 45
$t_{CMD-PDN}$	Last command before PDN entry		16	—	t_{CYCLE}	Figure 46
$t_{PDN-CFM}$	RSL CFM/CFMN stable after PDN entry		16	—	t_{CYCLE}	Figure 46
$t_{CFM-PDN}$	RSL CFM/CFMN stable before PDN exit		16	—	t_{CYCLE}	Figure 46
$t_{PDN-CMD}$	First command after PDN exit (includes lock time for CFM/CFMN)		4096	—	t_{CYCLE}	Figure 46

Operating Characteristics

Electrical Characteristics

Table 14 summarizes all electrical parameters (temperature, current and voltage) that characterize this memory component. The only exception is the supply current values (I_{DD}) under different operating conditions covered in the ***Supply Current Profile*** section.

The first section of parameters is concerned with the thermal characteristics of the memory component.

The second section of parameters is concerned with the current needed by the RQ pins and VREF pin.

The third section of parameters is concerned with the current needed by the DQ pins and voltage levels produced by the DQ pins when driving read data. This section is also concerned with the current needed by the VTERM pin, and with the resistance levels produced for the internal termination components that attach to the DQ pins.

The fourth section of parameters determines the output voltage levels and the current needed for the serial interface signals

Table 14. Electrical Characteristics

SYMBOL	PARAMETER		MIN	MAX	UNIT
Θ_{JC}	Junction-to-case thermal resistance		Manufacturer-specific values		°C/Watt
$I_{I, RSL}$	RSL RQ or Serial Interface input current @ ($0 \le V_{IN} \le V_{DD}$)		–10	10	μA
$I_{REF, RSL}$	$V_{REF, RSL}$ current @ $V_{REF, RSL, MAX}$ flowing into VREF pin		–10	10	μA
$V_{OSW,DQ}$	DRSL DQ outputs – high-low swing; $V_{OSW,DQ}$ = ($V_{IH,DQ} – V_{IL,DQN}$) or ($V_{IH,DQN} – V_{IL,DQ}$) (See Figure 27)	ODF = 00	0.225	0.350	V
		ODF = 01	TBD	TBD	
		ODF = 10	TBD	TBD	
		ODF = 11	TBD	TBD	
$R_{TERM,DQ}$	DRSL DQ outputs – termination resistance		40.0	60.0	Ω
$V_{OL, SI}$	RSL serial interface SDO output – low voltage		0.0	$V_{REF, RSL}$ – TBD	V
$V_{OH, SI}$	RSL serial interface SDO output – high voltage		$V_{REF, RSL}$ + TBD	$V_{TERM, RSL}$	V

Supply Current Profile

In this section, Table 15 summarizes the supply current (I_{DD}) that characterizes this memory component. This parameter is shown under different operating conditions.

Table 15. Supply Current Profile

Symbol	Parameter	Min @t_{CYCLE}=2.00ns	Max @t_{CYCLE}=2.50ns	Max @t_{CYCLE}=3.33ns	Unit
$I_{DD, PDN}$	Device in PDN, Self Refresh enabled	TBD	TBD	TBD	μA
$I_{DD, STBY}$	Device in STBY. This is for a device in STBY with no packets on the Channel.	TBD	TBD	TBD	mA
$I_{DD, REF}$	Device in STBY and refreshing rows at the $t_{REF,MAX}$ period.	TBD	TBD	TBD	mA
$I_{DD, WR}$	ACT command every t_{RR}, PRE command every t_{PP}, WR command every t_{CC}.	TBD	TBD	TBD	mA
$I_{DD, RD}$	ACT command every t_{RR}, PRE command every t_{PP}, RD command every t_{CC} [a].	TBD	TBD	TBD	mA
$I_{TERM,DRSL,NONE}$	No RD nor WR commands issued.[c]	TBD	TBD	TBD	mA
$I_{TERM,DRSL,RD}$	RD command every t_{CC}[c]	TBD	TBD	TBD	mA
$I_{TERM,DRSL,WR}$	WR command every t_{CC}[c]	TBD	TBD	TBD	mA

a. I_{DD} current @ $V_{DD,MAX}$ flowing into VDD pins.
b. This does not include the $I_{OL,DQ}$ sink current. The device dissipates $I_{OL,DQ}$*$V_{TERM,DQ}$ in each DQ/DQN pair when driving data.
c. $I_{TERM,DRSL}$ current @ $V_{TERM,DQ,MAX}$ flowing into VTERM pins.

Timing Characteristics

Table 16 summarizes all timing parameters that characterize this memory component. The only exceptions are the core timing parameters that are speed-bin dependent. Refer to the ***Timing Parameters*** section for more information.

The first section of parameters pertains to the timing of the DQ pins when driving read data.

The second section of parameters is concerned with the timing for the serial interface signals when driving register read data.

The third section of parameters is concerned with the time intervals needed by the interface to transition between power states.

Table 16. Timing Characteristics

Symbol	Parameter and Other Conditions		Min	Max	Unit	Figure (s)
$t_{Q, DQ}$	DRSL DQ output delay (variation across 16 Q bits on each DQ pin) from drive points – output delay	@ 2.500 ns > t_{CYCLE} ≥ 2.000 ns	TBD	TBD	ns	Figure 61
		@ 3.333 ns > t_{CYCLE} ≥ 2.500 ns	–0.0625	+0.0625		
		@ 3.830 ns ≥ t_{CYCLE} ≥ 3.333 ns	TBD	TBD		
$t_{QOFF, DQ}$	DRSL DQ output delay offset (a fixed value for all 16 Q bits on each DQ pin) from drive points – output delay		–0.080	+0.080	t_{CYCLE}	Figure 61
$t_{OR, DQ}$, $t_{OF, DQ}$	DRSL DQ output – rise and fall times (20% – 80%)		0.020	0.040	t_{CYCLE}	Figure 61
$t_{Q, SI}$	Serial SCK-to-SDO output delay @ $C_{LOAD, MAX}$ = 20 pF		2	12	ns	Figure 63
$t_{P, SI}$	Serial SDI-to-SDO propagation delay @ $C_{LOAD, MAX}$ = 20 pF		—	15	ns	Figure 63
$t_{OR, SI}$, $t_{OF, SI}$	Serial SDO output rise/fall (20% – 80%) @ $C_{LOAD, MAX}$ = 20 pF		—	5	ns	Figure 63
$t_{PDN-ENTRY}$	Time for power state to change after PDN entry		—	16	t_{CYCLE}	Figure 46
$t_{PDN-EXIT}$	Time for power state to change after PDN exit		0	—	t_{CYCLE}	Figure 46

Timing Parameters

Table 17 summarizes the timing parameters that characterize the core logic of this memory component. These timing parameters will vary as a function of the component's speed bin. The four sections deal with the timing intervals between packets with, respectively, row-row commands, row-column commands, column-column commands, and column-row commands.

