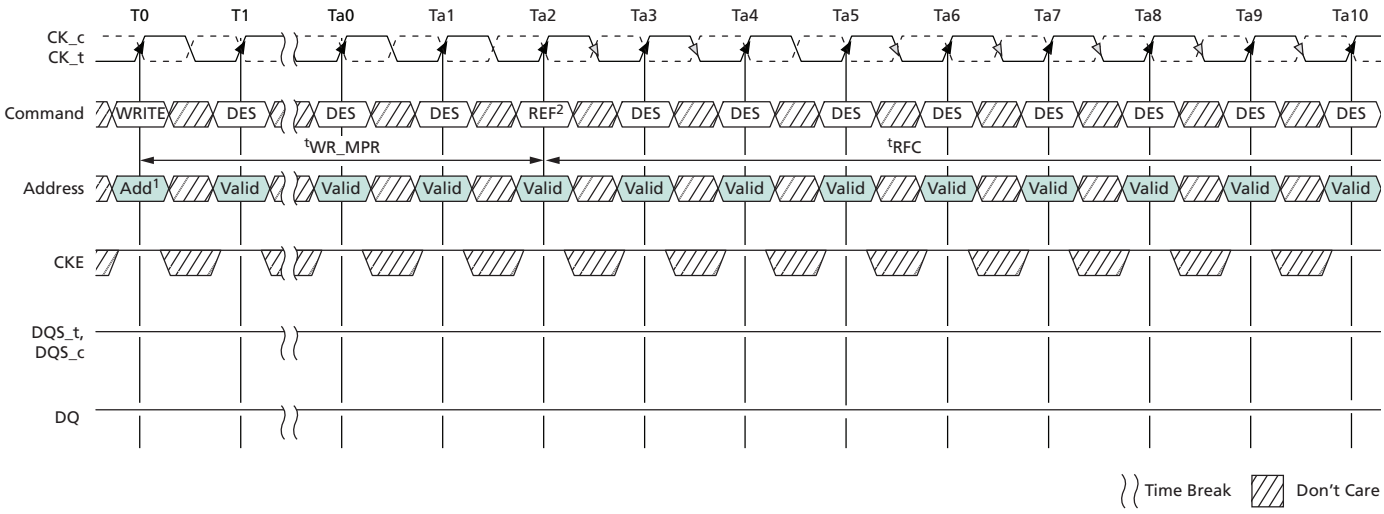
Figure 43: WRITE-to-REFRESH Timing



Notes: 1. Address setting:
BA1 and BA0 indicate the MPR location
A[7:0] = data for MPR
A10 and other address pins are "Don't Care"
2. 1x refresh is only allowed when MPR mode is enabled.

Gear-Down Mode

The DDR4 SDRAM defaults in 1/2 rate (1N) clock mode and uses a low-frequency MRS command (the MRS command has relaxed setup and hold) followed by a sync pulse (first CS pulse after MRS setting) to align the proper clock edge for operating the control lines CS_n, CKE, and ODT when in 1/4 rate (2N) mode. Gear-down mode is only supported at DDR4-2666 and faster. For operation in 1/2 rate mode, neither an MRS command or a sync pulse is required. Gear-down mode may only be entered during initialization or self refresh exit and may only be exited during self refresh exit. CAL mode and CA parity mode must be disabled prior to gear-down mode entry. The two modes may be enabled after ^{t}SYNC_GEAR and ^{t}CMD_GEAR periods have been satisfied. The general sequence for operation in 1/4 rate during initialization is as follows:

1. The device defaults to a 1N mode internal clock at power-up/reset.
2. Assertion of reset.
3. Assertion of CKE enables the DRAM.
4. MRS is accessed with a low-frequency N × ^{t}CK gear-down MRS command. (N^{t}CK static MRS command is qualified by 1N CS_n.)
5. The memory controller will send a 1N sync pulse with a low-frequency N × ^{t}CK NOP command. ^{t}SYNC_GEAR is an even number of clocks. The sync pulse is on an even edge clock boundary from the MRS command.
6. Initialization sequence, including the expiration of ^{t}DLLK and ^{t}ZQinit, starts in 2N mode after ^{t}CMD_GEAR from 1N sync pulse.

The device resets to 1N gear-down mode after entering self refresh. The general sequence for operation in gear-down after self refresh exit is as follows:

1. MRS is set to 1, via MR3[3], with a low-frequency N × ^{t}CK gear-down MRS command.
   a) The N^{t}CK static MRS command is qualified by 1N CS_n, which meets ^{t}XS or ^{t}XS_ABORT.
   b) Only a REFRESH command may be issued to the DRAM before the N^{t}CK static MRS command.
2. The DRAM controller sends a 1N sync pulse with a low-frequency N × ^{t}CK NOP command.
   a) ^{t}SYNC_GEAR is an even number of clocks.
   b) The sync pulse is on even edge clock boundary from the MRS command.
3. A valid command not requiring locked DLL is available in 2N mode after ^{t}CMD_GEAR from the 1N sync pulse.
   a) A valid command requiring locked DLL is available in 2N mode after ^{t}XSDLL or ^{t}DLLK from the 1N sync pulse.
4. If operation is in 1N mode after self refresh exit, N × ^{t}CK MRS command or sync pulse is not required during self refresh exit. The minimum exit delay to the first valid command is ^{t}XS, or ^{t}XS_ABORT.

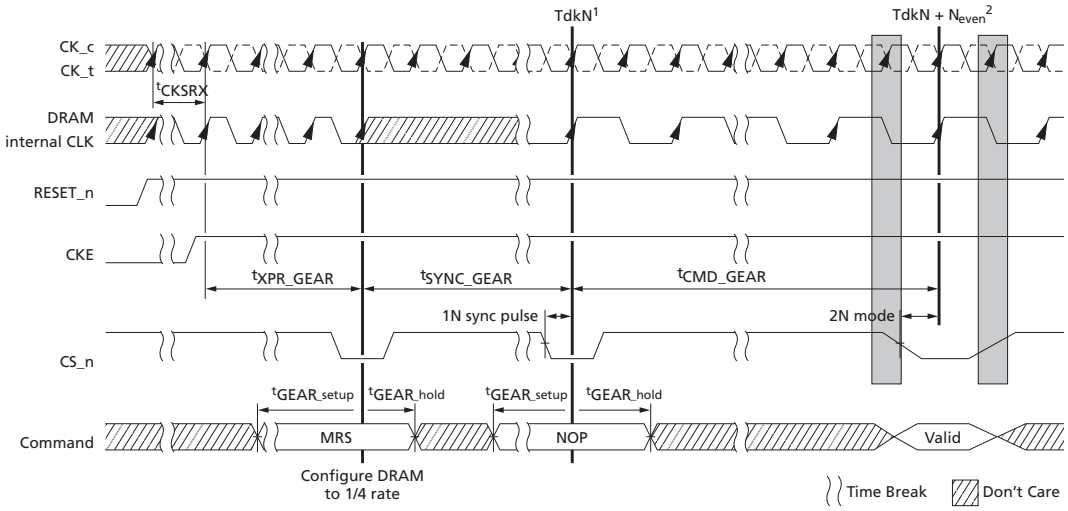
The DRAM may be changed from 2N to 1N by entering self refresh mode, which will reset to 1N mode. Changing from 2N to by any other means can result in loss of data and make operation of the DRAM uncertain.

When operating in 2N gear-down mode, the following MR settings apply:

- CAS latency (MR0[6:4,2]): Even number of clocks
- Write recovery and read to precharge (MR0[11:9]): Even number of clocks
- Additive latency (MR1[4:3]): CL - 2
- CAS WRITE latency (MR2 A[5:3]): Even number of clocks
- CS to command/address latency mode (MR4[8:6]): Even number of clocks

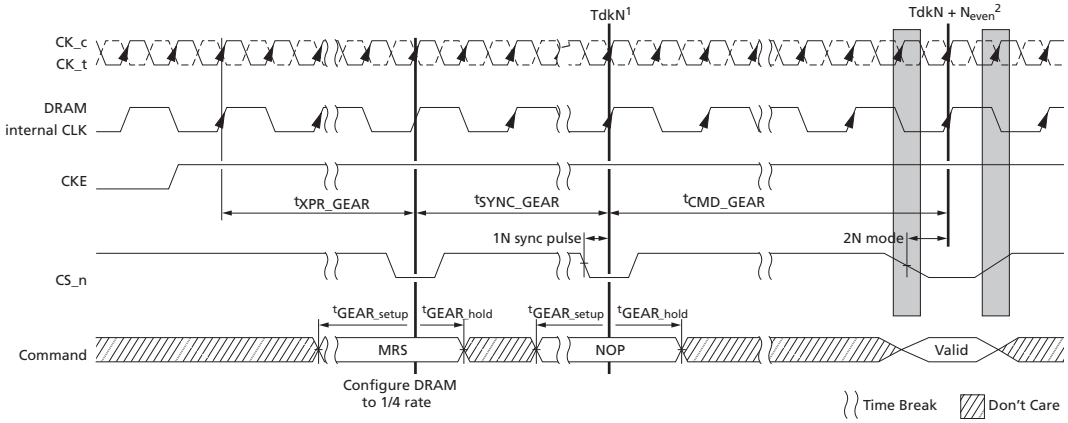
- CA parity latency mode (MR5[2:0]): Even number of clocks

Figure 44: Clock Mode Change from 1/2 Rate to 1/4 Rate (Initialization)



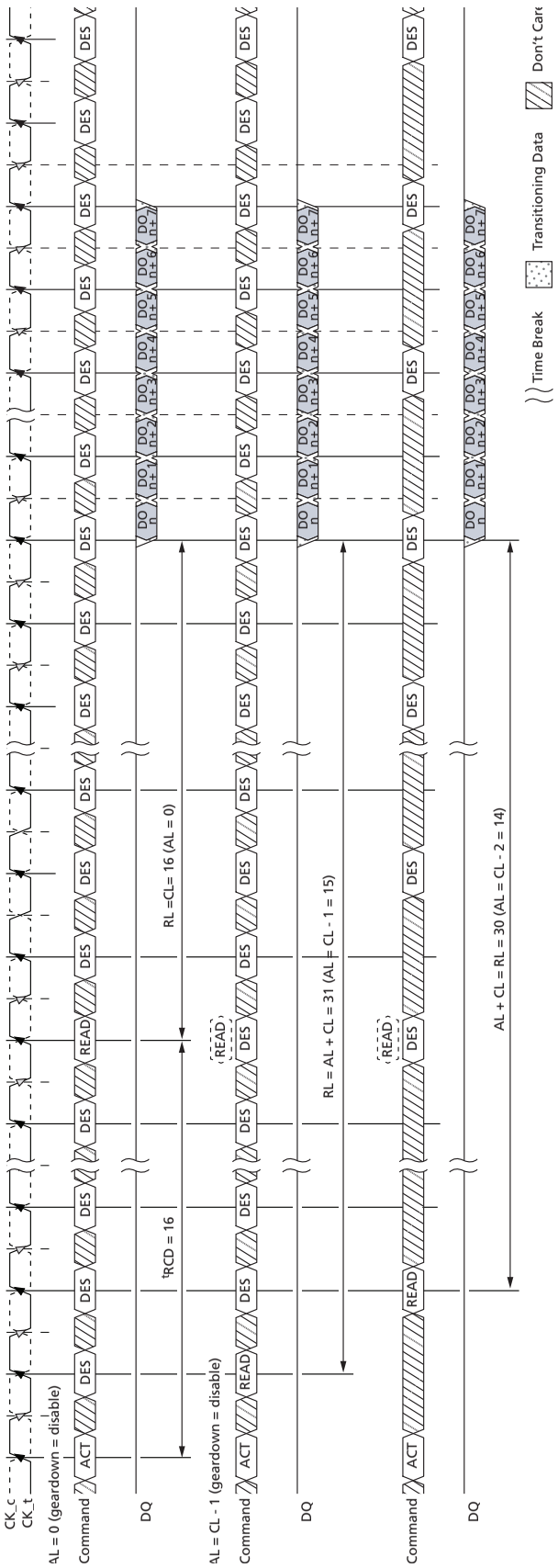
Notes: 1. After ^{t}SYNC_GEAR from GEAR-DOWN command, internal clock rate is changed at TdkN.
2. After ^{t}SYNC_GEAR + ^{t}CMD_GEAR from GEAR-DOWN command, both internal clock rate and command cycle are changed at TdkN + Neven.

Figure 45: Clock Mode Change After Exiting Self Refresh



Notes: 1. After ^{t}SYNC_GEAR from GEAR-DOWN command, internal clock rate is changed at TdkN.
2. After ^{t}SYNC_GEAR + ^{t}CMD_GEAR from GEAR-DOWN command, both internal clock rate and command cycle are changed at TdkN + Neven.

Figure 46: Comparison Between Gear-Down Disable and Gear-Down Enable



Maximum Power-Saving Mode

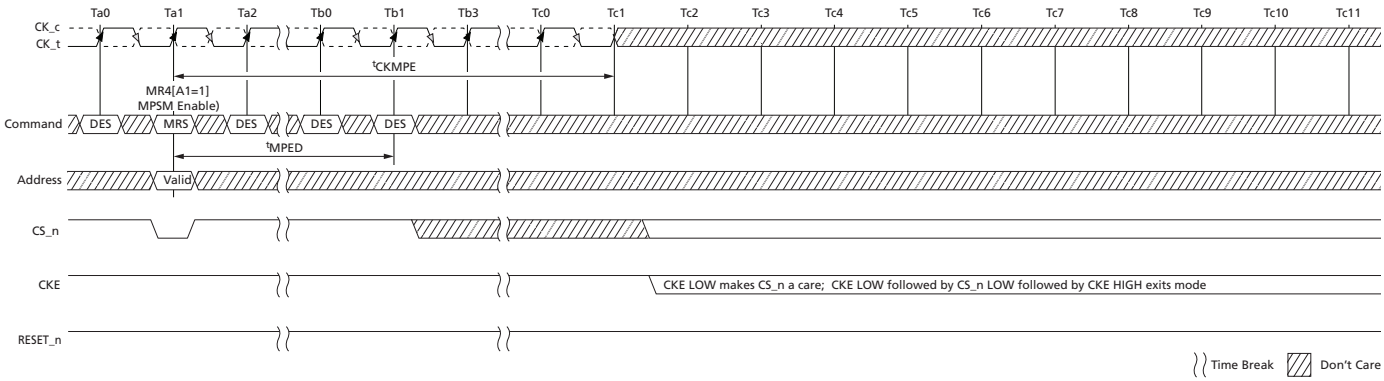
Maximum power-saving mode provides the lowest power mode where data retention is not required. When the device is in the maximum power-saving mode, it does not maintain data retention or respond to any external command, except the MAXIMUM POWER SAVING MODE EXIT command and during the assertion of RESET_n signal LOW. This mode is more like a “hibernate mode” than a typical power-saving mode. The intent is to be able to park the DRAM at a very low-power state; the device can be switched to an active state via the per-DRAM addressability (PDA) mode.

Maximum Power-Saving Mode Entry

Maximum power-saving mode is entered through an MRS command. For devices with shared control/address signals, a single DRAM device can be entered into the maximum power-saving mode using the per-DRAM addressability MRS command. Large CS_n hold time to CKE upon the mode exit could cause DRAM malfunction; as a result, CA parity, CAL, and gear-down modes must be disabled prior to the maximum power-saving mode entry MRS command.

The MRS command may use both address and DQ information, as defined in the Per-DRAM Addressability section. As illustrated in the figure below, after ^{t}MPED from the mode entry MRS command, the DRAM is not responsive to any input signals except CKE, CS_n, and RESET_n. All other inputs are disabled (external input signals may become High-Z). The system will provide a valid clock until ^{t}CKMPE expires, at which time clock inputs (CK) should be disabled (external clock signals may become High-Z).

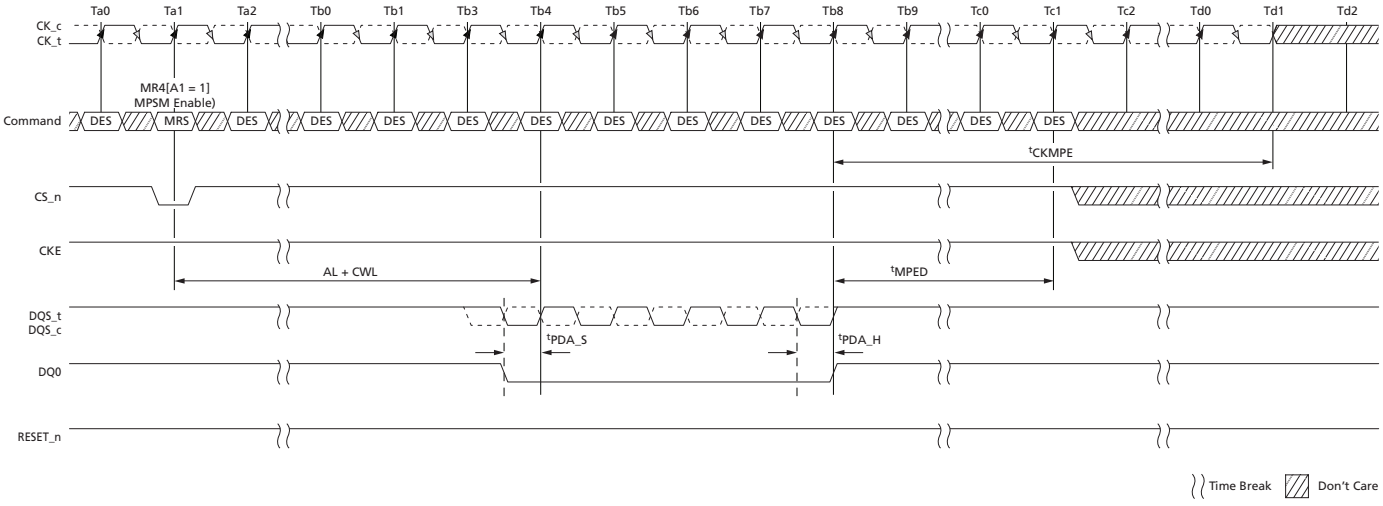
Figure 47: Maximum Power-Saving Mode Entry



Maximum Power-Saving Mode Entry in PDA

The sequence and timing required for the maximum power-saving mode with the per-DRAM addressability enabled is illustrated in the figure below.

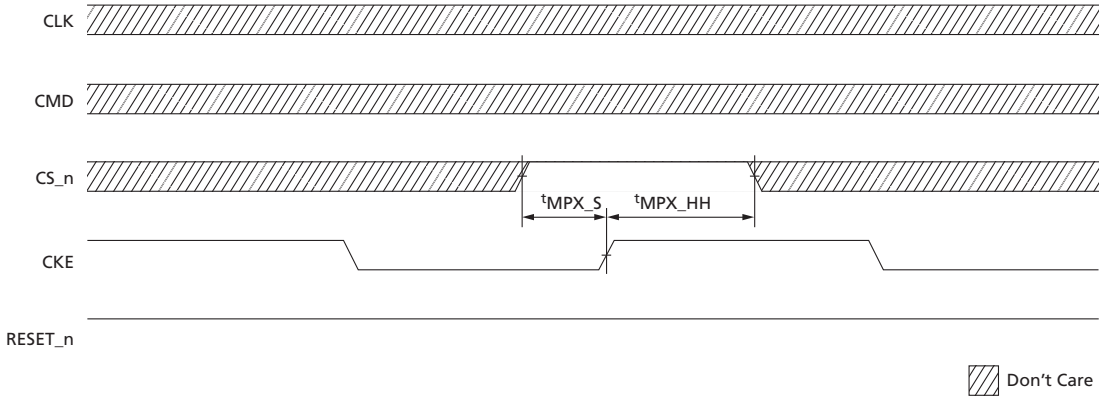
Figure 48: Maximum Power-Saving Mode Entry with PDA



CKE Transition During Maximum Power-Saving Mode

The following figure shows how to maintain maximum power-saving mode even though the CKE input may toggle. To prevent the device from exiting the mode, CS_n should be HIGH at the CKE LOW-to-HIGH edge, with appropriate setup (^{t}MPX_S) and hold (^{t}MPX_H) timings.

Figure 49: Maintaining Maximum Power-Saving Mode with CKE Transition

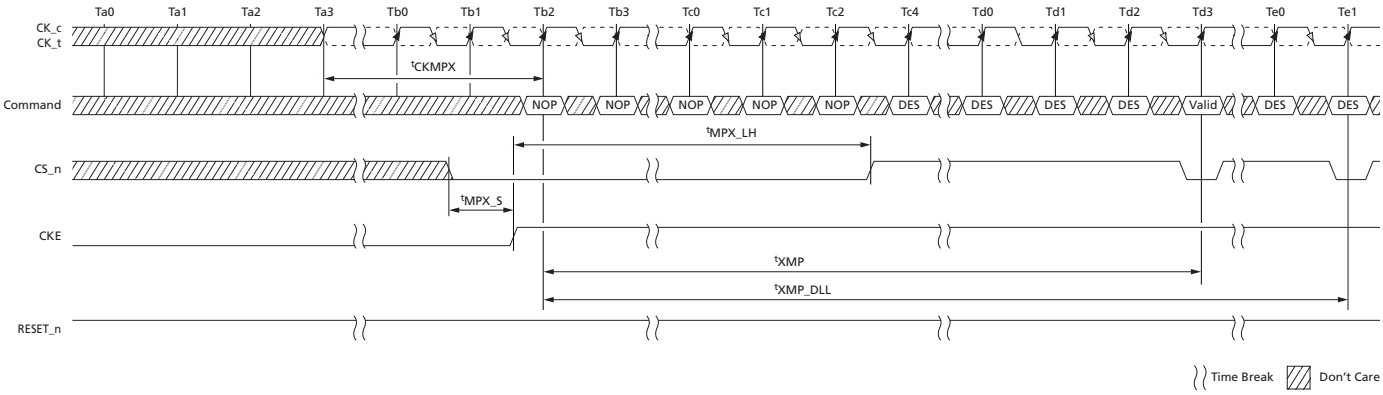


Maximum Power-Saving Mode Exit

To exit the maximum power-saving mode, CS_n should be LOW at the CKE LOW-to-HIGH transition, with appropriate setup (^{t}MPX_S) and hold (^{t}MPX_LH) timings, as shown in the figure below. Because the clock receivers (CK_t, CK_c) are disabled during this mode, CS_n = LOW is captured by the rising edge of the CKE signal. If the CS_n signal level is detected LOW, the DRAM clears the maximum power-saving mode MRS bit and begins the exit procedure from this mode. The external clock must be restarted and be stable by ^{t}CKMPX before the device can exit the maximum power-saving mode. During the exit time (^{t}XMP), only NOP and DES commands are allowed: NOP during ^{t}MPX_LH and

DES the remainder of ^{t}XMP. After ^{t}XMP expires, valid commands not requiring a locked DLL are allowed; after ^{t}XMP_DLL expires, valid commands requiring a locked DLL are allowed.

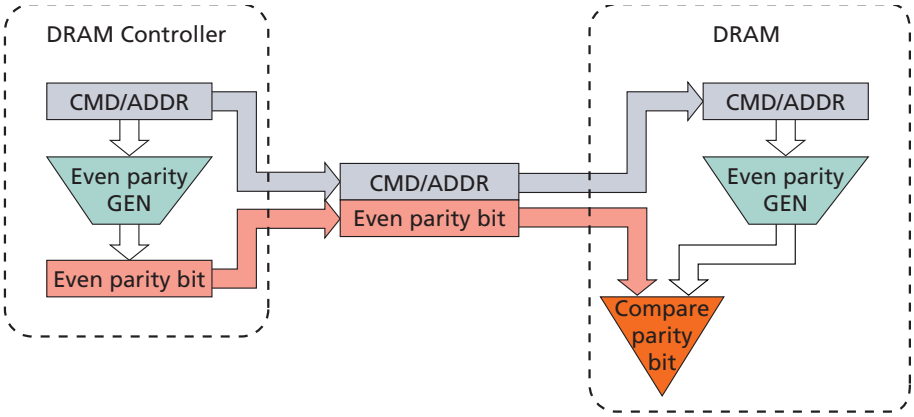
Figure 50: Maximum Power-Saving Mode Exit



Command/Address Parity

Command/address (CA) parity takes the CA parity signal (PAR) input carrying the parity bit for the generated address and commands signals and matches it to the internally generated parity from the captured address and commands signals. CA parity is supported in the DLL enabled state only; if the DLL is disabled, CA parity is not supported.

Figure 51: Command/Address Parity Operation



CA parity is disabled or enabled via an MRS command. If CA parity is enabled by programming a non-zero value to CA parity latency in the MR, the DRAM will ensure that there is no parity error before executing commands. There is an additional delay required for executing the commands versus when parity is disabled. The delay is programmed in the MR when CA parity is enabled (parity latency) and applied to all commands which are registered by CS_n (rising edge of CK_t and falling CS_n). The command is held for the time of the parity latency (PL) before it is executed inside the device. The command captured by the input clock has an internal delay before executing and is determined with PL. ALERT_n will go active when the DRAM detects a CA parity error.

CA parity covers ACT_n, RAS_n/A16, CAS_n/A15, WE_n/A14, the address bus including bank address and bank group bits, and C[2:0] on 3DS devices; the control signals CKE, ODT, and CS_n are not covered. For example, for a 4Gb x4 monolithic device, parity is computed across BG[1:0], BA[1:0], A16/RAS_n, A15/CAS_n, A14/ WE_n, A[13:0], and ACT_n. The DRAM treats any unused address pins internally as zeros; for example, if a common die has stacked pins but the device is used in a monolithic application, then the address pins used for stacking and not connected are treated internally as zeros.

The convention for parity is even parity; for example, valid parity is defined as an even number of ones across the inputs used for parity computation combined with the parity signal. In other words, the parity bit is chosen so that the total number of ones in the transmitted signal, including the parity bit, is even.

If a DRAM device detects a CA parity error in any command qualified by CS_n, it will perform the following steps:

1. Ignore the erroneous command. Commands in the MAX N*n*CK window (^{t}PAR_UNKNOWN) prior to the erroneous command are not guaranteed to be executed. When a READ command in this N*n*CK window is not executed, the device does not activate DQS outputs. If WRITE CRC is enabled and a WRITE CRC occurs during the ^{t}PAR_UNKNOWN window, the WRITE CRC Error Status Bit located at MR5[3] may or may not get set. When CA Parity and WRITE CRC are both enabled and a CA Parity occurs, the WRITE CRC Error Status Bit should be reset.
2. Log the error by storing the erroneous command and address bits in the MPR error log.

3. Set the parity error status bit in the mode register to 1. The parity error status bit must be set before the ALERT_n signal is released by the DRAM (that is, ^{t}PAR_ALERT_ON + ^{t}PAR_ALERT_PW (MIN)).
4. Assert the ALERT_n signal to the host (ALERT_n is active LOW) within ^{t}PAR_ALERT_ON time.
5. Wait for all in-progress commands to complete. These commands were received ^{t}PAR_UNKOWN before the erroneous command.
6. Wait for ^{t}RAS (MIN) before closing all the open pages. The DRAM is not executing any commands during the window defined by (^{t}PAR_ALERT_ON + ^{t}PAR_ALERT_PW).
7. After ^{t}PAR_ALERT_PW (MIN) has been satisfied, the device may de-assert ALERT_n.
   a) When the device is returned to a known precharged state, ALERT_n is allowed to be de-asserted.
8. After (^{t}PAR_ALERT_PW (MAX)) the DRAM is ready to accept commands for normal operation. Parity latency will be in effect; however, parity checking will not resume until the memory controller has cleared the parity error status bit by writing a zero. The DRAM will execute any erroneous commands until the bit is cleared; unless persistent mode is enabled.

- It is possible that the device might have ignored a REFRESH command during ^{t}PAR_ALERT_PW or the REFRESH command is the first erroneous frame, so it is recommended that extra REFRESH cycles be issued, as needed.
- The parity error status bit may be read anytime after ^{t}PAR_ALERT_ON + ^{t}PAR_ALERT_PW to determine which DRAM had the error. The device maintains the error log for the first erroneous command until the parity error status bit is reset to a zero or a second CA parity occurs prior to resetting.

The mode register for the CA parity error is defined as follows: CA parity latency bits are write only, the parity error status bit is read/write, and error logs are read-only bits. The DRAM controller can only program the parity error status bit to zero. If the DRAM controller illegally attempts to write a 1 to the parity error status bit, the DRAM can not be certain that parity will be checked; the DRAM may opt to block the DRAM controller from writing a 1 to the parity error status bit.

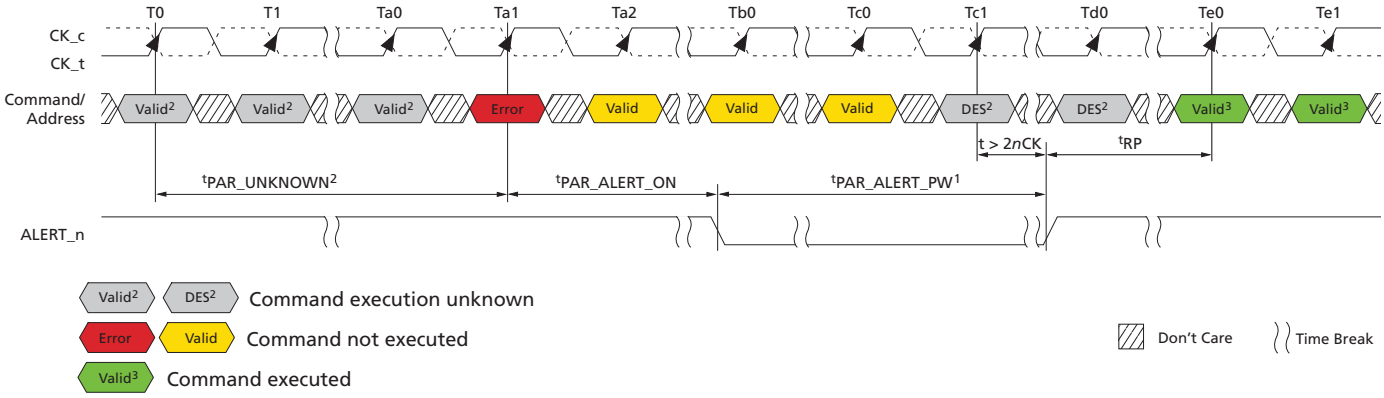
The device supports persistent parity error mode. This mode is enabled by setting MR5[9] = 1; when enabled, CA parity resumes checking after the ALERT_n is de-asserted, even if the parity error status bit remains a 1. If multiple errors occur before the error status bit is cleared the error log in MPR Page 1 should be treated as "Don't Care." In persistent parity error mode the ALERT_n pulse will be asserted and de-asserted by the DRAM as defined with the MIN and MAX value ^{t}PAR_ALERT_PW. The DRAM controller must issue DESELECT commands once it detects the ALERT_n signal, this response time is defined as ^{t}PAR_ALERT_RSP. The following figures capture the flow of events on the CA bus and the ALERT_n signal.