Table 17. Timing Parameters

Symbol	Parameter and Other Conditions		Min (A)	Min (B)	Min (C)	Unit	Figure(s)
t_{RC}	Row-cycle time: interval between successive ROWA-ACT or ROWP-REFA or ROWP-REFI activate commands to the same bank.	t_{RC}	16	20	24	t_{CYCLE}	4, 7
		$t_{RC-R,2tCC} = t_{RCD-R} + t_{CC} + t_{RDP} + t_{RP}$ [a]	16	20	24		
		$t_{RC-R,2tCC,noERAW} = t_{RCD-W} + t_{CC} + t_{WRP} + t_{RP}$ [a]	19	24	24		
		$t_{RC-W,2tCC,\ ERAW} = t_{RCD-W} + t_{CC} + t_{WRP} + t_{RP}$ [a]	23	28	28		
t_{RAS}	Row-asserted time: interval between a ROWA-ACT or ROWP-REFA or ROWP-REFI activate command and a ROWP-PRE or ROWP-REFP precharge command to the same bank.		10	13	17	t_{CYCLE}	4, 7
t_{RP}	Row-precharge time: interval between a ROWP-PRE or ROWP-REFP precharge command and a ROWA-ACT or ROWP-REFA or ROWP-REFI activate command to the same bank.		6	7	7	t_{CYCLE}	4, 7
t_{PP}	Precharge-to-precharge time: interval between successive ROWP-PRE or ROWP-REFP precharge commands to different banks.	t_{PP}	4	4	4	t_{CYCLE}	4, 7
		t_{PP-D} [b]	1	1	1	t_{CYCLE}	
t_{RR}	Row-to-row time: interval between ROWA-ACT or ROWP-REFA or ROWP-REFI activate commands to different banks.	t_{RR}	4	4	4	t_{CYCLE}	4, 7
		t_{RR-D} [c]	4	4	4	t_{CYCLE}	
t_{RCD-R}	Row-to-column-read delay: interval between a ROWA-ACT activate command and a COL-RD read command to the same bank.		5	7	7	t_{CYCLE}	4, 7
t_{RCD-W}	Row-to-column-write delay: interval between a ROWA-ACT activate command and a COL-WR or COL-WRM write command to the same bank.		1	3	3	t_{CYCLE}	4, 7
t_{CAC}	Column access delay: interval from COL-RD read command to Q read data		6	7	7	t_{CYCLE}	10
t_{CWD}	Column write delay: interval from a COL-WR or COLM-WRM write command to D write data.		3	3	3	t_{CYCLE}	9
t_{CC}	Column-to-column time: interval between successive COL-RD commands, or between successive COL-WR or COLM-WRM commands.		2	2	2	t_{CYCLE}	4, 7
$t_{RW-BUB,\ XDRDRAM}$	Read-to-write bubble time: interval between the end of a Q read data packet and the start of D write data packet (the end of a data packet is the time interval t_{CC} after its start).		3	3	3	t_{CYCLE}	13
$t_{WR-BUB,\ XDRDRAM}$	Write-to-read bubble time: interval between the end of a D write data packet and the start of Q read data packet (the end of a data packet is the time interval t_{CC} after its start).		3	3	3	t_{CYCLE}	13
$t_{\Delta RW}$	Read-to-write time: interval between a COL-RD read command and a COL-WR or COLM-WRM write command.[d]		8	9	9	t_{CYCLE}	12
$t_{\Delta WR}$	Write-to-read time: interval between a COL-WR or COLM-WRM write command and a COL-RD read command.	$t_{\Delta WR}$	9	10	10	t_{CYCLE}	12
		$t_{\Delta WR-D}$ [e]	2	2	2	t_{CYCLE}	

Table 17. Timing Parameters (continued)

Symbol	Parameter and Other Conditions	Min (A)	Min (B)	Min (C)	Unit	Figure(s)
t_{RDP}	Read-to-precharge time: interval between a COL-RD read command and a ROWP-PRE precharge command to the same bank.	3	4	4	t_{CYCLE}	4, 7
t_{WRP}	Write-to-precharge time: interval between a COL-WR or COLM-WRM write command and a ROWP-PRE precharge command to the same bank.	10	12	12	t_{CYCLE}	4, 7
t_{DR}	Data-to-read time : interval between a D write data and a COL-RD read command to the same bank.	6	7	7	t_{CYCLE}	12, 13
t_{DP}	Data-to-precharge time : interval between a D write data and a ROWP-PRE precharge command to the same bank.	7	9	9	t_{CYCLE}	9
$t_{LRRn-LRRn}$	Interval between ROWP-LRRn command and a subsequent ROWP-LRRn command. [f]	16	20	24	t_{CYCLE}	Table 4
$t_{REFx-LRRn}$	Interval between ROWP-REFx command and a subsequent ROWP-LRRn command.	16	20	24	t_{CYCLE}	Table 4
$t_{REFx-LRRx}$	Interval between ROWP-LRRn command and a subsequent ROWP-REFx command.	16	20	24	t_{CYCLE}	Table 4

a. The $t_{RC,MIN}$ parameter is applicable to all transaction types (read, write, refresh, etc). Read and write transactions may have an additional limitation, depending upon how many column accesses (each requiring t_{CC}) are performed in each row access (t_{RC}). The table lists the special cases ($t_{RC-R,\ 2tCC}$, $t_{RC-W,\ 2tCC,\ noERAW}$, $t_{RC-W,\ 2tCC,\ ERAW}$) in which two column accesses are performed in each row access. Note that $t_{RC-W,\ 2tCC,\ ERAW}$ uses a relaxed value of t_{RCD-W} that is equal to $t_{RCD-R,MIN}$. All other parameters are minimum.

b. t_{PP-D} is the t_{PP} parameter for precharges to different bank sets. See “Simultaneous Precharge” on page 55.

c. t_{RR-D} is the t_{RR} parameter for aceivates to different bank sets. See “Simultaneous Precharge” on page 56.

d. See “Propagation Delay” on page 30.

e. $t_{\Delta WR-D}$ is the $t_{\Delta WR}$ parameter for write-read accesses to different bank sets. See “Multiple Bank Sets and the ERAW Feature” on page 53. Also, note that the value of $t_{\Delta WR-D}$ may not take on the values {3,5,7} within the range{$t_{\Delta WR-D,MIN}$, ... $t_{\Delta WR,MIN-1}$}. $t_{\Delta WR-D}$ may assume any value $\geq t_{\Delta WR,MIN}$.

f. ROWP-LRRn includes the commands {ROWP-LRR0, ROWP-LRR1, ROWP-LRR2}
ROWP-REFx inclues the commands {ROWP-REFA, ROWP-REF1, ROWP-REFP}.

Receive/Transmit Timing

Clocking

Figure 57 shows a timing diagram for the CFM/CFMN clock pins of the memory component. This diagram represents a magnified view of these pins. This diagram shows only one clock cycle.

CFM and CFMN are differential signals: one signal is the complement of the other. They are also high-true signals — a low voltage represents a logical zero and a high voltage represents a logical one. There are two crossing points in each clock cycle. The primary crossing point includes the high–voltage–to–low–voltage transition of CFM (indicated with the arrowhead in the diagram). The secondary crossing point includes the low–voltage–to–high–voltage transition of CFM. All timing events on the RSL signals are referenced to the first set of edges.