Table 36: Mode Register Setting for CA Parity

CA Parity Latency MR5[2:0][1]	Applicable Speed Bin	Parity Error Status	Parity Persistent Mode	Erroneous CA Frame
000 = Disabled	N/A	MR5 [4] 0 = Clear MR5 [4] 1 = Error	MR5 [9] 0 = Disabled- MR5 [9] 1 = Enabled	C[2:0], ACT_n, BG1, BG0, BA[1:0], PAR, A17, A16/RAS_n, A15/CAS_n, A14/WE_n, A[13:0]
001 = 4 clocks	1600, 1866, 2133			
010 = 5 clocks	2400, 2666			
011 = 6 clocks	2933, 3200			
100 = 8 clocks	RFU			
101 = Reserved	RFU			
110 = Reserved	RFU			
111 = Reserved	RFU			

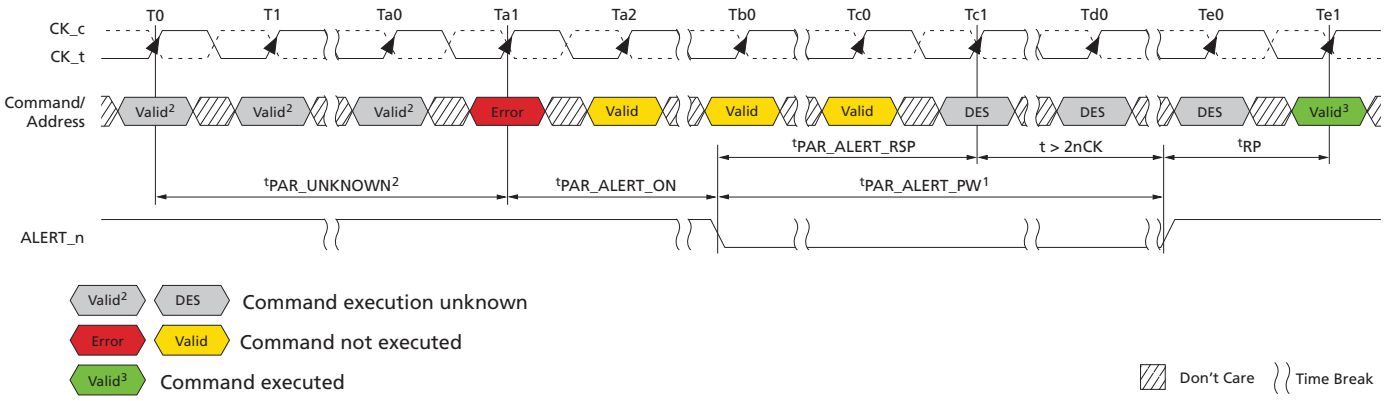
Notes: 1. Parity latency is applied to all commands.
2. Parity latency can be changed only from a CA parity disabled state; for example, a direct change from PL = 3 to PL = 4 is not allowed. The correct sequence is PL = 3 to disabled to PL = 4.
3. Parity latency is applied to WRITE and READ latency. WRITE latency = AL + CWL + PL. READ latency = AL + CL + PL.

Figure 52: Command/Address Parity During Normal Operation



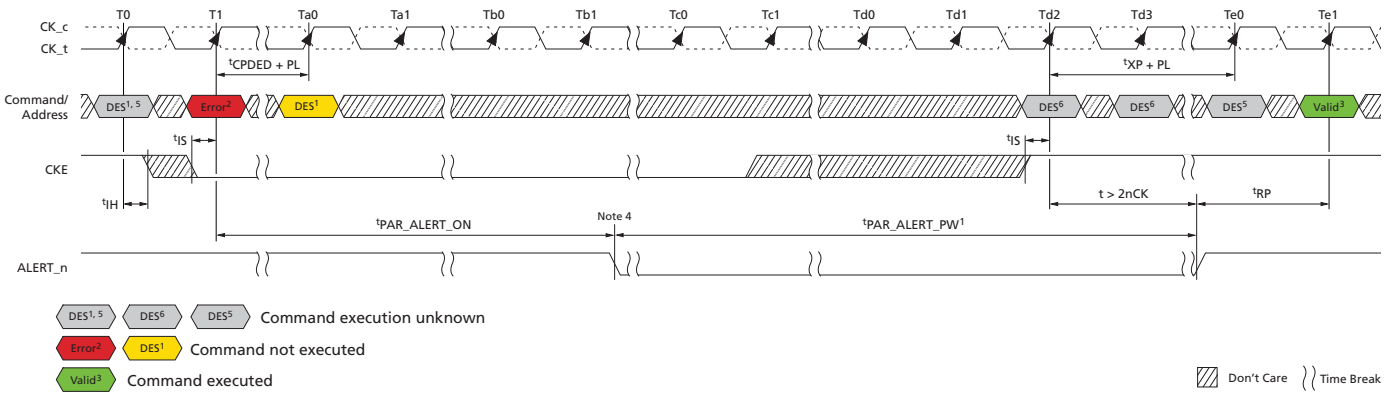
Notes: 1. DRAM is emptying queues. Precharge all and parity checking are off until parity error status bit is cleared.
2. Command execution is unknown; the corresponding DRAM internal state change may or may not occur. The DRAM controller should consider both cases and make sure that the command sequence meets the specifications.
If WRITE CRC is enabled and a WRITE CRC occurs during the tPAR_UNKNOWN window, the WRITE CRC Error Status Bit located at MR5[3] may or may not get set.
3. Normal operation with parity latency (CA parity persistent error mode disabled). Parity checking is off until parity error status bit is cleared.

Figure 53: Persistent CA Parity Error Checking Operation



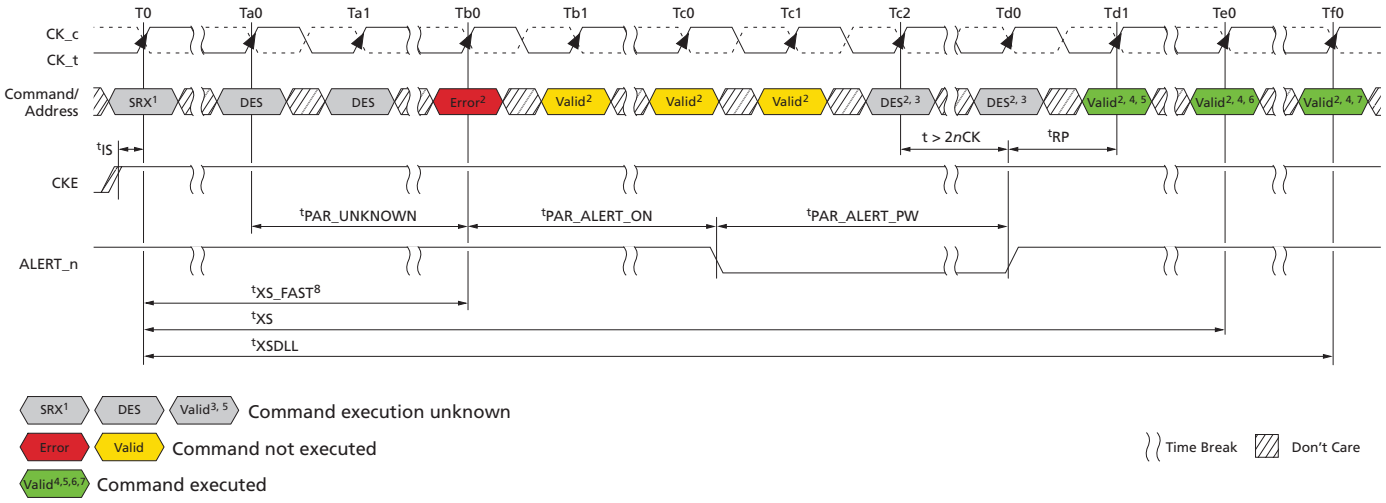
Notes: 1. DRAM is emptying queues. Precharge all and parity check re-enable finished by tPAR_ALERT_PW.
2. Command execution is unknown; the corresponding DRAM internal state change may or may not occur. The DRAM controller should consider both cases and make sure that the command sequence meets the specifications.
If WRITE CRC is enabled and a WRITE CRC occurs during the tPAR_UNKNOWN window, the WRITE CRC Error Status Bit located at MR5[3] may or may not get set
3. Normal operation with parity latency and parity checking (CA parity persistent error mode enabled).

Figure 54: CA Parity Error Checking – SRE Attempt



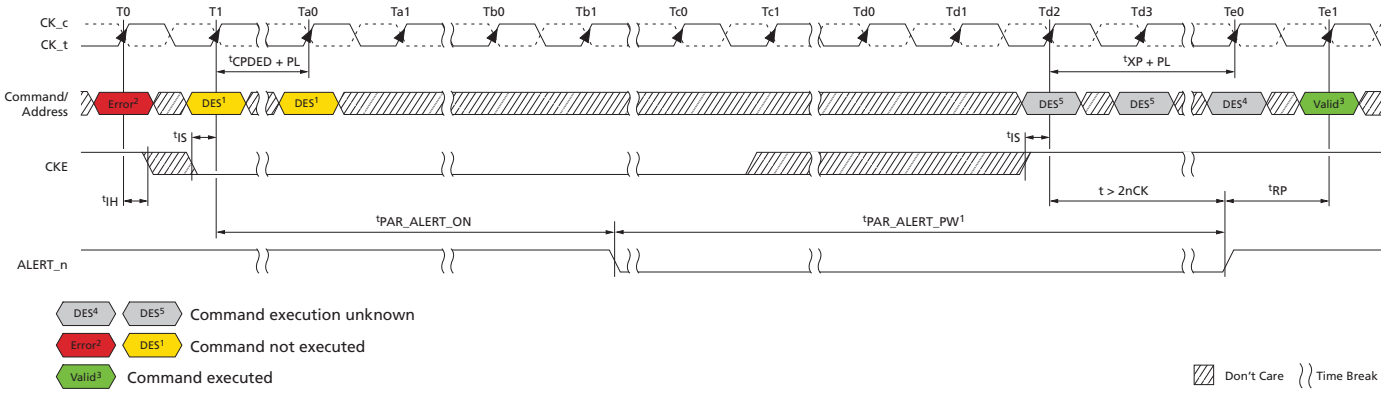
Notes: 1. Only DESELECT command is allowed.
2. SELF REFRESH command error. The DRAM masks the intended SRE command and enters precharge power-down.
3. Normal operation with parity latency (CA parity persistent error mode disabled). Parity checking is off until the parity error status bit cleared.
4. The controller cannot disable the clock until it has been capable of detecting a possible CA parity error.
5. Command execution is unknown; the corresponding DRAM internal state change may or may not occur. The DRAM controller should consider both cases and make sure that the command sequence meets the specifications.
6. Only a DESELECT command is allowed; CKE may go HIGH prior to Tc2 as long as DES commands are issued.

Figure 55: CA Parity Error Checking – SRX Attempt



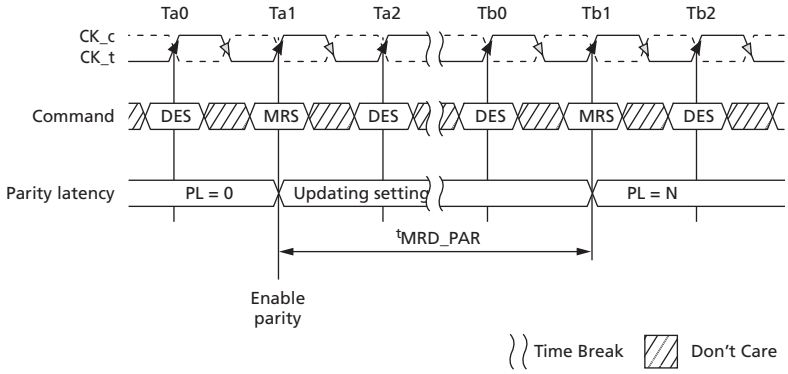
Notes: 1. Self refresh abort = disable: MR4 [9] = 0.
2. Input commands are bounded by tXSDLL, tXS, tXS_ABORT, and tXS_FAST timing.
3. Command execution is unknown; the corresponding DRAM internal state change may or may not occur. The DRAM controller should consider both cases and make sure that the command sequence meets the specifications.
4. Normal operation with parity latency (CA parity persistent error mode disabled). Parity checking off until parity error status bit cleared.
5. Only an MRS (limited to those described in the SELF REFRESH Operation section), ZQCS, or ZQCL command is allowed.
6. Valid commands not requiring a locked DLL.
7. Valid commands requiring a locked DLL.
8. This figure shows the case from which the error occurred after tXS_FAST. An error may also occur after tXS_ABORT and tXS.

Figure 56: CA Parity Error Checking – PDE/PDX



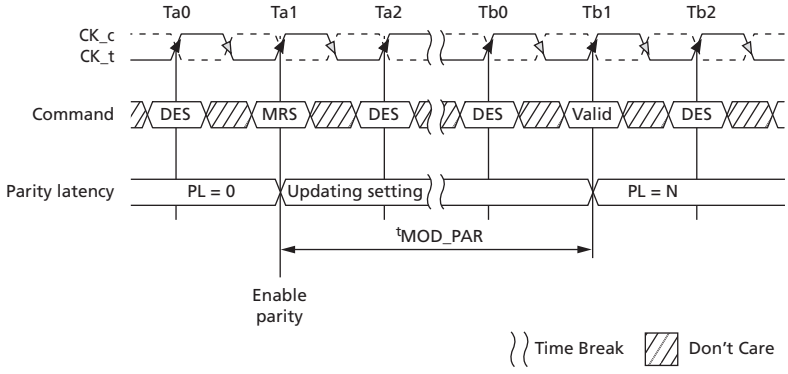
Notes: 1. Only DESELECT command is allowed.
2. Error could be precharge or activate.
3. Normal operation with parity latency (CA parity persistent error mode disabled). Parity checking is off until parity error status bit cleared.
4. Command execution is unknown; the corresponding DRAM internal state change may or may not occur. The DRAM controller should consider both cases and make sure that the command sequence meets the specifications.
5. Only a DESELECT command is allowed; CKE may go HIGH prior to Td2 as long as DES commands are issued.

Figure 57: Parity Entry Timing Example – tMRD_PAR



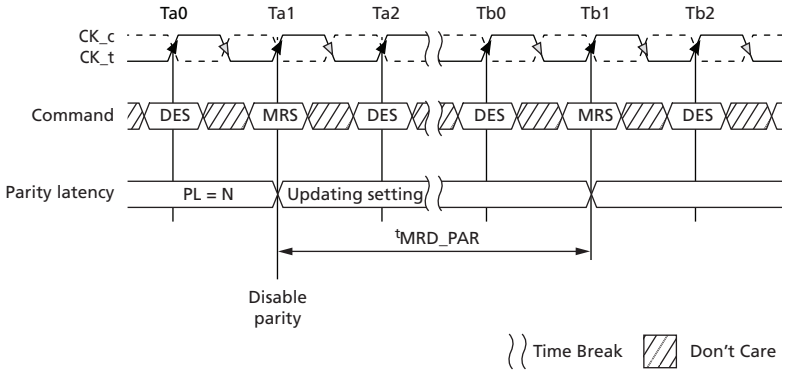
Note: 1. tMRD_PAR = tMOD + N; where N is the programmed parity latency.

Figure 58: Parity Entry Timing Example – tMOD_PAR



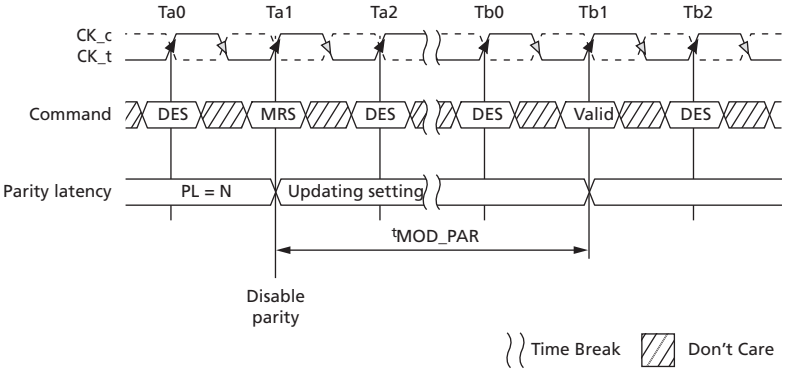
Note: 1. tMOD_PAR = tMOD + N; where N is the programmed parity latency.

Figure 59: Parity Exit Timing Example – ^{t}MRD_PAR



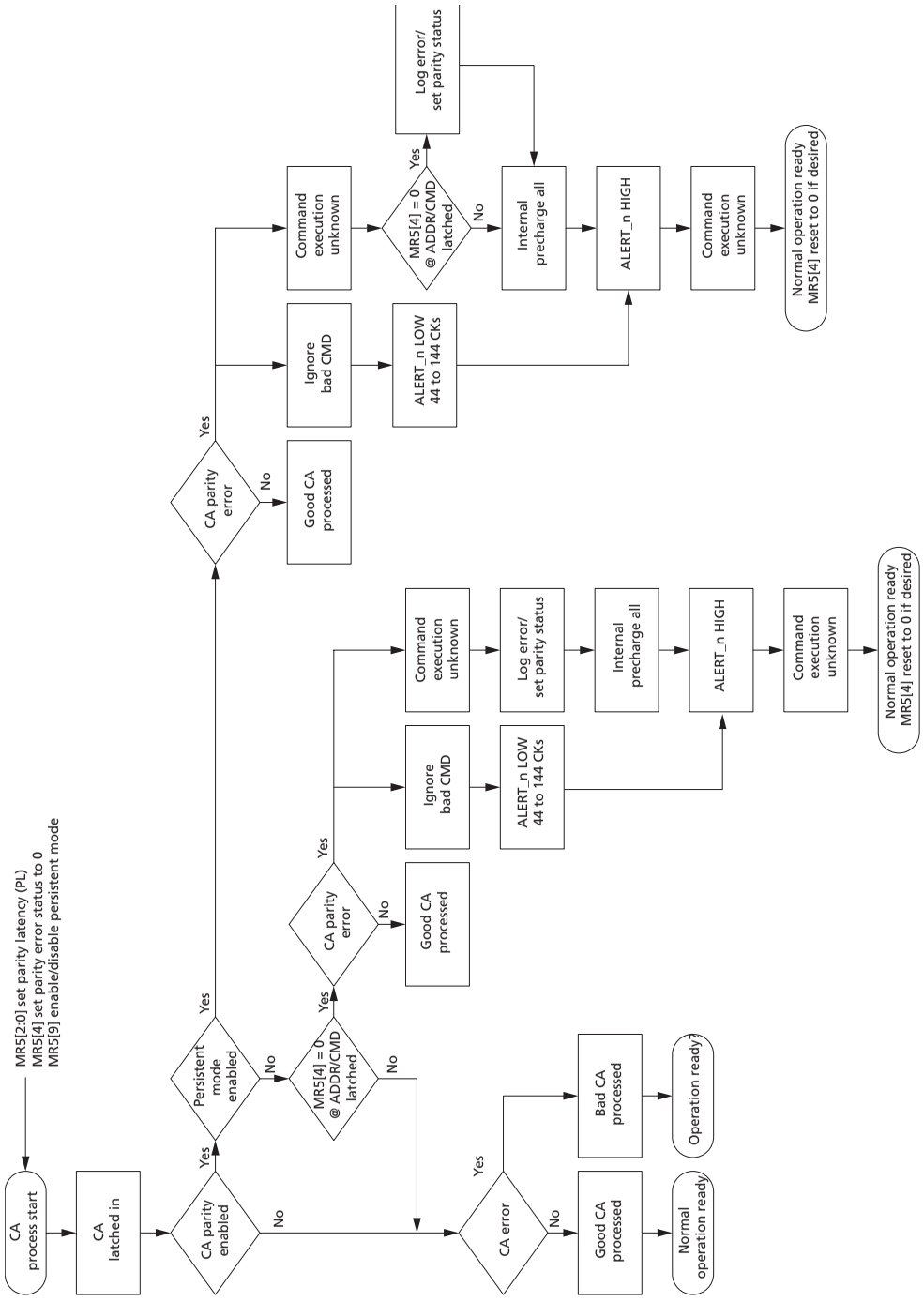
Note: 1. ^{t}MRD_PAR = ^{t}MOD + N; where N is the programmed parity latency.

Figure 60: Parity Exit Timing Example – ^{t}MOD_PAR



Note: 1. ^{t}MOD_PAR = ^{t}MOD + N; where N is the programmed parity latency.

Figure 61: CA Parity Flow Diagram



Per-DRAM Addressability

DDR4 allows programmability of a single, specific DRAM on a rank. As an example, this feature can be used to program different ODT or V_{REF} values on each DRAM on a given rank. Because per-DRAM addressability (PDA) mode may be used to program optimal V_{REF} for the DRAM, the data set up for first DQ0 transfer or the hold time for the last DQ0 transfer cannot be guaranteed. The DRAM may sample DQ0 on either the first falling or second rising DQS transfer edge. This supports a common implementation between BC4 and BL8 modes on the DRAM. The DRAM controller is required to drive DQ0 to a stable LOW or HIGH state during the length of the data transfer for BC4 and BL8 cases. Note, both fixed and on-the-fly (OTF) modes are supported for BC4 and BL8 during PDA mode.

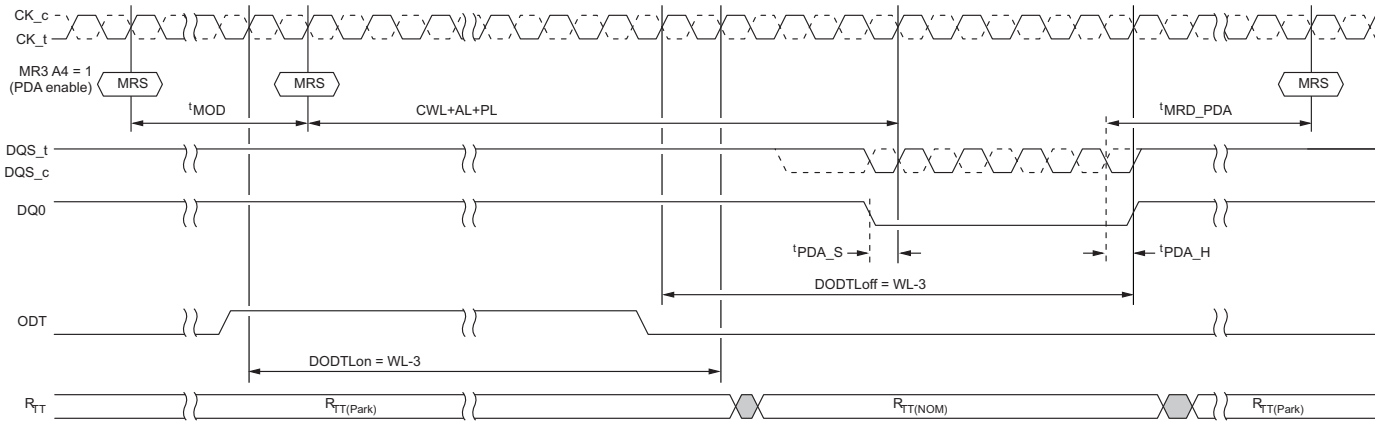
1. Before entering PDA mode, write leveling is required.
   - BL8 or BC4 may be used.
2. Before entering PDA mode, the following MR settings are possible:
   - $R_{TT(Park)}$ MR5 A[8:6] = Enable
   - $R_{TT(NOM)}$ MR1 A[10:8] = Enable
3. Enable PDA mode using MR3 [4] = 1. (The default programed value of MR3[4] = 0.)
4. In PDA mode, all MRS commands are qualified with DQ0. The device captures DQ0 by using DQS signals. If the value on DQ0 is LOW, the DRAM executes the MRS command. If the value on DQ0 is HIGH, the DRAM ignores the MRS command. The controller can choose to drive all the DQ bits.
5. Program the desired DRAM and mode registers using the MRS command and DQ0.
6. In PDA mode, only MRS commands are allowed.
7. The MODE REGISTER SET command cycle time in PDA mode, AL + CWL + BL/2 - 0.5^{t}CK + ^{t}MRD_PDA + PL, is required to complete the WRITE operation to the mode register and is the minimum time required between two MRS commands.
8. Remove the device from PDA mode by setting MR3[4] = 0. (This command requires DQ0 = 0.)

Note: Removing the device from PDA mode will require programming the entire MR3 when the MRS command is issued. This may impact some PDA values programmed within a rank as the EXIT command is sent to the rank. To avoid such a case, the PDA enable/disable control bit is located in a mode register that does not have any PDA mode controls.

In PDA mode, the device captures DQ0 using DQS signals the same as in a normal WRITE operation; however, dynamic ODT is not supported. Extra care is required for the ODT setting. If $R_{TT(NOM)}$ MR1 [10:8] = enable, device data termination needs to be controlled by the ODT pin, and applies the same timing parameters (defined below).

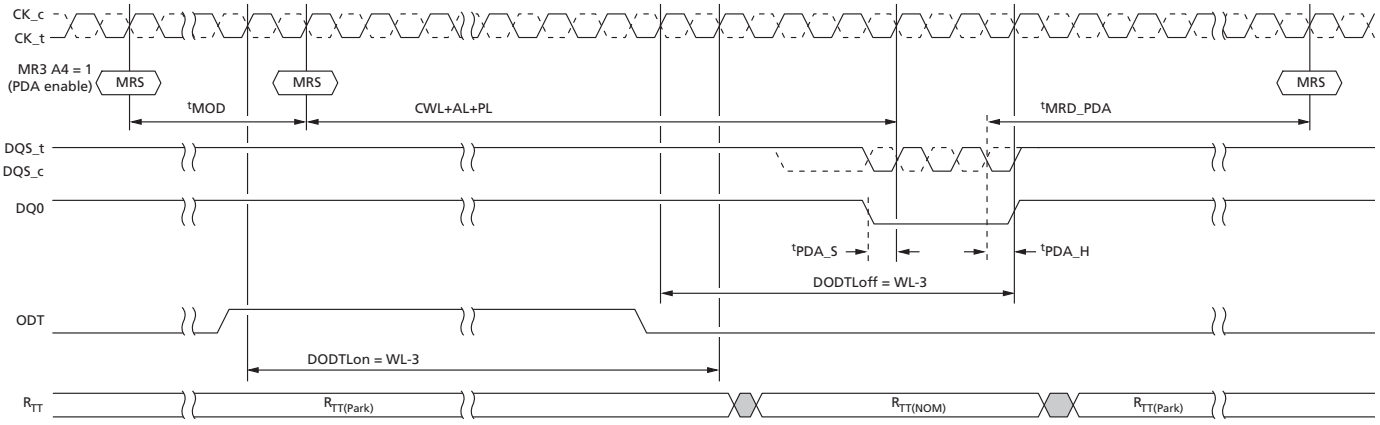
Symbol	Parameter
DODTLon	Direct ODT turnon latency
DODTLoff	Direct ODT turn off latency
^{t}ADC	R_{TT} change timing skew
^{t}AONAS	Asynchronous $R_{TT(NOM)}$ turn-on delay
^{t}AOFAS	Asynchronous $R_{TT(NOM)}$ turn-off delay

Figure 62: PDA Operation Enabled, BL8



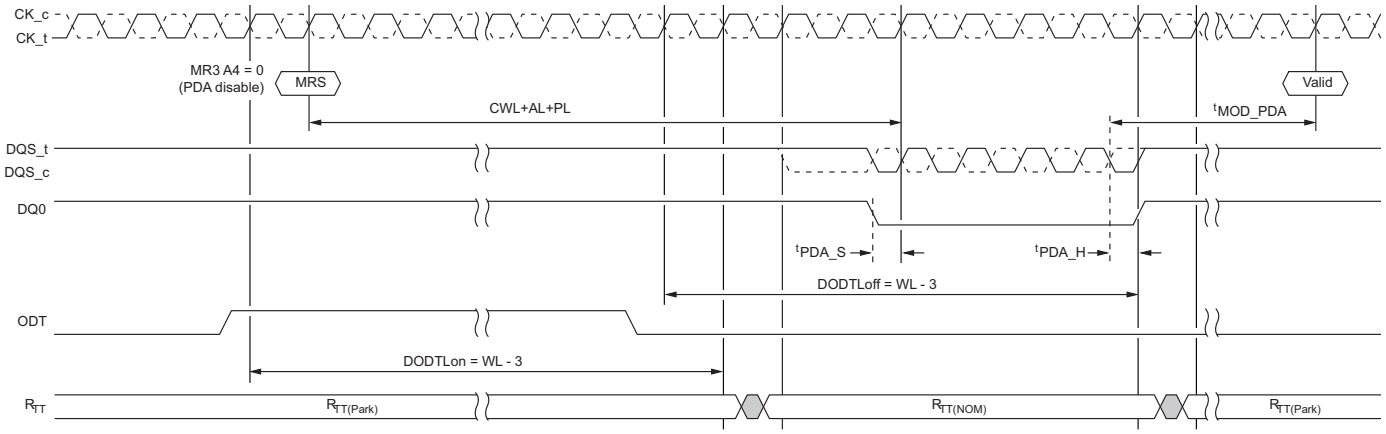
Note: 1. $R_{TT(Park)}$ = Enable; $R_{TT(NOM)}$ = Enable; WRITE preamble set = 2^{t}CK; and DLL = On.

Figure 63: PDA Operation Enabled, BC4



Note: 1. $R_{TT(Park)}$ = Enable; $R_{TT(NOM)}$ = Enable; WRITE preamble set = 2^{t}CK; and DLL = On.

Figure 64: MRS PDA Exit



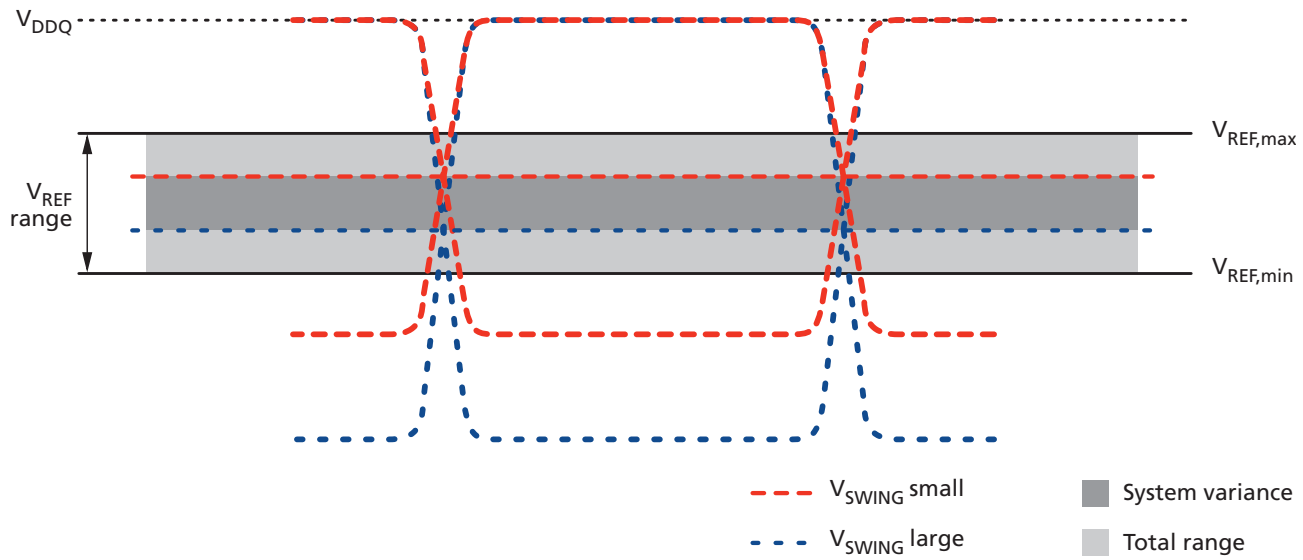
Note: 1. $R_{TT(Park)}$ = Enable; $R_{TT(NOM)}$ = Enable; WRITE preamble set = 2^{t}CK; and DLL = On.

V_{REFDQ} Calibration

The V_{REFDQ} level, which is used by the DRAM DQ input receivers, is internally generated. The DRAM V_{REFDQ} does not have a default value upon power-up and must be set to the desired value, usually via V_{REFDQ} calibration mode. If PDA or PPR modes (hPPR or sPPR) are used prior to V_{REFDQ} calibration, V_{REFDQ} should initially be set at the midpoint between the $V_{DD,max}$, and the LOW as determined by the driver and ODT termination selected with wide voltage swing on the input levels and setup and hold times of approximately 0.75UI. The memory controller is responsible for V_{REFDQ} calibration to determine the best internal V_{REFDQ} level. The V_{REFDQ} calibration is enabled/disabled via MR6[7], MR6[6] selects Range 1 (60% to 92.5% of V_{DDQ}) or Range 2 (45% to 77.5% of V_{DDQ}), and an MRS protocol using MR6[5:0] to adjust the V_{REFDQ} level up and down. MR6[6:0] bits can be altered using the MRS command if MR6[7] is enabled. The DRAM controller will likely use a series of writes and reads in conjunction with V_{REFDQ} adjustments to obtain the best V_{REFDQ}, which in turn optimizes the data eye.