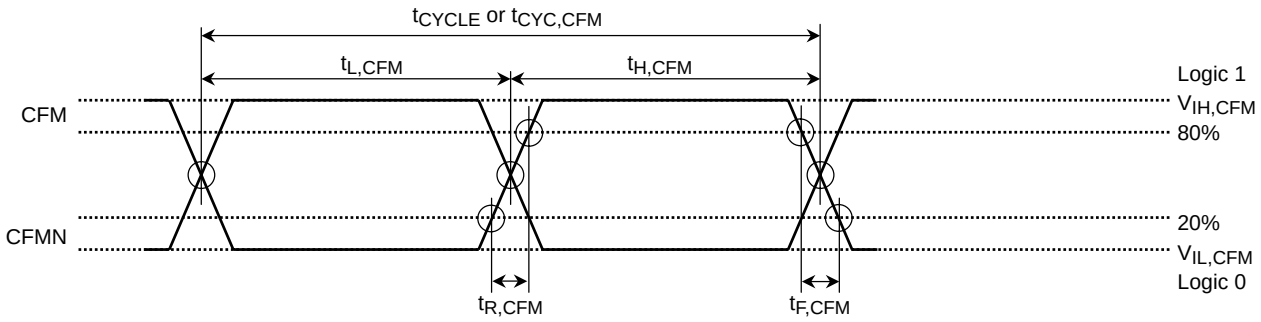
Timing events are measured to and from the crossing point of the CFM and CFMN signals. In the timing diagram, this is how the clock-cycle time (t_{CYCLE} or $t_{CYC,CFM}$), clock-low time ($t_{L,CFM}$) and clock-high time ($t_{H,CFM}$) are measured.

Because timing intervals are measured in this fashion, it is necessary to constrain the slew rate of the signals. The rise ($t_{R,CFM}$) and fall time ($t_{F,CFM}$) of the signals are measured from the 20% and 80% points of the full-swing levels.

$20\% = V_{IL,CFM} + 0.2*(V_{IH,CFM}–V_{IL,CFM})$

$80\% = V_{IL,CFM} + 0.8*(V_{IH,\ CFM}–V_{IL,CFM})$

Figure 57. Clocking Waveforms



RSL RQ Receive Timing

Figure 58 shows a timing diagram for the RQ11…RQ0 request pins of the memory component. This diagram represents a magnified view of the pins and only a few clock cycles (CFM and CFMN are the clock signals). Timing events are measured to and from the primary CFM/CFMN crossing point in which CFM makes its high-voltage-to-low-voltage transition. The RQ11…RQ0 signals are low true: a high voltage represents a logical zero and a low voltage represents a logical one. Timing events on the RQ11…RQ0 pins are measured to and from the point that the signal reaches the level of the reference voltage $V_{REF, RSL}$.

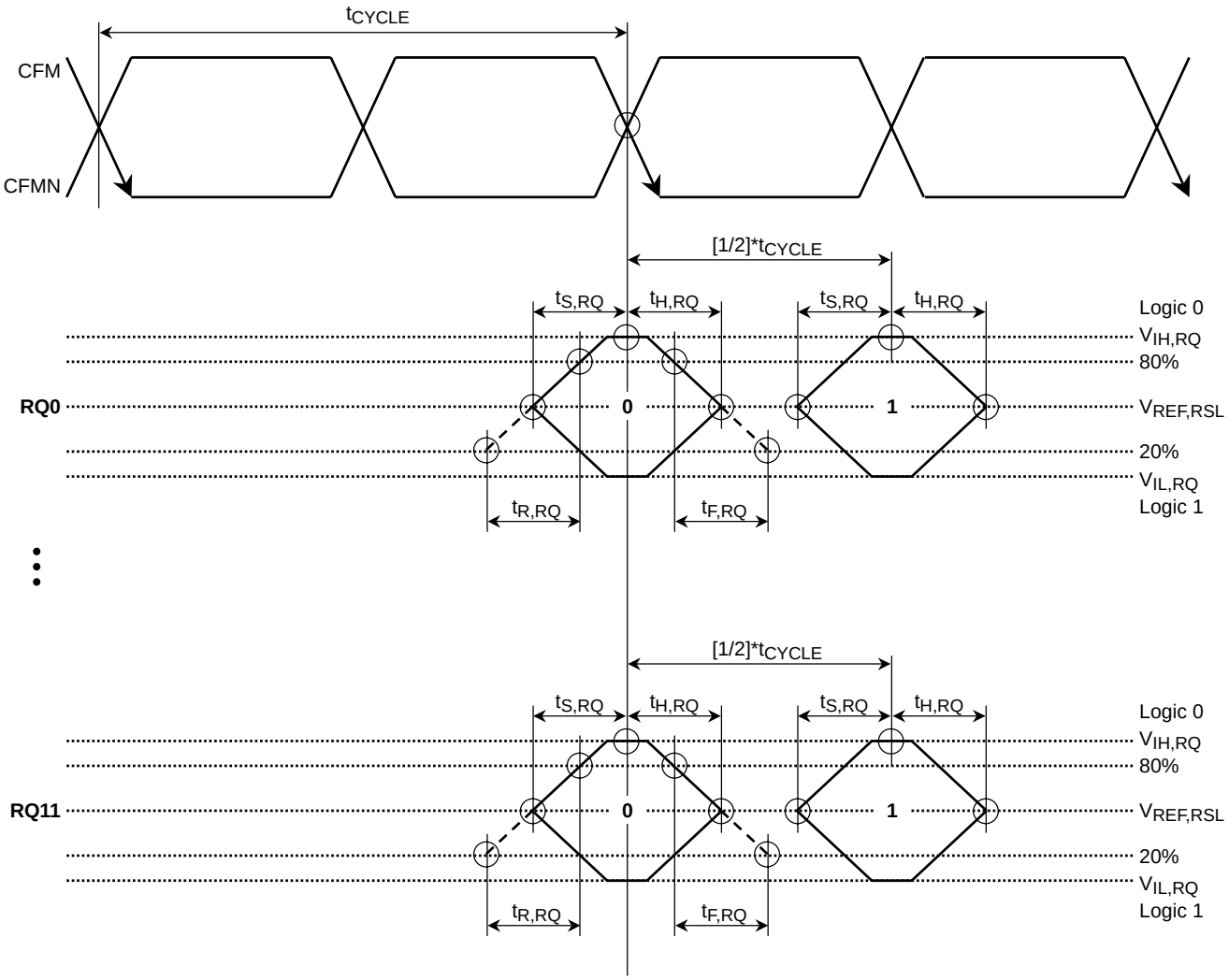
Because timing intervals are measured in this fashion, it is necessary to constrain the slew rate of the signals. The rise ($t_{R,RQ}$) and fall time ($t_{F,RQ}$) of the signals are measured from the 20% and 80% points of the full-swing levels.

$20\% = V_{IL,RQ} + 0.2*(V_{IH,RQ}-V_{IL,RQ})$

$80\% = V_{IL,RQ} + 0.8*(V_{IH,RQ}-V_{IL,RQ})$

There are two data receiving windows defined for each RQ11…RQ0 signal. The first of these (labeled “0”) has a set time, $t_{S,RQ}$, and a hold time, $t_{H,RQ}$, measured around the primary CFM/CFMN crossing point. The second (labeled “1”) has a set time ($t_{S,RQ}$) and a hold time ($t_{H,RQ}$) measured around a point $0.5 \times t_{CYCLE}$ after the primary CFM/CFMN crossing point.

Figure 58. RSL RQ Receive Waveform



DRSL DQ Receive Timing

Figure 59 shows a timing diagram for receiving write data on the DQ/DQN data pins of the memory component. This diagram represents a magnified view of the pins and shows only a few clock cycles are shown (CFM and CFMN are the clock signals). Timing events are measured to and from the primary CFM/CFMN crossing point in which CFM makes its high−voltage−to−low−voltage transition. The DQ15…DQ0/DQN15…DQN0 signals are high-true: a low voltage represents a logical zero and a high voltage represents a logical one. They are also differential—timing events on the DQ15…DQ0/DQN15...DQN0 pins are measured to and from the point that each differential pair crosses.

Because timing intervals are measured in this fashion, it is necessary to constrain the slew rate of the signals. The rise time ($t_{IR,DQ}$) and fall time ($t_{IF,DQ}$) of the signals are measured from the 20% and 80% points of the full-swing levels.

$20\% = V_{IL,DQ} + 0.2 \times (V_{IH,DQ} - V_{IL,DQ})$

$80\% = V_{IL,DQ} + 0.8 \times (V_{IH,DQ} - V_{IL,DQ})$

There are 16 data receiving windows defined for each DQ15…DQ0/DQN15…DQN0 pin pair. The receiving windows for a particular DQi/DQNi pin pair is referenced to an offset parameter $t_{DOFF,DQi}$ (the index “i” may take on the values {0, 1, ..15} and refers to each of the DQ15…DQ0/DQN15…DQN0 pin pairs).

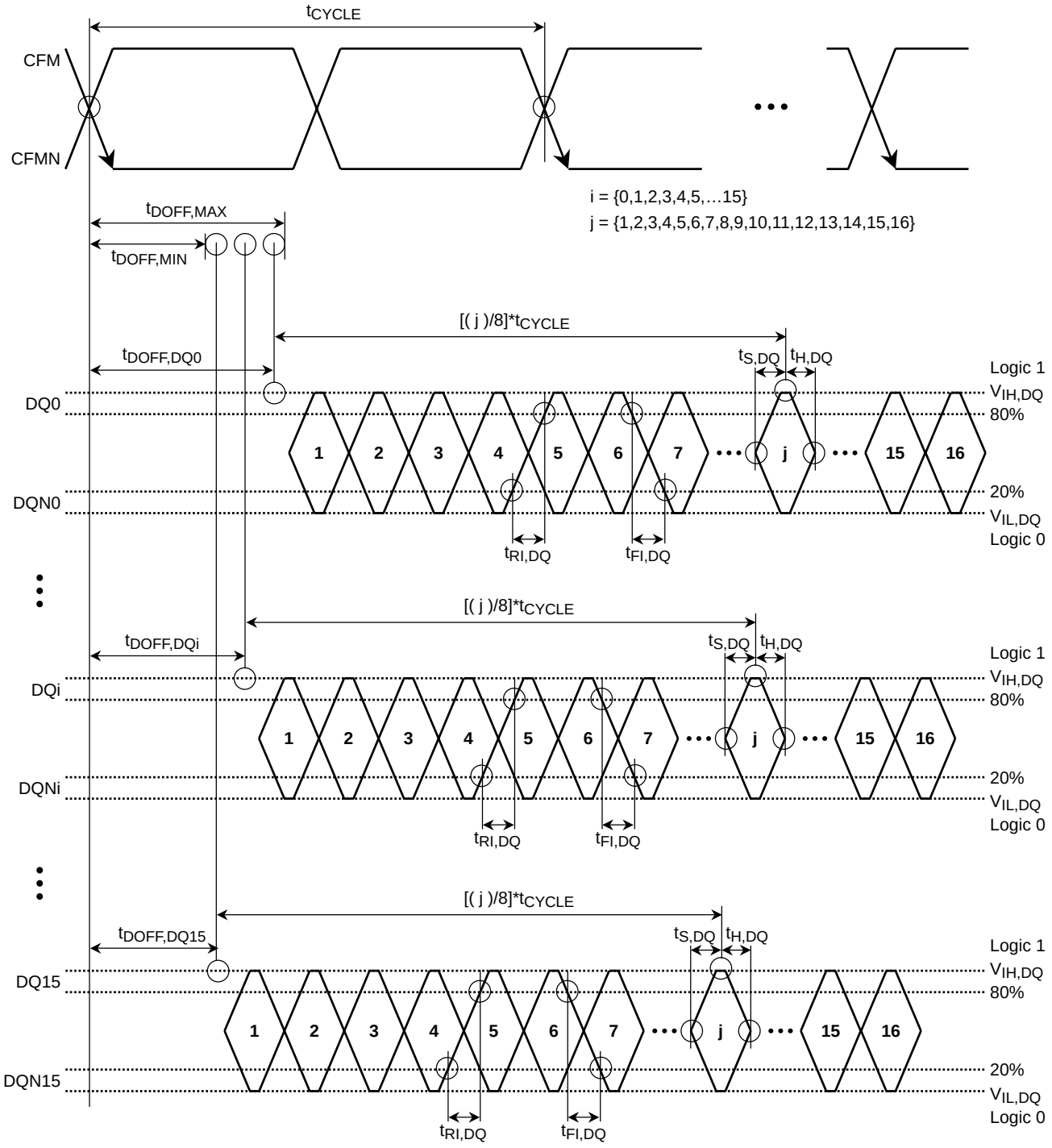
The $t_{DOFF,DQi}$ parameter determines the time between the primary CFM/CFMN crossing point and the offset point for the DQi/DQNi pin pair. The 16 receiving windows are placed at times $t_{DOFF,DQi} + (j/8) \times t_{CYCLE}$ (the index “j” may take on the values {1, 2, ..16} and refers to each of the receiving windows for the DQi/DQNi pin pair).

The offset values $t_{DOFF,DQi}$ for each of the 16 DQi/DQNi pin pairs can be different. However, each is constrained to lie inside the range $\{t_{DOFF,MIN}, t_{DOFF,MAX}\}$. Furthermore, each offset value $t_{DOFF,DQi}$ is static and will not change during system operation. Its value can be determined at initialization.

The 16 receiving windows (j = 1…16) for the first pair DQ0/DQN0 are labeled “1” through “16”. Each window has a set time ($t_{S,DQ}$) and a hold time ($t_{H,DQ}$) measured around a point $t_{DOFF,DQ0} + (j/8) \times t_{CYCLE}$ after the primary CFM/CFMN crossing point.

The 16 receiving windows (j = 1…16) for the each of the other pairs DQi/DQNi are also labeled “1” through “16”. Each window has a set time ($t_{S,DQ}$) and a hold time ($t_{H,DQ}$) measured around a point $t_{DOFF,DQi} + (j/8) \times t_{CYCLE}$ after the primary CFM/CFMN crossing point.

Figure 59. DRSL DQ Receive Waveforms



DRSL DQ Transmit Timing

Figure 60 shows a timing diagram for transmitting read data on the DQ15…DQ0/DQN15…DQN0 data pins of the memory component. This diagram represents a magnified view of these pins and only a few clock cycles are shown (CFM and CFMN are the clock signals). Timing events are measured to and from the primary CFM/CFMN crossing point in which CFM makes its high−voltage−to−low−voltage transition. The DQ15…DQ0/DQN15…DQN0 signals are high-true: a low voltage represents a logical zero and a high voltage represents a logical one. They are also differential — timing events on the DQ15…DQ0/DQN15…DQN0 pins are measured to and from the point that each differential pair crosses.

Because timing intervals are measured in this fashion, it is necessary to constrain the slew rate of the signals. The rise ($t_{OR,DQ}$) and fall time ($t_{OF,DQ}$) of the signals are measured from the 20% and 80% points of the full-swing levels.