The internal V_{REFDQ} specification parameters are voltage range, step size, V_{REF} step time, V_{REF} full step time, and V_{REF} valid level. The voltage operating range specifies the minimum required V_{REF} setting range for DDR4 SDRAM devices. The minimum range is defined by $V_{REFDQ,min}$ and $V_{REFDQ,max}$. As noted, a calibration sequence, determined by the DRAM controller, should be performed to adjust $V_{REFDQ}$and optimize the timing and voltage margin of the DRAM data input receivers. The internal V_{REFDQ} voltage value may not be exactly within the voltage range setting coupled with the V_{REF} set tolerance; the device must be calibrated to the correct internal V_{REFDQ} voltage.

Figure 65: V_{REFDQ} Voltage Range



V_{REFDQ} Range and Levels

Table 37: V_{REFDQ} Range and Levels

MR6[5:0]	Range 1 MR6[6] 0	Range 2 MR6[6] 1	MR6[5:0]	Range 1 MR6[6] 0	Range 2 MR6[6] 1
00 0000	60.00%	45.00%	01 1010	76.90%	61.90%
00 0001	60.65%	45.65%	01 1011	77.55%	62.55%
00 0010	61.30%	46.30%	01 1100	78.20%	63.20%
00 0011	61.95%	46.95%	01 1101	78.85%	63.85%
00 0100	62.60%	47.60%	01 1110	79.50%	64.50%
00 0101	63.25%	48.25%	01 1111	80.15%	65.15%
00 0110	63.90%	48.90%	10 0000	80.80%	65.80%
00 0111	64.55%	49.55%	10 0001	81.45%	66.45%
00 1000	65.20%	50.20%	10 0010	82.10%	67.10%
00 1001	65.85%	50.85%	10 0011	82.75%	67.75%
00 1010	66.50%	51.50%	10 0100	83.40%	68.40%
00 1011	67.15%	52.15%	10 0101	84.05%	69.05%
00 1100	67.80%	52.80%	10 0110	84.70%	69.70%
00 1101	68.45%	53.45%	10 0111	85.35%	70.35%
00 1110	69.10%	54.10%	10 1000	86.00%	71.00%
00 1111	69.75%	54.75%	10 1001	86.65%	71.65%
01 0000	70.40%	55.40%	10 1010	87.30%	72.30%
01 0001	71.05%	56.05%	10 1011	87.95%	72.95%
01 0010	71.70%	56.70%	10 1100	88.60%	73.60%
01 0011	72.35%	57.35%	10 1101	89.25%	74.25%
01 0100	73.00%	58.00%	10 1110	89.90%	74.90%
01 0101	73.65%	58.65%	10 1111	90.55%	75.55%
01 0110	74.30%	59.30%	11 0000	91.20%	76.20%
01 0111	74.95%	59.95%	11 0001	91.85%	76.85%
01 1000	75.60%	60.60%	11 0010	92.50%	77.50%
01 1001	76.25%	61.25%	11 0011 to 11 1111 = Reserved		

V_{REFDQ} Step Size

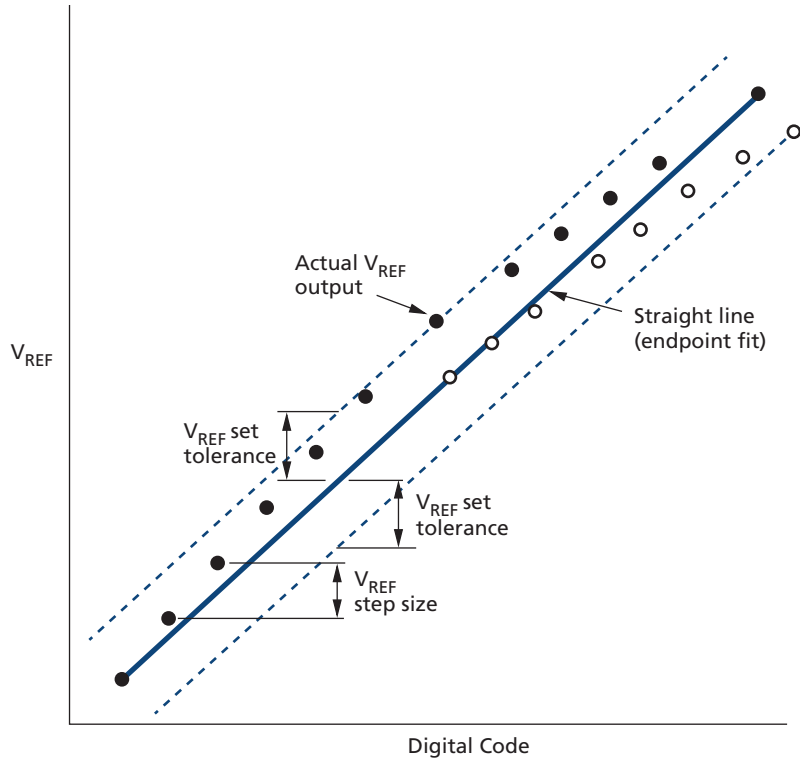
The V_{REF} step size is defined as the step size between adjacent steps. V_{REF} step size ranges from 0.5% V_{DDQ} to 0.8% V_{DDQ}. However, for a given design, the device has one value for V_{REF} step size that falls within the range.

The V_{REF} set tolerance is the variation in the V_{REF} voltage from the ideal setting. This accounts for accumulated error over multiple steps. There are two ranges for V_{REF} set tolerance uncertainty. The range of V_{REF} set tolerance uncertainty is a function of number of steps *n*.

The V_{REF} set tolerance is measured with respect to the ideal line, which is based on the MIN and MAX V_{REF} value endpoints for a specified range. The internal V_{REFDQ} voltage value may not be exactly within

the voltage range setting coupled with the V_{REF} set tolerance; the device must be calibrated to the correct internal V_{REFDQ} voltage.

Figure 66: Example of V_{REF} Set Tolerance and Step Size

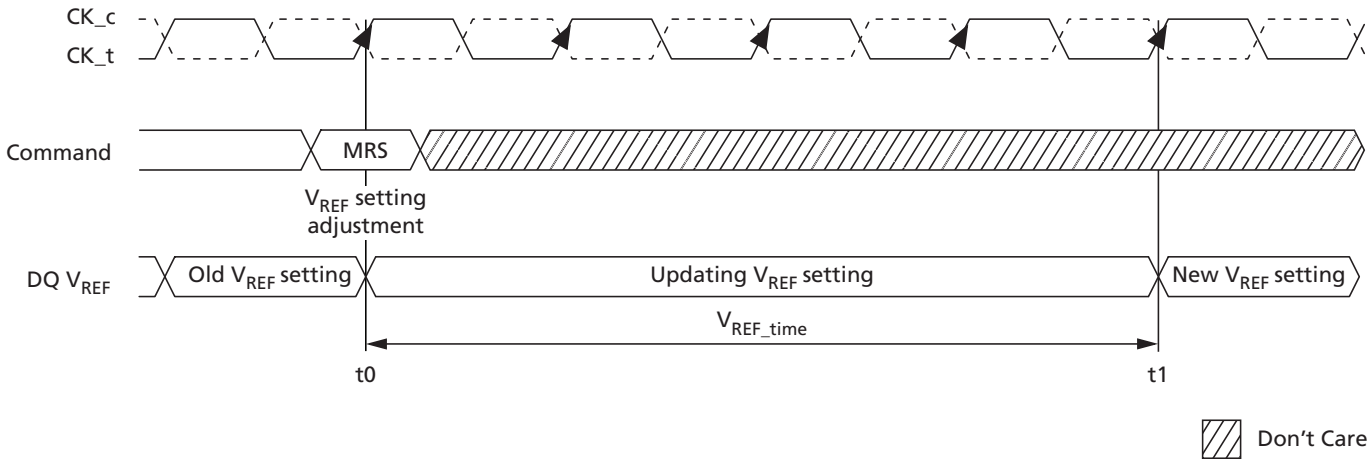


Note: 1. Maximum case shown.

V_{REFDQ} Increment and Decrement Timing

The V_{REF} increment/decrement step times are defined by $V_{REF,time}$. $V_{REF,time}$ is defined from t0 to t1, where t1 is referenced to the V_{REF} voltage at the final DC level within the V_{REF} valid tolerance (V_{REF,val_tol}). The V_{REF} valid level is defined by $V_{REF,val}$ tolerance to qualify the step time t1. This parameter is used to insure an adequate RC time constant behavior of the voltage level change after any V_{REF} increment/decrement adjustment.

Figure 67: V_{REFDQ} Timing Diagram for $V_{REF,time}$ Parameter



Note: 1. t0 is referenced to the MRS command clock
t1 is referenced to $V_{REF,tol}$

V_{REFDQ} calibration mode is entered via an MRS command, setting MR6[7] to 1 (0 disables V_{REFDQ} calibration mode) and setting MR6[6] to either 0 or 1 to select the desired range (MR6[5:0] are "Don't Care"). After V_{REFDQ} calibration mode has been entered, V_{REFDQ} calibration mode legal commands may be issued once ^{t}VREFDQE has been satisfied. Legal commands for V_{REFDQ} calibration mode are ACT, WR, WRA, RD, RDA, PRE, DES, and MRS to set V_{REFDQ} values, and MRS to exit V_{REFDQ} calibration mode. Also, after V_{REFDQ} calibration mode has been entered, "dummy" WRITE commands are allowed prior to adjusting the V_{REFDQ} value the first time V_{REFDQ} calibration is performed after initialization.

Setting V_{REFDQ} values requires MR6[7] be set to 1 and MR6[6] be unchanged from the initial range selection; MR6[5:0] may be set to the desired V_{REFDQ} values. If MR6[7] is set to 0, MR6[6:0] are not written. $V_{REF,time-short}$ or $V_{REF,time-long}$ must be satisfied after each MR6 command to set V_{REFDQ} value before the internal V_{REFDQ} value is valid.

If PDA mode is used in conjunction with V_{REFDQ} calibration, the PDA mode requirement that only MRS commands are allowed while PDA mode is enabled is not waived. That is, the only V_{REFDQ} calibration mode legal commands noted above that may be used are the MRS commands: MRS to set V_{REFDQ} values and MRS to exit V_{REFDQ} calibration mode.

The last MR6[6:0] setting written to MR6 prior to exiting V_{REFDQ} calibration mode is the range and value used for the internal V_{REFDQ} setting. V_{REFDQ} calibration mode may be exited when the DRAM is in idle state. After the MRS command to exit V_{REFDQ} calibration mode has been issued, DES must be issued until ^{t}VREFDQX has been satisfied where any legal command may then be issued. V_{REFDQ} setting should be updated if the die temperature changes too much from the calibration temperature.

The following are typical script when applying the above rules for V_{REFDQ} calibration routine when performing V_{REFDQ} calibration in Range 1:

- MR6[7:6]10 [5:0]XXXXXXX.
  - Subsequent legal commands while in V_{REFDQ} calibration mode: ACT, WR, WRA, RD, RDA, PRE, DES, and MRS (to set V_{REFDQ} values and exit V_{REFDQ} calibration mode).
- All subsequent V_{REFDQ} calibration MR setting commands are MR6[7:6]10 [5:0]VVVVVV.
  - "VVVVVV" are desired settings for V_{REFDQ}.

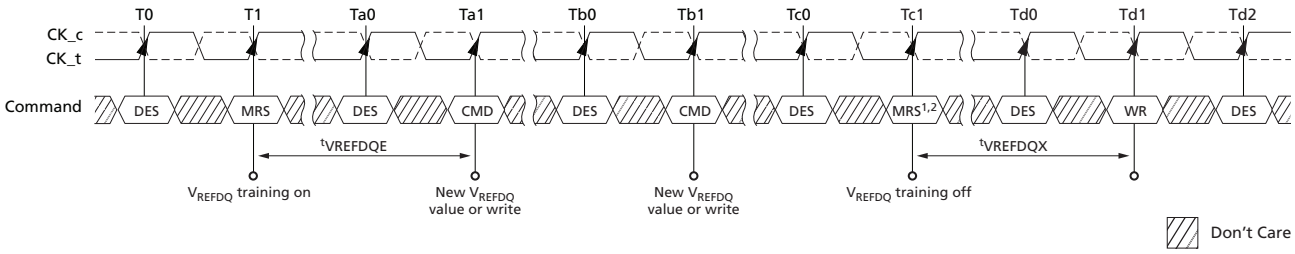
- Issue ACT/WR/RD looking for pass/fail to determine V_{CENT} (midpoint) as needed.
- To exit V_{REFDQ} calibration, the last two V_{REFDQ} calibration MR commands are:
  - MR6[7:6]10 [5:0]VVVVVV* where VVVVVV* = desired value for V_{REFDQ}.
  - MR6[7]0 [6:0]XXXXXXX to exit V_{REFDQ} calibration mode.

The following are typical script when applying the above rules for V_{REFDQ} calibration routine when performing V_{REFDQ} calibration in Range 2:

- MR6[7:6]11 [5:0]XXXXXXX.
  - Subsequent legal commands while in V_{REFDQ} calibration mode: ACT, WR, WRA, RD, RDA, PRE, DES, and MRS (to set V_{REFDQ} values and exit V_{REFDQ} calibration mode).
- All subsequent V_{REFDQ} calibration MR setting commands are MR6[7:6]11 [5:0]VVVVVV.
  - "VVVVVV" are desired settings for V_{REFDQ}.
- Issue ACT/WR/RD looking for pass/fail to determine V_{CENT} (midpoint) as needed.
- To exit V_{REFDQ} calibration, the last two V_{REFDQ} calibration MR commands are:
  - MR6[7:6]11 [5:0]VVVVVV* where VVVVVV* = desired value for V_{REFDQ}.
  - MR6[7]0 [6:0]XXXXXXX to exit V_{REFDQ} calibration mode.

Note: Range may only be set or changed when entering V_{REFDQ} calibration mode; changing range while in or exiting V_{REFDQ} calibration mode is illegal.

Figure 68: V_{REFDQ} Training Mode Entry and Exit Timing Diagram



Notes: 1. New V_{REFDQ} values are not allowed with an MRS command during calibration mode entry.
2. Depending on the step size of the latest programmed V_{REF} value, V_{REF} must be satisfied before disabling V_{REFDQ} training mode.