$20\% = V_{OL,DQ} + 0.2 \times (V_{OH,DQ} - V_{OL,DQ})$

$80\% = V_{OL,DQ} + 0.8 \times (V_{OH,DQ} - V_{OL,DQ})$

There are 16 data transmit windows defined for each DQ15…DQ0/DQN15…DQN0 pin pair. The transmitting windows for a particular DQi/DQNi pin pair is referenced to an offset parameter $t_{QOFF,DQi}$ (the index “i” may take on the values {0, 1, ..15} and refers to each of the DQ15…DQ0/DQN15…DQN0 pin pairs).

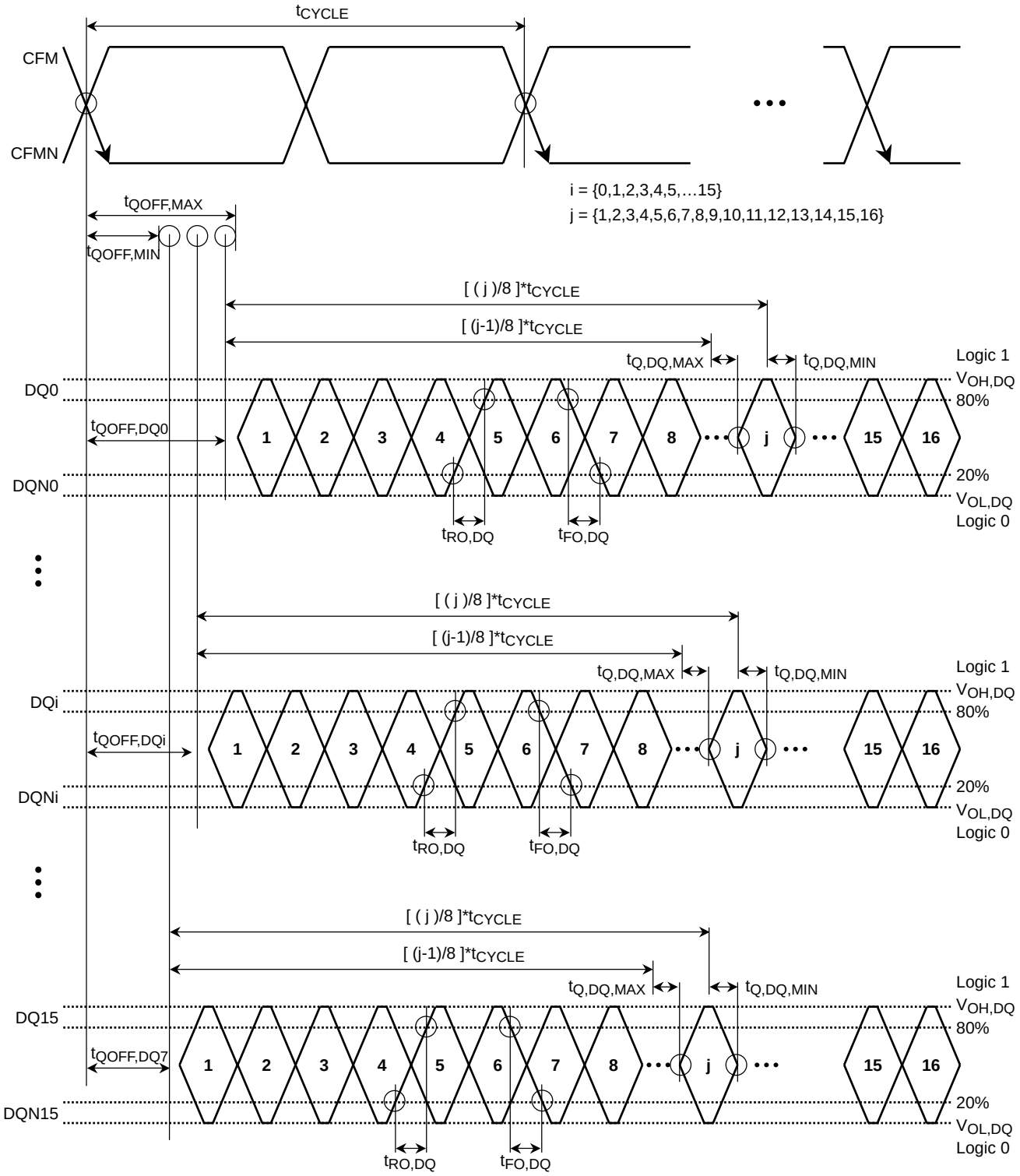
The $t_{QOFF,DQi} + t_{Q,DQ,MAX}$ expression determines the time between the primary CFM/CFMN crossing point and the offset point for the DQi/DQNi pin pair. The 16 receiving windows are placed at times $t_{QOFF,DQi} + t_{Q,DQ,MAX} + (j / 8) \times t_{CYCLE}$ (the index “j” may take on the values {1, 2, ..16} and refers to each of the transmit windows for the DQi/DQNi pin pair).

The offset values $t_{QOFF,DQi}$ for each of the 15 DQi/DQNi pin pairs can be different. However, each is constrained to lie inside the range $\{t_{QOFF,MIN}, t_{QOFF,MAX}\}$. Furthermore, each offset value $t_{QOFF,DQi}$ is static; its value will not change during system operation. Its value can be determined at initialization time.

The 16 transmit windows (j = 1…16) for the first pair DQ0/DQN0 are labeled “1” through “16”. Each window begins at the time ($t_{QOFF,DQ0} + t_{Q,DQ,MAX} + (j-1 / 8) \times t_{CYCLE}$) and ends at the time ($t_{QOFF,DQ0} + t_{Q,DQ,MIN} + ((j) / 8) \times t_{CYCLE}$) measured after the primary CFM/CFMN crossing point.

The 16 transmit windows (j=1…16) for the other pairs DQi/DQNi are also labeled “1” through “16”. Each window begins at the time ($t_{QOFF,DQi} + t_{Q,DQ,MAX} + (j/8) \times t_{CYCLE}$) and ends at the time ($t_{QOFF,DQi} + t_{Q,DQ,MIN} + ((j + 1) / 8) \times t_{CYCLE}$) measured after the primary CFM/CFMN crossing point.

Figure 60. RSL DQ Transmit Waveforms



Serial Interface Receive Timing

Figure 61 shows a timing diagram for the serial interface pins of the memory component. This diagram represents a magnified view of the pins only a few clock cycles.

The serial interface pins carry low-true signals: a high voltage represents a logical zero and a low voltage represents a logical one. Timing events are measured to and from the $V_{REF, RSL}$ level. Because timing intervals are measured in this fashion, it is necessary to constrain the slew rate of the signals. The rise time ($t_{R,SCK}$ and $t_{RI,SI}$) and fall time ($t_{F,SCK}$ and $t_{IF,SI}$) of the signals are measured from the 20% and 80% points of the full-swing levels.