Figure 69: V_{REF} Step: Single Step Size Increment Case

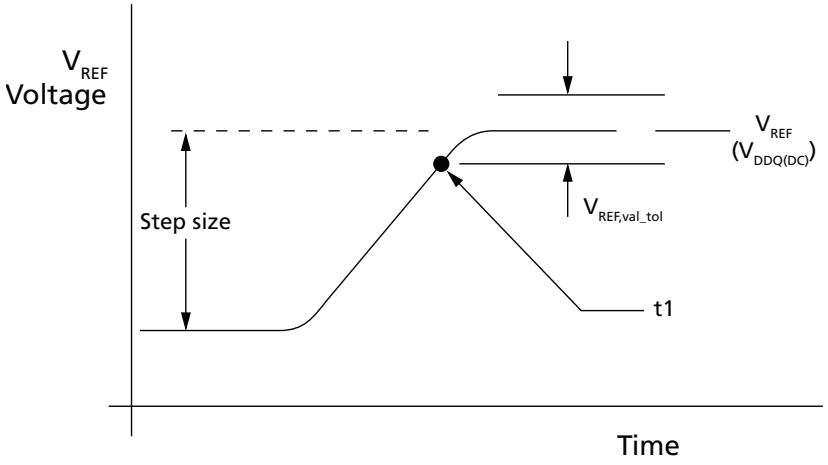


Figure 70: V_{REF} Step: Single Step Size Decrement Case

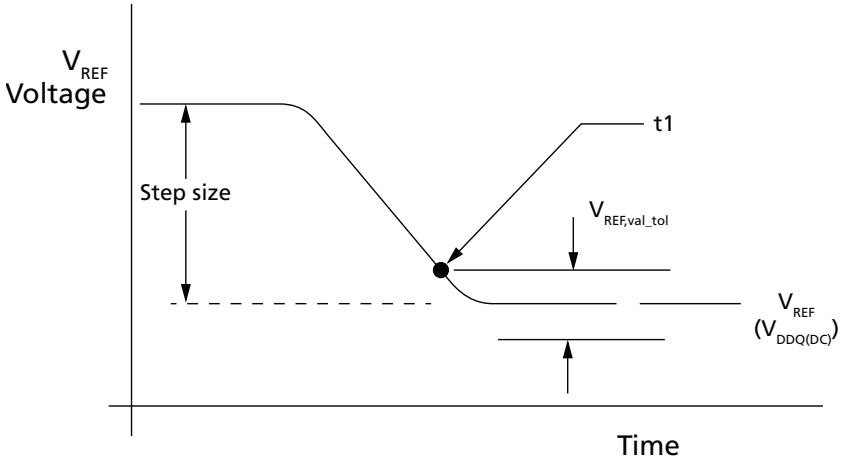


Figure 71: V_{REF} Full Step: From $V_{REF,min}$ to $V_{REF,max}$Case

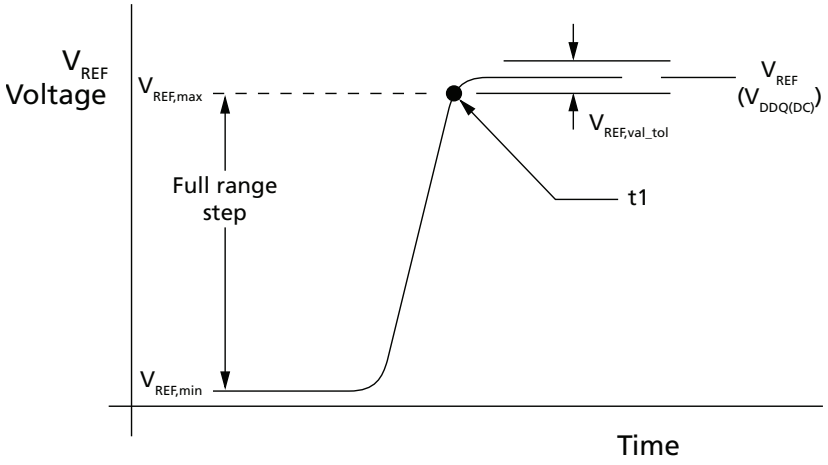
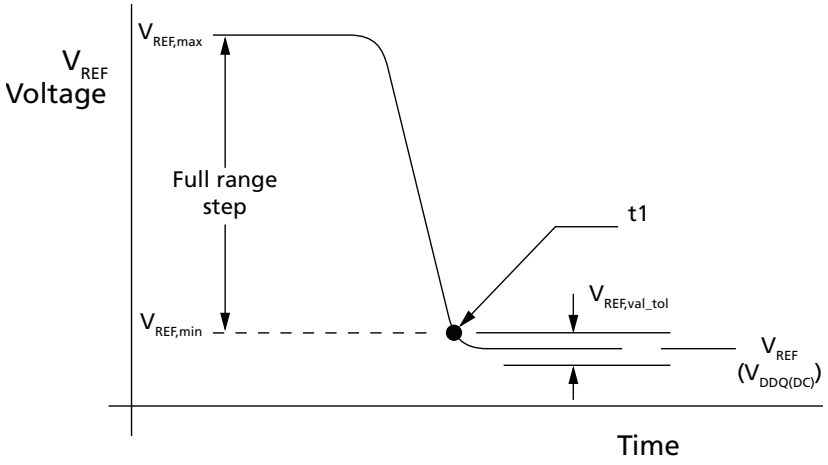


Figure 72: V_{REF} Full Step: From $V_{REF,max}$ to $V_{REF,min}$Case



V_{REFDQ} Target Settings

The V_{REFDQ} initial settings are largely dependant on the ODT termination settings. The table below shows all of the possible initial settings available for V_{REFDQ} training; it is unlikely the lower ODT settings would be used in most cases.

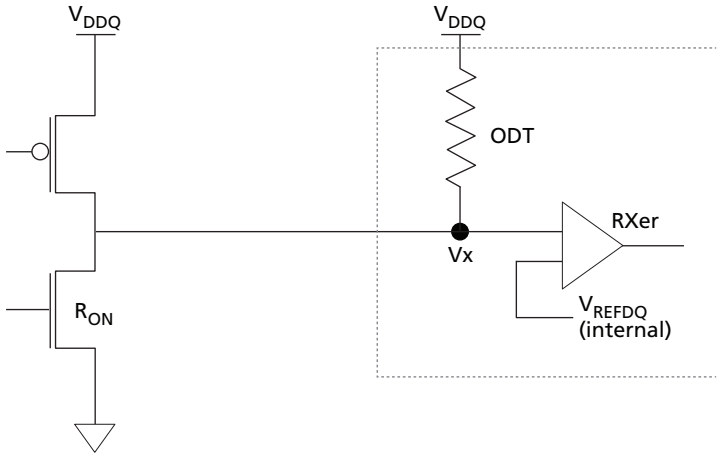
Table 38: V_{REFDQ} Settings (V_{DDQ} = 1.2V)

R_{ON}	ODT	Vx – V_{IN} LOW (mV)	V_{REFDQ} (mv)	V_{REFDQ} (%V_{DDQ})
34 ohm	34 ohm	600	900	75%
	40 ohm	550	875	73%
	48 ohm	500	850	71%
	60 ohm	435	815	68%
	80 ohm	360	780	65%
	120 ohm	265	732	61%
	240 ohm	150	675	56%

Table 38: V_{REFDQ} Settings (V_{DDQ} = 1.2V)

R_{ON}	ODT	Vx – V_{IN} LOW (mV)	V_{REFDQ} (mv)	V_{REFDQ} (%V_{DDQ})
48 ohm	34 ohm	700	950	79%
	40 ohm	655	925	77%
	48 ohm	600	900	75%
	60 ohm	535	865	72%
	80 ohm	450	825	69%
	120 ohm	345	770	64%
	240 ohm	200	700	58%

Figure 73: V_{REFDQ} Equivalent Circuit



Connectivity Test Mode

Connectivity test (CT) mode is similar to boundary scan testing but is designed to significantly speed up the testing of electrical continuity of pin interconnections between the device and the memory controller on the PC boards. Designed to work seamlessly with any boundary scan device, CT mode is supported in all ×4, ×8, and ×16 non-3DS devices (JEDEC states CT mode for ×4 and ×8 is not required on 4Gb and is an optional feature on 8Gb and above). 3DS devices do not support CT mode and the TEN pin should be considered RFU maintained LOW at all times.

Contrary to other conventional shift-register-based test modes, where test patterns are shifted in and out of the memory devices serially during each clock, the CT mode allows test patterns to be entered on the test input pins in parallel and the test results to be extracted from the test output pins of the device in parallel. These two functions are also performed at the same time, significantly increasing the speed of the connectivity check. When placed in CT mode, the device appears as an asynchronous device to the external controlling agent. After the input test pattern is applied, the connectivity test results are available for extraction in parallel at the test output pins after a fixed propagation delay time.

Note: A reset of the device is required after exiting CT mode (see RESET and Initialization Procedure).

Pin Mapping

Only digital pins can be tested using the CT mode. For the purposes of a connectivity check, all the pins used for digital logic in the device are classified as one of the following types:

- **Test enable (TEN):** When asserted HIGH, this pin causes the device to enter CT mode. In CT mode, the normal memory function inside the device is bypassed and the I/O pins appear as a set of test input and output pins to the external controlling agent. Additionally, the device will set the internal V_{REFDQ} to V_{DDQ} × 0.5 during CT mode (this is the only time the DRAM takes direct control over setting the internal V_{REFDQ}). The TEN pin is dedicated to the connectivity check function and will not be used during normal device operation.
- **Chip select (CS_n):** When asserted LOW, this pin enables the test output pins in the device. When de-asserted, these output pins will be High-Z. The CS_n pin in the device serves as the CS_n pin in CT mode.
- **Test input:** A group of pins used during normal device operation designated as test input pins. These pins are used to enter the test pattern in CT mode.
- **Test output:** A group of pins used during normal device operation designated as test output pins. These pins are used for extraction of the connectivity test results in CT mode.
- **RESET_n:** This pin must be fixed high level during CT mode, as in normal function.

Table 39: Connectivity Mode Pin Description and Switching Levels

CT Mode Pins		Pin Name During Normal Memory Operation	Switching Level	Notes
Test enable		TEN	CMOS (20%/80% V_{DD})	1, 2
Chip select		CS_n	V_{REFCA} ±200mV	3
Test input	A	BA[1:0], BG[1:0], A[9:0], A10/AP, A11, A12/BC_n, A13, WE_n/A14, CAS_n/A15, RAS_n/A16, A17, CKE, ACT_n, ODT, CLK_t, CLK_c, PAR	V_{REFCA} ±200mV	3
	B	LDM_n/LDBI_n, UDM_n/UDBI_n; DM_n/DBI_n	V_{REFDQ} ±200mV	4
	C	ALERT_n	CMOS (20%/80% V_{DD})	2, 5
	D	RESET_n	CMOS (20%/80% V_{DD})	2
Test output		DQ[15:0], UDQS_t, UDQS_c, LDQS_t, LDQS_c; DQS_t, DQS_c	V_{TT} ±100mV	6

Notes: 1. TEN: Connectivity test mode is active when TEN is HIGH and inactive when TEN is LOW. TEN must be LOW during normal operation.
2. CMOS is a rail-to-rail signal with DC HIGH at 80% and DC LOW at 20% of V_{DD} (960mV for DC HIGH and 240mV for DC LOW.)
3. V_{REFCA} should be $V_{DD}/2$.
4. V_{REFDQ} should be $V_{DDQ}/2$.
5. ALERT_n switching level is not a final setting.
6. V_{TT} should be set to $V_{DD}/2$.

Minimum Terms Definition for Logic Equations

The test input and output pins are related by the following equations, where INV denotes a logical inversion operation and XOR a logical exclusive OR operation:

MT0 = XOR (A1, A6, PAR)
MT1 = XOR (A8, ALERT_n, A9)
MT2 = XOR (A2, A5, A13) or XOR (A2, A5, A13, A17)
MT3 = XOR (A0, A7, A11)
MT4 = XOR (CK_c, ODT, CAS_n/A15)
MT5 = XOR (CKE, RAS_n/A16, A10/AP)
MT6 = XOR (ACT_n, A4, BA1)
MT7 = ×16: XOR (DMU_n/DBIU_n, DML_n/DBIL_n, CK_t)
= x8: XOR (BG1, DML_n/DBIL_n, CK_t)
= x4: XOR (BG1, CK_t)
MT8 = XOR (WE_n/A14, A12 / BC, BA0)
MT9 = XOR (BG0, A3, RESET_n and TEN)

Logic Equations for a x4 Device

DQ0 = XOR (MT0, MT1)
DQ1 = XOR (MT2, MT3)
DQ2 = XOR (MT4, MT5)
DQ3 = XOR (MT6, MT7)
DQS_t = MT8
DQS_c = MT9

Logic Equations for a x8 Device

DQ0 = MT0	DQ5 = MT5
DQ1 = MT1	DQ6 = MT6
DQ2 = MT2	DQ7 = MT7
DQ3 = MT3	DQS_t = MT8
DQ4 = MT4	DQS_c = MT9

Logic Equations for a x16 Device

DQ0 = MT0	DQ10 = INV DQ2
DQ1 = MT1	DQ11 = INV DQ3
DQ2 = MT2	DQ12 = INV DQ4
DQ3 = MT3	DQ13 = INV DQ5
DQ4 = MT4	DQ14 = INV DQ6
DQ5 = MT5	DQ15 = INV DQ7
DQ6 = MT6	LDQS_t = MT8
DQ7 = MT7	LDQS_c = MT9
DQ8 = INV DQ0	UDQS_t = INV LDQS_t
DQ9 = INV DQ1	UDQS_c = INV LDQS_c

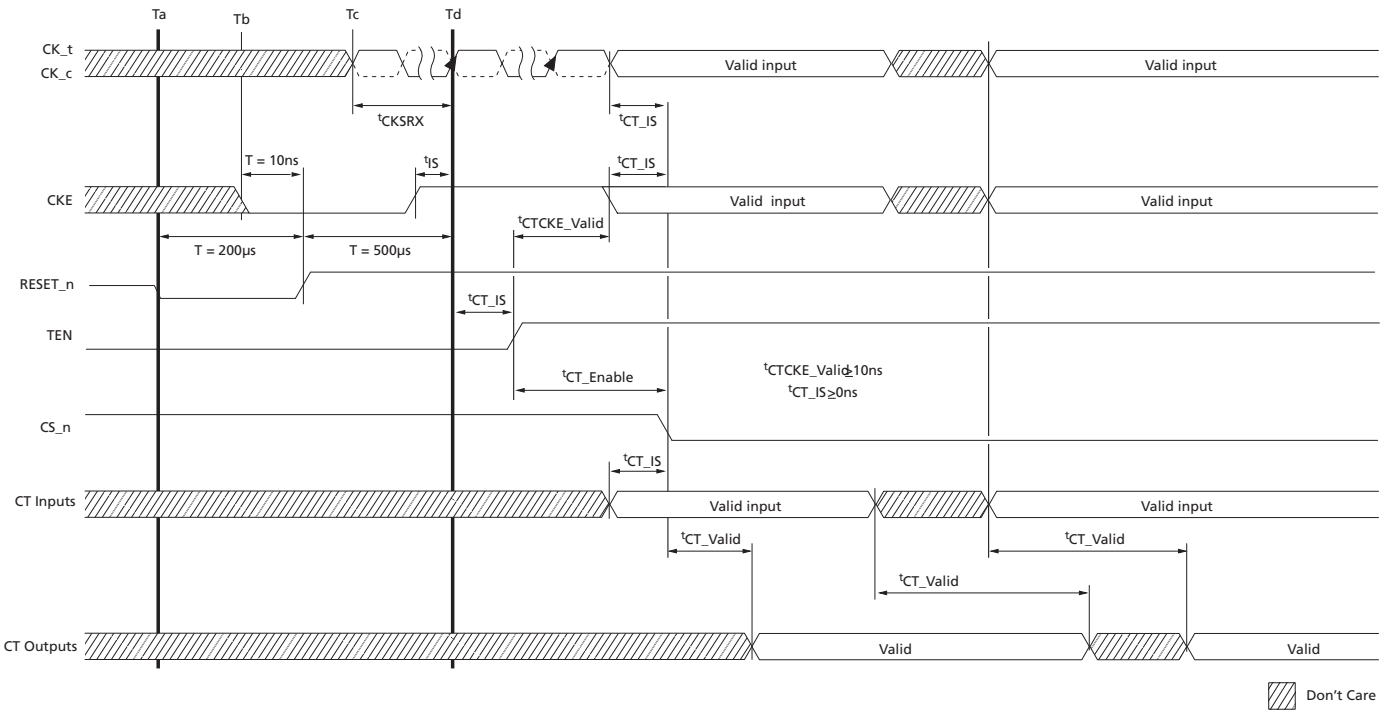
CT Input Timing Requirements

Prior to the assertion of the TEN pin, all voltage supplies, including V_{REFCA}, must be valid and stable and RESET_n registered high prior to entering CT mode. Upon the assertion of the TEN pin HIGH with RESET_n, CKE, and CS_n held HIGH; CLK_t, CLK_c, and CKE signals become test inputs within ^{t}CTECT_Valid. The remaining CT inputs become valid ^{t}CT_Enable after TEN goes HIGH when CS_n allows input to begin sampling, provided inputs were valid for at least ^{t}CT_Valid. While in CT mode, refresh activities in the memory arrays are not allowed; they are initiated either externally (auto refresh) or internally (self refresh).

The TEN pin may be asserted after the DRAM has completed power-on. After the DRAM is initialized and V_{REFDQ} is calibrated, CT mode may no longer be used. The TEN pin may be de-asserted at any time in CT mode. Upon exiting CT mode, the states and the integrity of the original content of the memory array are unknown. A full reset of the memory device is required.

After CT mode has been entered, the output signals will be stable within ^{t}CT_Valid after the test inputs have been applied as long as TEN is maintained HIGH and CS_n is maintained LOW.

Figure 74: Connectivity Test Mode Entry



Excessive Row Activation

Rows can be accessed a limited number of times within a certain time period before adjacent rows require refresh. The maximum activate count (MAC) is the maximum number of activates that a single row can sustain within a time interval of equal to or less than the maximum activate window (^{t}MAW) before the adjacent rows need to be refreshed, regardless of how the activates are distributed over ^{t}MAW.