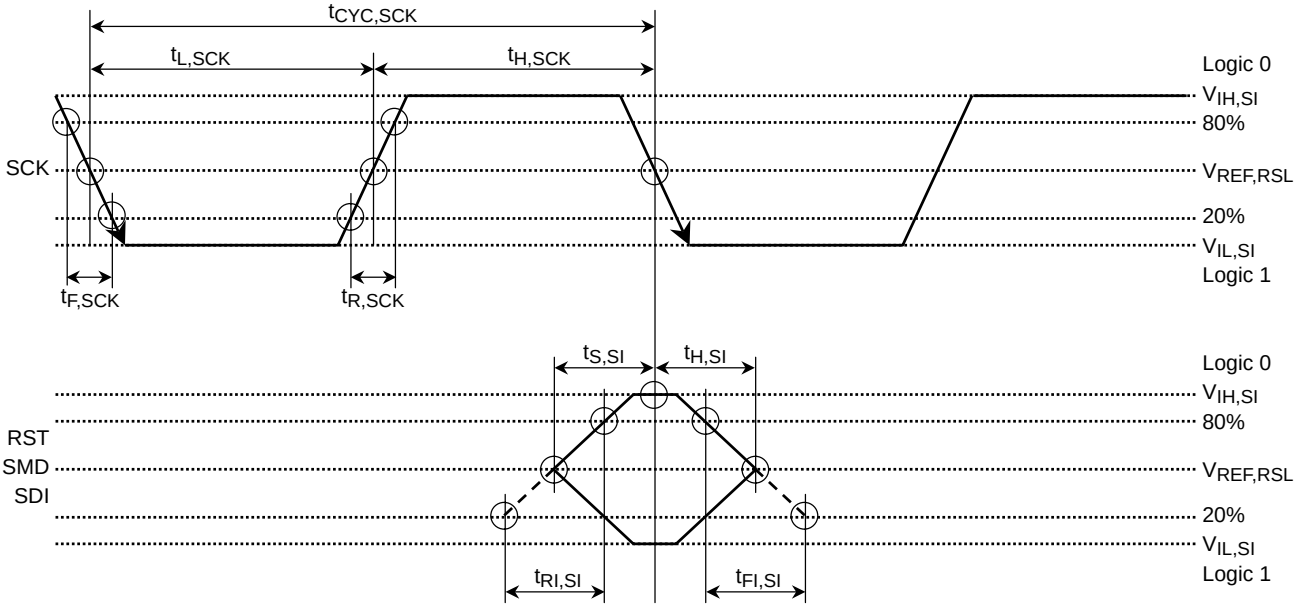
$20\% = V_{IL,\ SI} + 0.2 \times (V_{IH,SI} - V_{IL,SI})$

$50\% = V_{IL,\ SI} + 0.5 \times (V_{IH,SI} - V_{IL,SI})$

$80\% = V_{IL,\ SI} + 0.8 \times (V_{IH,SI} - V_{IL,SI})$

There is one receiving window defined for each serial interface signal (RST, CMD and SDI pins). This window has a set time ($t_{S,RQ}$) and a hold time ($t_{H,RQ}$) measured around the falling edge of the SCK clock signal.

Figure 61. Serial Interface Receive Waveforms



Serial Interface Transmit Timing

Figure 62 shows a timing diagram for the serial interface pins of the memory component. This diagram represents a magnified view of the pins and only a few clock cycles are shown.

The serial interface pins carry low-true signals: a high voltage represents a logical zero and a low voltage represents a logical one. Timing events are measured to and from the $V_{REF,\ RSL}$ level. Because timing intervals are measured in this fashion, it is necessary to constrain the slew rate of the signals. The rise time ($t_{OR,SI}$) and fall time ($t_{OF,SI}$) of the signals are measured from the 20% and 80% points of the full-swing levels.

$20\% = V_{OL,SI} + 0.2 \times (V_{OH,SI} - V_{OL,SI})$

$50\% = V_{OL,SI} + 0.5 \times (V_{OH,SI} - V_{OL,SI})$

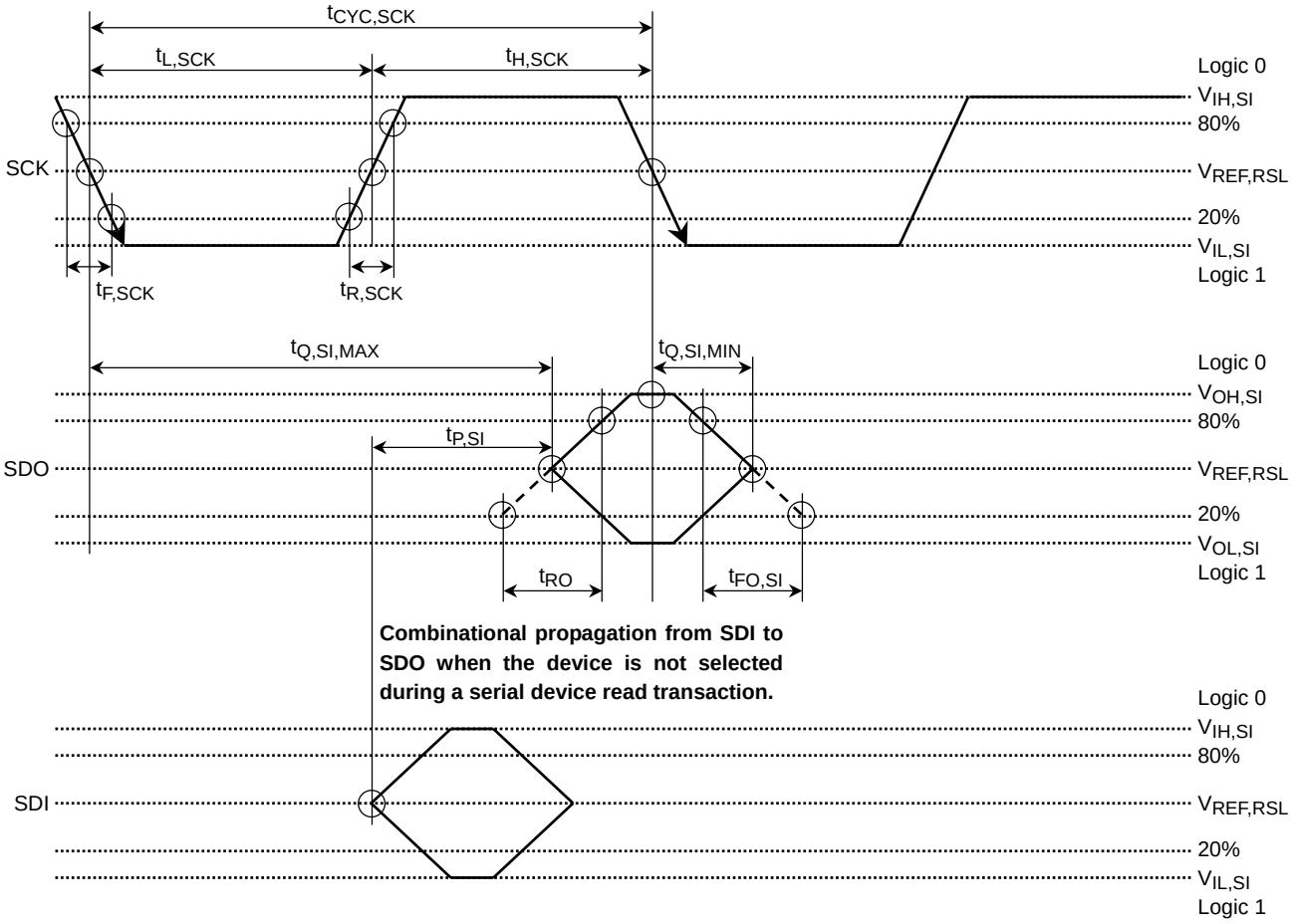
$80\% = V_{OL,SI} + 0.8 \times (V_{OH,SI} - V_{OL,SI})$

There is one transmit window defined for the serial interface data signal (SDO pins). This window has a maximum delay time ($t_{Q,SI,MAX}$) from the falling edge of the SCK clock signal and a minimum delay time ($t_{Q,SI,MIN}$) from the next falling edge of the SCK clock signal.