Micron's DDR4 devices automatically perform a type of TRR mode in the background and provide an MPR Page 3 MPR3[3:0] of 1000, indicating there is no restriction to the number of ACTIVATE commands to a given row in a refresh period provided DRAM timing specifications are not violated. However, specific attempts to by-pass TRR may result in data disturb.

Table 40: MAC Encoding of MPR Page 3 MPR3

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]	MAC	Comments
x	x	x	x	0	0	0	0	Untested	The device has not been tested for MAC.
x	x	x	x	0	0	0	1	^{t}MAC = 700K	
x	x	x	x	0	0	1	0	^{t}MAC = 600K	
x	x	x	x	0	0	1	1	^{t}MAC = 500K	
x	x	x	x	0	1	0	0	^{t}MAC = 400K	
x	x	x	x	0	1	0	1	^{t}MAC = 300K	
x	x	x	x	0	1	1	0	Reserved	
x	x	x	x	0	1	1	1	^{t}MAC = 200K	
x	x	x	x	1	0	0	0	Unlimited	There is no restriction to the number of ACTIVATE commands to a given row in a refresh period provided DRAM timing specifications are not violated.
x	x	x	x	1	0	0	1	Reserved	
x	x	x	x	:	:	:	:	Reserved	
x	x	x	x	1	1	1	1	Reserved	

Notes: 1. MAC encoding in MPR Page 3 MPR3.

Post Package Repair

Post Package Repair

JEDEC defines two modes of Post Package Repair (PPR): soft Post Package Repair (sPPR) and hard Post Package Repair (hPPR). sPPR is non-persistent so the repair row maybe altered; that is, sPPR is NOT a permanent repair and even though it will repair a row, the repair can be reversed, reassigned via another sPPR, or made permanent via hPPR. Hard Post Package Repair is persistent so once the repair row is assigned for a hPPR address, further PPR commands to a previous hPPR section should not be performed, that is, hPPR is a permanent repair; once repaired, it cannot be reversed. The controller provides the failing row address in the hPPR/sPPR sequence to the device to perform the row repair. hPPR Mode and sPPR Mode may not be enabled at the same time.

JEDEC states hPPR is optional for 4Gb and sPPR is optional for 4Gb and 8Gb parts however Micron 4Gb and 8Gb DDR4 DRAMs should have both sPPR and hPPR support. The hPPR support is identified via an MPR read from MPR Page 2, MPR0[7] and sPPR support is identified via an MPR read from MPR Page 2, MPR0[6].

The JEDEC minimum support requirement for DDR4 PPR (hPPR or sPPR) is to provide one row of repair per bank group (BG), x4/x8 have 4 BG and x16 has 2 BG; this is a total of 4 repair rows available on x4/x8 and 2 repair rows available on x16. Micron PPR support exceeds the JEDEC minimum requirements; Micron DDR4 DRAMs have at least one row of repair for each bank which is essentially 4 row repairs per BG for a total of 16 repair rows for x4 and x8 and 8 repair rows for x16; a 4x increase in repair rows.

JEDEC requires the user to have all sPPR row repair addresses reset and cleared prior to enabling hPPR Mode. Micron DDR4 PPR does not have this restriction, the existing sPPR row repair addresses are not required to be cleared prior to entering hPPR mode. Each bank in a BG is PPR independent: sPPR or hPPR issued to a bank will not alter a sPPR row repair existing in a different bank.

sPPR followed by sPPR to same bank

When PPR is issued to a bank for the first time and is a sPPR command, the repair row will be a sPPR. When a subsequent sPPR is issued to the same bank, the previous sPPR repair row will be cleared and used for the subsequent sPPR address as the sPPR operation is non-persistent.

sPPR followed by hPPR to same bank

When a PPR is issued to a bank for the first time and is a sPPR command, the repair row will be a sPPR. When a subsequent hPPR is issued to the same bank, the initial sPPR repair row will be cleared and used for the hPPR address[1]. If a further subsequent PPR (hPPR or sPPR) is issued to the same bank, the further subsequent PPR (hPPR or sPPR) repair row will not clear or overwrite the previous hPPR address as the hPPR operation is persistent.

hPPR followed by hPPR or sPPR to same bank

When a PPR is issued to a bank for the first time and is a hPPR command, the repair row will be a hPPR. When a subsequent PPR (hPPR or sPPR) is issued to the same bank, the subsequent PPR (hPPR or sPPR) repair row will not clear or overwrite the initial hPPR address as the initial hPPR is persistent.

Note: Newer Micron DDR4 designs may not guarantee that an sPPR followed by an hPPR to the same bank will result the same repair row being used. Contact factory for more information.

Hard Post Package Repair

All banks must be precharged and idle. DBI and CRC modes must be disabled. Both sPPR and hPPR must be disabled. sPPR is disabled with MR4[5] = 0. hPPR is disabled with MR4[13] = 0, which is the

normal state, and hPPR is enabled with MR4 [13]= 1, which is the hPPR enabled state. There are two forms of hPPR mode. Both forms of hPPR have the same entry requirement as defined in the sections below. The first command sequence uses a WRA command and supports data retention with a REFRESH operation except for the bank containing the row that is being repaired; JEDEC has relaxed this requirement and allows BA[0] to be a Don't Care regarding the banks which are not required to maintain data a REFRESH operation during hPPR. The second command sequence uses a WR command (a REFRESH operation can't be performed in this command sequence). The second command sequence doesn't support data retention for the target DRAM.

hPPR Row Repair - Entry

As stated above, all banks must be precharged and idle. DBI and CRC modes must be disabled, and all timings must be followed as shown in the timing diagram that follows.

All other commands except those listed in the following sequences are illegal.

1. Issue MR4[13] 1 to enter hPPR mode enable.
   a) All DQ are driven HIGH.
2. Issue four consecutive guard key commands (shown in the table below) to MR0 with each command separated by ^{t}MOD. The PPR guard key settings are the same whether performing sPPR or hPPR mode.
   a) Any interruption of the key sequence by other commands, such as ACT, WR, RD, PRE, REF, ZQ, and NOP, are not allowed.
   b) If the guard key bits are not entered in the required order or interrupted with other MR commands, hPPR will not be enabled, and the programming cycle will result in a NOP.
   c) When the hPPR entry sequence is interrupted and followed by ACT and WR commands, these commands will be conducted as normal DRAM commands.
   d) JEDEC allows A6:0 to be Don't Care on 4Gb and 8Gb devices from a supplier perspective and the user should rely on vendor datasheet.

Table 41: PPR MR0 Guard Key Settings

MR0	BG1:0	BA1:0	A17:12	A11	A10	A9	A8	A7	A6:0
First guard key	0	0	xxxxxx	1	1	0	0	1	1111111
Second guard key	0	0	xxxxxx	0	1	1	1	1	1111111
Third Guard key	0	0	xxxxxx	1	0	1	1	1	1111111
Fourth guard key	0	0	xxxxxx	0	0	1	1	1	1111111

hPPR Row Repair – WRA Initiated (REF Commands Allowed)

1. Issue an ACT command with failing BG and BA with the row address to be repaired.
2. Issue a WRA command with BG and BA of failing row address.
   a) The address must be at valid levels, but the address is Don't Care.
3. All DQ of the target DRAM should be driven LOW for 4*n*CK (bit 0 through bit 7) after WL (WL = CWL + AL + PL) in order for hPPR to initiate repair.
   a) Repair **will be** initiated to the target DRAM only if all DQ during bit 0 through bit 7 are LOW. The bank under repair does not get the REFRESH command applied to it.
   b) Repair **will not be** initiated to the target DRAM if any DQ during bit 0 through bit 7 is HIGH.
      i) JEDEC states: All DQs of target DRAM should be LOW for 4^{t}CK. If HIGH is driven to all DQs of a DRAM consecutively for equal to or longer than 2^{t}CK, then DRAM does not conduct hPPR

and retains data if REF command is properly issued; if all DQs are neither LOW for 4^{t}CK nor HIGH for equal to or longer than 2^{t}CK, then hPPR mode execution is unknown.

c) DQS should function normally.

4. REF command may be issued anytime after the WRA command followed by WL + 4*n*CK + ^{t}WR + ^{t}RP.
   a) Multiple REF commands are issued at a rate of ^{t}REFI or ^{t}REFI/2, however back-to-back REF commands must be separated by at least ^{t}REFI/4 when the DRAM is in hPPR mode.
   b) All banks except the bank under repair will perform refresh.
5. Issue PRE after ^{t}PGM time so that the device can repair the target row during ^{t}PGM time.
   a) Wait ^{t}PGM_Exit after PRE to allow the device to recognize the repaired target row address.
6. Issue MR4[13] 0 command to hPPR mode disable.
   a) Wait ^{t}PGMPST for hPPR mode exit to complete.
   b) After ^{t}PGMPST has expired, any valid command may be issued.

The entire sequence from hPPR mode enable through hPPR mode disable may be repeated if more than one repair is to be done.

After completing hPPR mode, MR0 must be re-programmed to a prehPPR mode state if the device is to be accessed.

After hPPR mode has been exited, the DRAM controller can confirm if the target row was repaired correctly by writing data into the target row and reading it back.

Figure 75: hPPR WRA – Entry

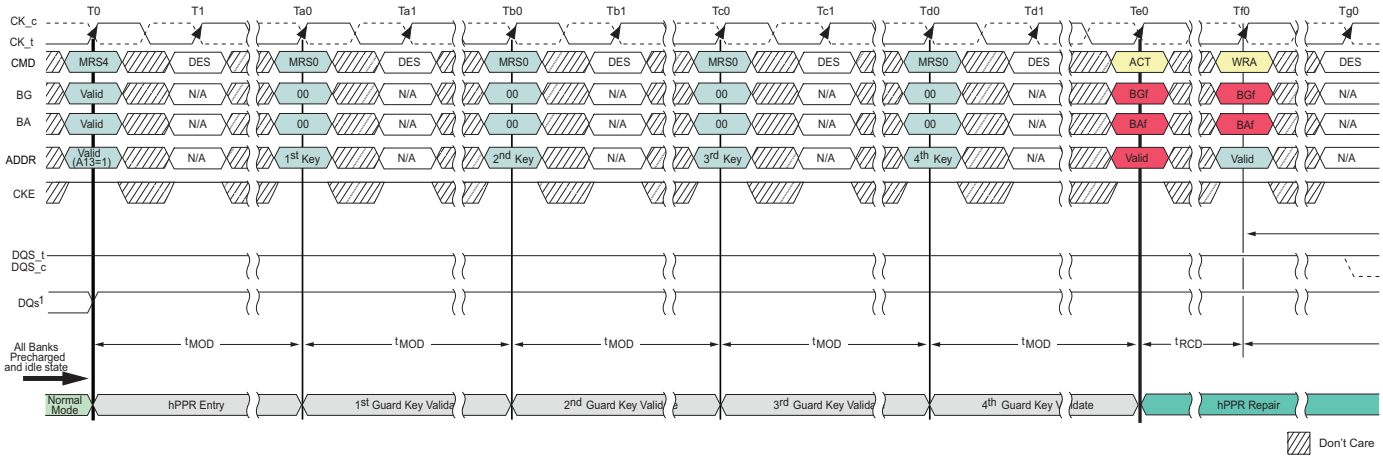
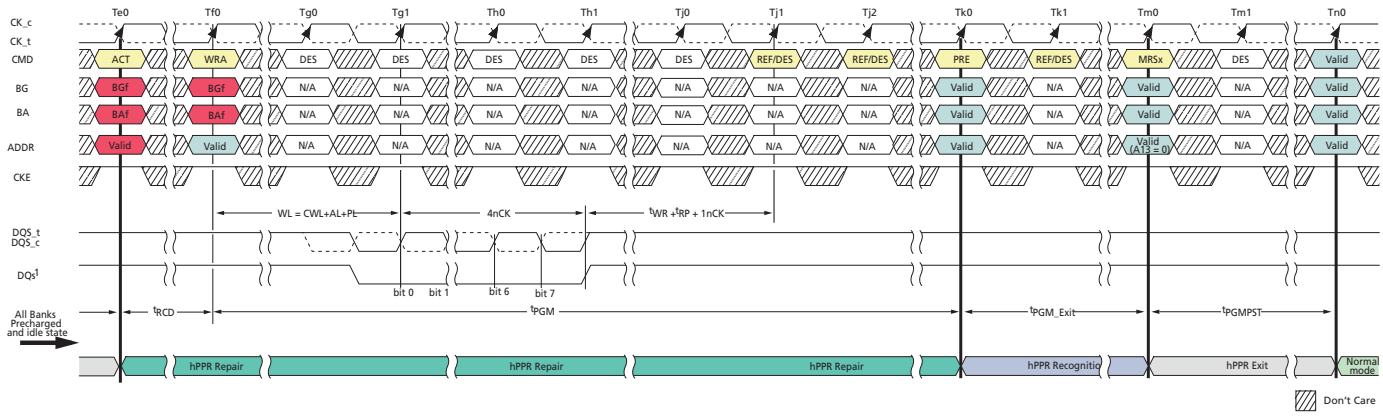


Figure 76: hPPR WRA – Repair and Exit



hPPR Row Repair – WR Initiated (REF Commands NOT Allowed)

1. Issue an ACT command with failing BG and BA with the row address to be repaired.
2. Issue a WR command with BG and BA of failing row address.
   a) The address must be at valid levels, but the address is Don't Care.
3. All DQ of the target DRAM should be driven LOW for 4*n*CK (bit 0 through bit 7) after WL (WL = CWL + AL + PL) in order for hPPR to initiate repair.
   a) Repair **will be** initiated to the target DRAM only if all DQ during bit 0 through bit 7 are LOW.
   b) Repair **will not be** initiated to the target DRAM if any DQ during bit 0 through bit 7 is HIGH.
      i) JEDEC states: All DQs of target DRAM should be LOW for 4^{t}CK. If HIGH is driven to all DQs of a DRAM consecutively for equal to or longer than 2^{t}CK, then DRAM does not conduct hPPR and retains data if REF command is properly issued; if all DQs are neither LOW for 4^{t}CK nor HIGH for equal to or longer than 2^{t}CK, then hPPR mode execution is unknown.
   c) DQS should function normally.
4. REF commands may NOT be issued at anytime while in PPT mode.
5. Issue PRE after ^{t}PGM time so that the device can repair the target row during ^{t}PGM time.
   a) Wait ^{t}PGM_Exit after PRE to allow the device to recognize the repaired target row address.
6. Issue MR4[13] 0 command to hPPR mode disable.
   a) Wait ^{t}PGMPST for hPPR mode exit to complete.
   b) After ^{t}PGMPST has expired, any valid command may be issued.

The entire sequence from hPPR mode enable through hPPR mode disable may be repeated if more than one repair is to be done.

After completing hPPR mode, MR0 must be re-programmed to a prehPPR mode state if the device is to be accessed.

After hPPR mode has been exited, the DRAM controller can confirm if the target row was repaired correctly by writing data into the target row and reading it back.