When the memory component is not selected during a serial device read transaction, it will simply pass the information on the SDI input to the SDO output. This combinational propagation delay parameter is $t_{P,SI}$. The $t_{CYC,SCK}$ will need to be increased during a serial read transaction (relative to the $t_{CYC,SCK}$ value for a serial write transaction) because of the accumulated propagation delay through all of the XDR DRAM devices on the serial interface.

During Initialization, when the serial identification is determined, the SDI–to–SDO path is registered, so the $t_{CYC,SCK}$ value can be set to the same value as for serial write transactions. See ”Initialization” on page 45.

Figure 62. Serial Interface Transmit Waveforms



Package Description

Package Parasitic Summary

Table 18 summarizes inductance, capacitance, and resistance values associated with each pin group for the memory component. Most of the parameters have maximum values only, however some have both maximum and minimum values.

The first group of parameters are for the CFM/CFMN clock pair pins. They include inductance, capacitance, and resistance values.

The second group of parameters are for the RQ request pins. They include inductance, mutual inductance, capacitance, and resistance values. There are also limits on the spread in inductance and capacitance values allowed in any one memory component.

The third group of parameters are specific to the DQ data pins and include inductance, mutual inductance, capacitance, and resistance values. There are also limits on the spread in inductance and capacitance values allowed in any one memory component.

The fourth group of parameters are for the serial interface pins. They include inductance and capacitance values.

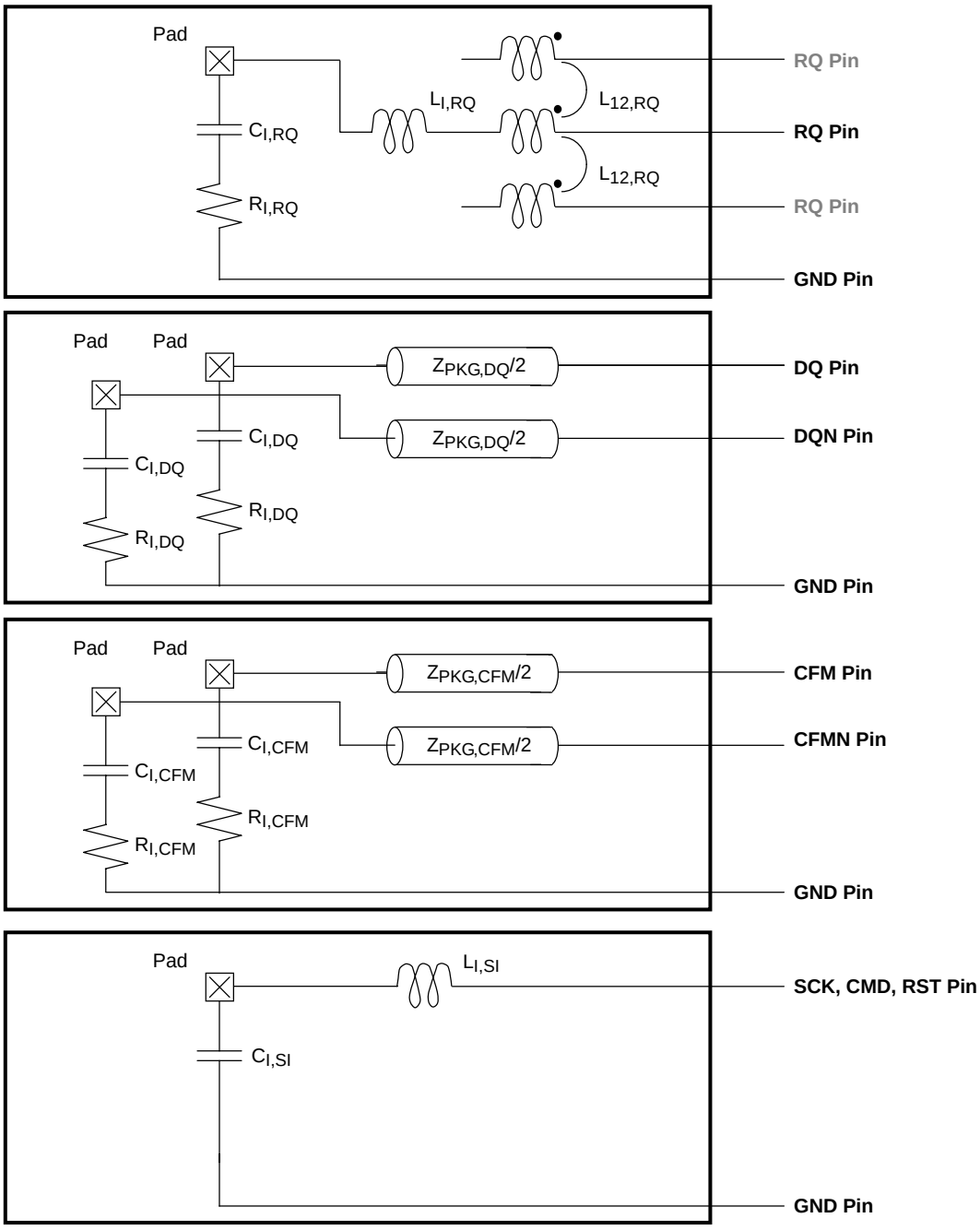
Table 18. Package RSL Parasitic Summary

Symbol	Parameter and Other Conditions		Min	Max	Units
L_{VTERM}	V_{TERM} pin – effective input inductance per for bits		—	1.2	nH
$L_{I, CFM}$	CFM/CFMN pins – effective input inductance [b]	–2400	—	TBD	nH
		–3200	—	4.0	
		–4000	—	TBD	
$C_{I, CFM}$	CFM/CFMN pins – effective input capacitance [b]	–2400	—	TBD	pF
		–3200	1.8	2.4	
		–4000	—	TBD	
$R_{I, CFM}$	CFM/CFMN pins – effective input resistance		4	15	Ω
$L_{I, RQ}$	RSL RQ pins – effective input inductance [b]	–2400	—	TBD	nH
		–3200	—	4.0	
		–4000	—	TBD	
$C_{I, RQ}$	RSL RQ pins – effective input capacitance [b]	–2400	TBD	TBD	pF
		–3200	1.8	2.4	
		–4000	TBD	TBD	
$R_{I, RQ}$	RSL RQ pins – effective input resistance		4	15	Ω
$L_{12, RQ}$	Mutual inductance between adjacent RSL RQ signals		—	0.6	nH
$\Delta L_{I, RQ}$	Difference in $L_{I, RQ}$ between any RSL RQ pins of a single device		—	1.8	nH
$\Delta C_{I, RQ}$	Difference in C_I between CFM/CFMN average and RSL RQ pins of single device		–0.06	+0.06	pF
$Z_{PKG, DQ}$	DRSL DQ pins – package differential impedance note – package trace length should be less than 10mm long	–2400	TBD	TBD	Ω
		–3200	70	130	
		–4000	TBD	TBD	
$C_{I, DQ}$	DRSL DQ pins – effective input capacitance [a]	–2400	—	TBD	pF
		–3200	—	2.0	
		–4000	—	TBD	
$\Delta C_{I, DQ}$	Difference in C_I between DQi and DQNi of each DRSL pair [a]	–2400	—	TBD	pF
		–3200	—	0.06	
		–4000	—	TBD	
$R_{I, RQ}$	DRSL DQ pins – effective input resistance		4	25	Ω
$L_{I, SI}$	Serial Interface effective input inductance [b]		—	8.0	nH
$C_{I, SI}$	Serial Interface effective input capacitance [b]	(RST, SCK, CMD)	1.7	3.0	pF
		(SDI, SDO)	—	7.0	

a. This is the effective die input capacitance, and does not include package capacitance.

b. CFM/RQ/SI should include package capacitance / Inductance, only DQ does not include package Capacitance. This value is a combination of the device IO circuitry and package capacitance and inductance.

Figure 63. Equivalent Circuits for Package Parasitic



Package Mechanical Drawing

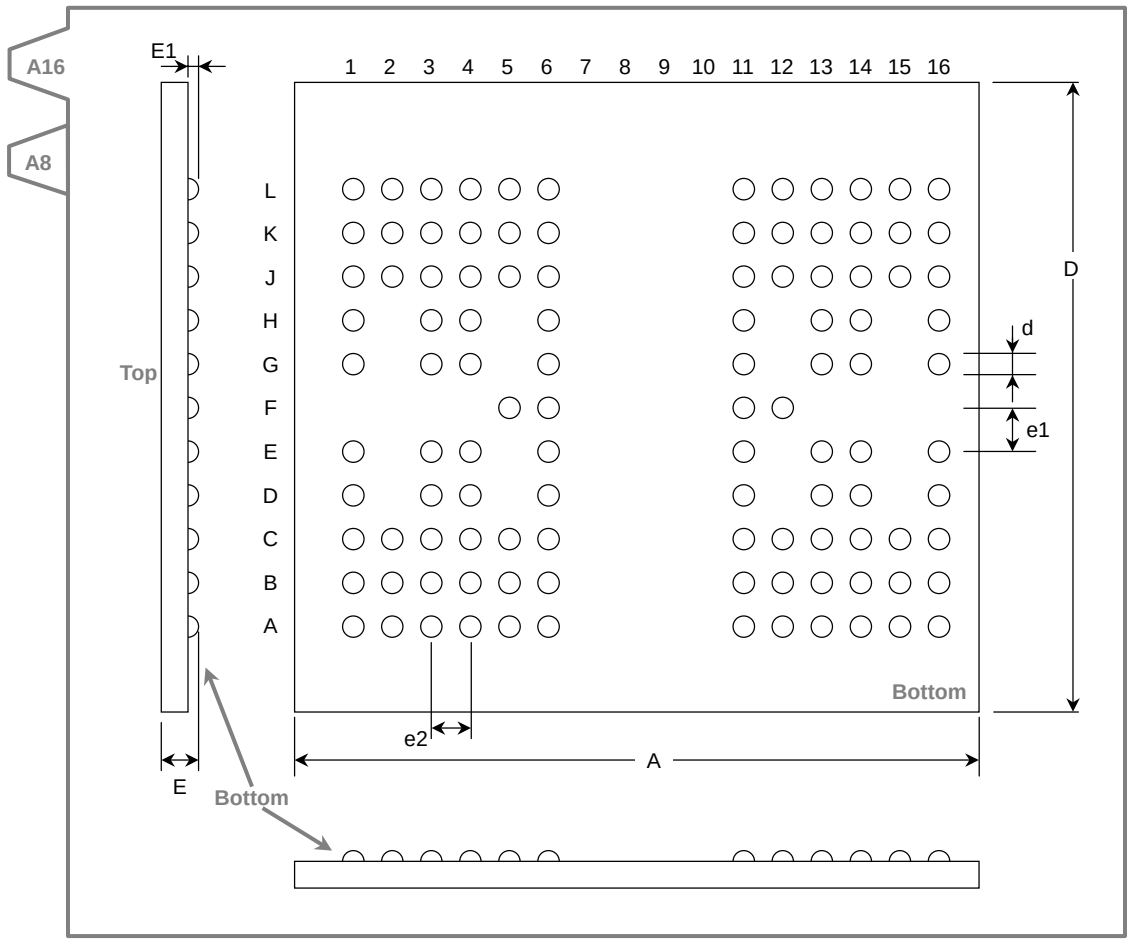
Figure 64 illustrates the x16 XDR DRAM device package and Table 19 summarizes the mechanical parameters for that package.

Table 19. CSP x16 Package Mechanical Parameters

Symbol	Parameter	Min	Max	Unit
e1	Ball pitch (x-axis)	1.27	1.27	mm
e2	Ball pitch (y-axis)	0.80	0.80	mm
A	Package body length	varies[b]	varies[b]	—
D	Package body width	varies[b]	varies[b]	—
E	Package total thickness	—	1.20[a]	mm
E1	Ball height	0.35	0.45	mm
d	Ball diameter	0.45	0.55	mm

a: The E,MAX parameter for SO-RIMM™ applications is 0.94mm.
b: Package length and width vary with die size for chip-scale package.

Figure 64. CSP x16 Package Mechanical Drawing



Rev 0.85

Package Pin Numbering

Figure 65 summarizes the device package's pin assignments.

Figure 65. CSP x16 Package - Pin Numbering (top view)

	L	K	J	H	G	F	E	D	C	B	A
1	DQN3	DQN9	VDD	GND	VDD		GND	VDD	SDI	DQN8	DQN2
2	DQ3	DQ9	VDD						GND	DQ8	DQ2
3	DQN15	DQ5	VDD	RQ10	CFM		RSRV	RQ4	RQ0	DQN4	DQN14
4	DQ15	DQN5	GND	RQ11	CFMN		RSRV	RQ3	GND	DQ4	DQ14
5	VDD	VDD	VTERM			VDD			VTERM	VDD	VDD
6	GND	GND	VTERM	GND	GND	GND	VDD	VDD	VTERM	GND	GND
7											
8											
9											
10											
11	GND	VTERN	GND	GND	VDD	GND	GND	VDD	GND	VTERM	GND
12	VDD	GND	GND			VDD			GND	GND	VDD
13	DQN7	DQN13	VDD	RQ9	RQ7		VREF	RQ1	VDD	DQN12	DQN6
14	DQ7	DQ13	CMD	RQ8	RQ6		RQ5	RQ2	GND	DQ12	DQ6
15	DQN11	DQN1	SCK						RST	DQN0	DQN10
16	DQ11	DQ1	GND	VDD	VDD		GND	VDD	SDO	DQ0	DQ10

Note:
- RSRV: Reserved pin
- DQ8…DQ15, DQN8…DQN15 are RSRV's for ×8
- DQ4…DQ15, DQN4…DQN15 are RSRV's for ×4

RESTRICTIONS ON PRODUCT USE

030619EBA

- The information contained herein is subject to change without notice.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the “Handling Guide for Semiconductor Devices,” or “TOSHIBA Semiconductor Reliability Handbook” etc..
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer’s own risk.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law and regulations.