Figure 77: hPPR WR – Entry

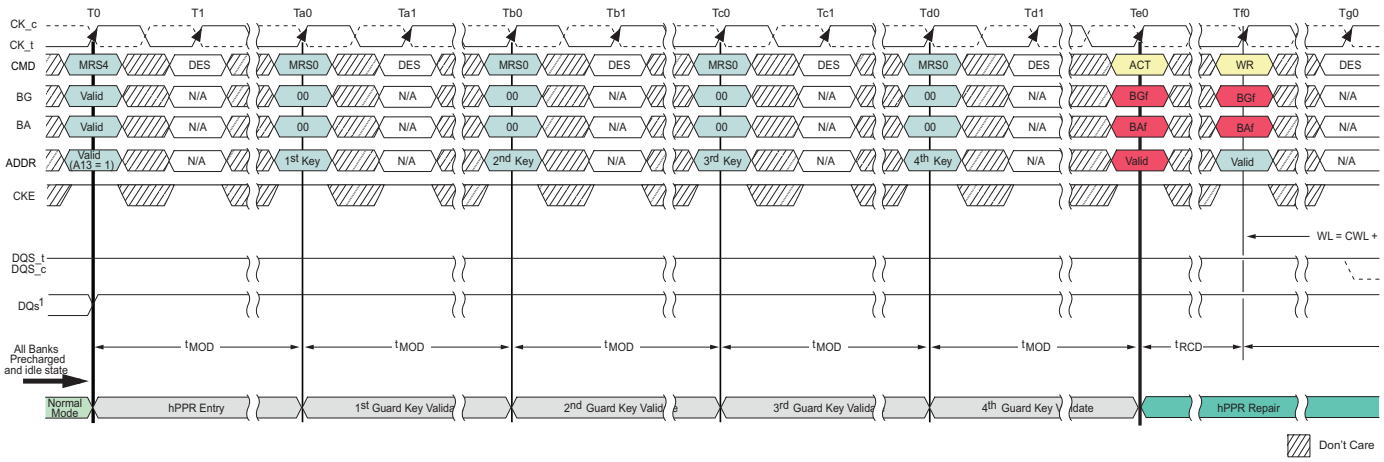


Figure 78: hPPR WR – Repair and Exit

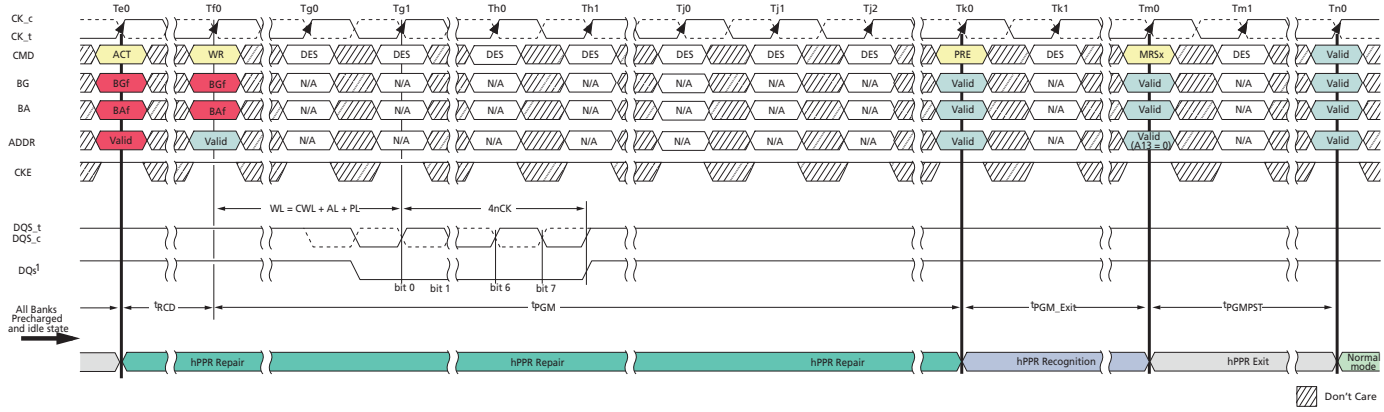


Table 42: DDR4 hPPR Timing Parameters DDR4-1600 through DDR4-3200

Parameter	Symbol		Min	Max	Unit
hPPR programming time	tPGM	×4, ×8	1000	–	ms
		×16	2000	–	ms
hPPR precharge exit time	tPGM_Exit		15	–	ns
hPPR exit time	tPGMPST		50	–	µs

sPPR Row Repair

Soft post package repair (sPPR) is a way to quickly, but temporarily, repair a row element in a bank on a DRAM device, where hPPR takes longer but permanently repairs a row element. sPPR mode is entered in a similar fashion as hPPR, sPPR uses MR4[5] while hPPR uses MR4[13]. sPPR is disabled with MR4[5] = 0, which is the normal state, and sPPR is enabled with MR4[5] = 1, which is the sPPR enabled state.

sPPR requires the same guard key sequence as hPPR to qualify the MR4 PPR entry. After sPPR entry, an ACT command will capture the target bank and target row, herein seed row, where the row repair will be made. After tRCD time, a WR command is used to select the individual DRAM, through the DQ bits,

to transfer the repair address into an internal register in the DRAM. After a write recovery time and PRE command, the sPPR mode can be exited and normal operation can resume.

The DRAM will retain the soft repair information as long as V_{DD} remains within the operating region unless rewritten by a subsequent sPPR entry to the same bank. If DRAM power is removed or the DRAM is reset, the soft repair will revert to the unrepaired state. hPPR and sPPR should not be enabled at the same time; Micron sPPR does not have to be disabled and cleared prior to entering hPPR mode, but sPPR must be disabled and cleared prior to entering MBIST-PPR mode.

With sPPR, Micron DDR4 can repair one row per bank. When a subsequent sPPR request is made to the same bank, the subsequently issued sPPR address will replace the previous sPPR address. When the hPPR resource for a bank is used up, the bank should be assumed to not have available resources for sPPR. If a repair sequence is issued to a bank with no repair resource available, the DRAM will ignore the programming sequence.

The bank receiving sPPR change is expected to retain memory array data in all rows except for the seed row and its associated row addresses. If the data in the memory array in the bank under sPPR repair is not required to be retained, then the handling of the seed row's associated row addresses is not of interest and can be ignored. If the data in the memory array is required to be retained in the bank under sPPR mode, then prior to executing the sPPR mode, the seed row and its associated row addresses should be backed up and subsequently restored after sPPR has been completed. sPPR associated seed row addresses are specified in the Table below; BA0 is not required by Micron DRAMs however it is JEDEC reserved.

Table 43: sPPR Associated Rows

sPPR Associated Row Address							
BA0*	A17	A16	A15	A14	A13	A1	A0

All banks must be precharged and idle. DBI and CRC modes must be disabled, and all sPPR timings must be followed as shown in the timing diagram that follows.

All other commands except those listed in the following sequences are illegal.

1. Issue MR4[5] 1 to enter sPPR mode enable.
   a) All DQ are driven HIGH.
2. Issue four consecutive guard key commands (shown in the table below) to MR0 with each command separated by ^{t}MOD. Please note that JEDEC recently added the four guard key entry used for hPPR to sPPR entry; early DRAMs may not require four guard key entry code. A prudent controller design should accommodate either option in case an earlier DRAM is used.
   a) Any interruption of the key sequence by other commands, such as ACT, WR, RD, PRE, REF, ZQ, and NOP, are not allowed.
   b) If the guard key bits are not entered in the required order or interrupted with other MR commands, sPPR will not be enabled, and the programming cycle will result in a NOP.
   c) When the sPPR entry sequence is interrupted and followed by ACT and WR commands, these commands will be conducted as normal DRAM commands.
   d) JEDEC allows A6:0 to be "Don't Care" on 4Gb and 8Gb devices from a supplier perspective and the user should rely on vendor datasheet.

Table 44: PPR MR0 Guard Key Settings

MR0	BG1:0	BA1:0	A17:12	A11	A10	A9	A8	A7	A6:0
First guard key	0	0	xxxxxx	1	1	0	0	1	1111111
Second guard key	0	0	xxxxxx	0	1	1	1	1	1111111

Table 44: PPR MR0 Guard Key Settings (Continued)

MR0	BG1:0	BA1:0	A17:12	A11	A10	A9	A8	A7	A6:0
Third guard key	0	0	xxxxxx	1	0	1	1	1	1111111
Fourth guard key	0	0	xxxxxx	0	0	1	1	1	1111111

3. After ^{t}MOD, issue an ACT command with failing BG and BA with the row address to be repaired.
4. After ^{t}RCD, issue a WR command with BG and BA of failing row address.
   a) The address must be at valid levels, but the address is a "Don't Care."
5. All DQ of the target DRAM should be driven LOW for 4*n*CK (bit 0 through bit 7) after WL (WL = CWL + AL + PL) in order for sPPR to initiate repair.
   a) Repair **will be** initiated to the target DRAM only if all DQ during bit 0 through bit 7 are LOW.
   b) Repair **will not be** initiated to the target DRAM if any DQ during bit 0 through bit 7 is HIGH.
      i) JEDEC states: All DQs of target DRAM should be LOW for 4^{t}CK. If HIGH is driven to all DQs of a DRAM consecutively for equal to or longer than the first 2^{t}CK, then DRAM does not conduct hPPR and retains data if REF command is properly issued; if all DQs are neither LOW for 4^{t}CK nor HIGH for equal to or longer than the first 2^{t}CK, then hPPR mode execution is unknown.
   c) DQS should function normally.
6. REF command may NOT be issued at anytime while in sPPR mode.
7. Issue PRE after ^{t}WR time so that the device can repair the target row during ^{t}WR time.
   a) Wait ^{t}PGM_Exit_s after PRE to allow the device to recognize the repaired target row address.
8. Issue MR4[5] 0 command to sPPR mode disable.
   a) Wait ^{t}PGMPST_s for sPPR mode exit to complete.
   b) After ^{t}PGMPST_s has expired, any valid command may be issued.

The entire sequence from sPPR mode enable through sPPR mode disable may be repeated if more than one repair is to be done.

After sPPR mode has been exited, the DRAM controller can confirm if the target row was repaired correctly by writing data into the target row and reading it back.

Figure 79: sPPR – Entry

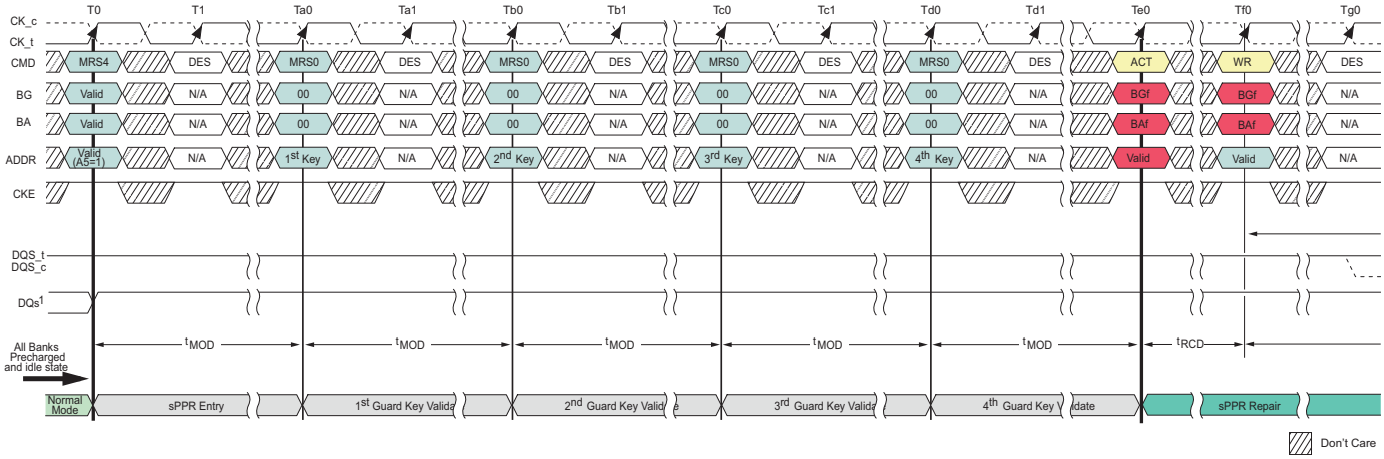


Figure 80: sPPR – Repair, and Exit

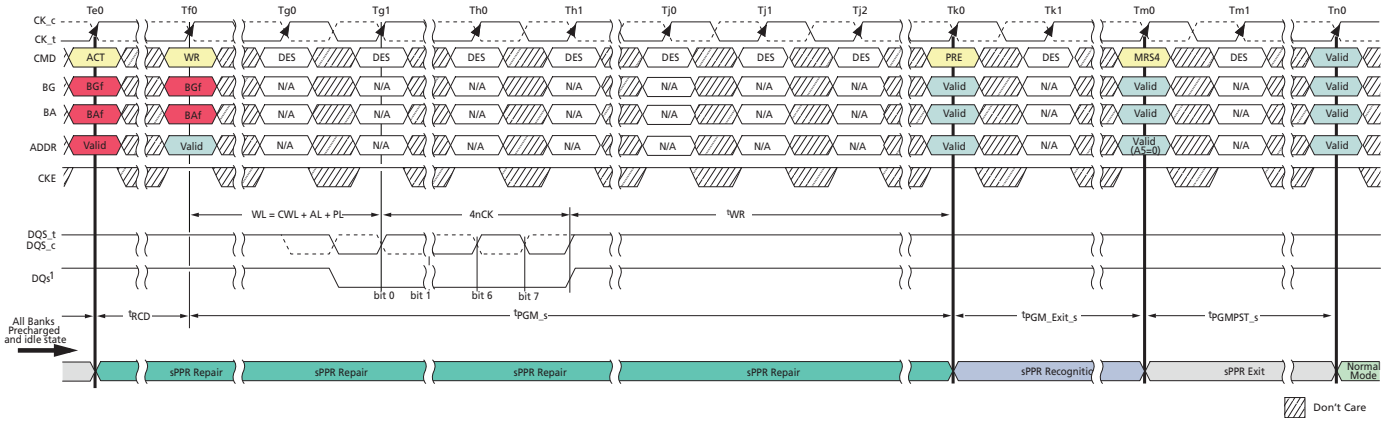


Table 45: DDR4 sPPR Timing Parameters DDR4-1600 Through DDR4-3200

Parameter	Symbol	Min	Max	Unit
sPPR programming time	^{t}PGM_s	^{t}RCD(MIN)+ WL + 4nCK + ^{t}WR(MIN)	–	ns
sPPR precharge exit time	^{t}PGM_Exit_s	20	–	ns
sPPR exit time	^{t}PGMPST_s	^{t}MOD	–	ns

MBIST-PPR

DDR4 devices can support optional memory built-in self-test post-package repair (MBIST-PPR) to help with hard failures such as single-bit or multi-bit failures in a single device so that weak cells can be scanned and repaired during the initialization phase. The DRAM will use vendor-specific patterns to investigate the status of all cell arrays and automatically perform PPR for weak bits during this operation. This operation introduces proactive, automated PPR by the DRAM, and it is recommended to be done for a very first boot-up at least. After that, it is at the controller's discretion whether to activate MBIST. MBIST mode can only be entered from the all banks idle state. The DLL is required to be enabled and locked prior to MBIST-PPR execution.

MBIST-PPR resources are separated from normal hPPR/sPPR resources. MBIST-PPR resources are typically used for initial scan and repair, and hPPR/sPPR resources must still satisfy the number of repair elements, one per BG, specified in the DDR4 Bank Group Timing Examples 1. Once the MBIST-PPR is completed, the DRAM will update the status flag in MPR3[7] of MPR page 3. Detailed status is described in the MPR Page and MPRx Definitions .

The test time of MBIST-PPR will not exceed 10 seconds for all mono-die DRAM densities. For DDP devices, test time will be 20 seconds.

The controller is required to inject an MRS command to enter this operation. The controller sets MR4:A0 to 1, followed by MR0 commands for the guard key. Then the DRAM enters MBIST-PPR operation. The ALERT_n signal notifies the host of the status of this operation. When the controller sets MR4:A0 to 1, followed by the MR0 guard key sequence, the DRAM drives ALERT_n to 0. Once the

MBIST-PPR is completed, the DRAM drives ALERT_n to 1 to notify the controller that this operation is completed. DRAM data will not be guaranteed after the MBIST-PPR operation.

Table 46: MBIST-PPR Timing Parameter

Parameter		Value		Unit
		Min	Max	
^{t}SELFHEAL	Monolithic	–	10	s
	DDP	–	20	

MBIST-PPR Procedure

The following sequences are required for MBIST-PPR and are shown in the figure below.

1. The DRAM needs to finalize initialization, MR training, and ZQ calibration prior to entering MBIST-PPR.
2. Four consecutive guard key commands must be issued to MR0, with each command separated by ^{t}MOD. The PPR guard key settings are the same whether performing sPPR, hPPR, or MBIST-PPR mode.
3. Anytime after Tk in the Read Termination Disable Window 15, the host must set MR4:A0 to 1, followed by subsequent MR0 guard key sequences (which is identical to typical hPPR/sPPR guard key sequences and specified in Table 73) to start MBIST-PPR operation, and the DRAM drives the ALERT_n signal to 0.
4. During MBIST-PPR mode, only DESELECT commands are allowed.
5. The ODT pin must be driven LOW during MBIST-PPR to satisfy DODTLoff from time Tb0 until Tc2. The DRAM may or may not provide RTT_PARK termination during MBIST-PPR regardless of whether RTT_PARK is enabled in MR5.

Figure 81: MBIST-PPR Sequence

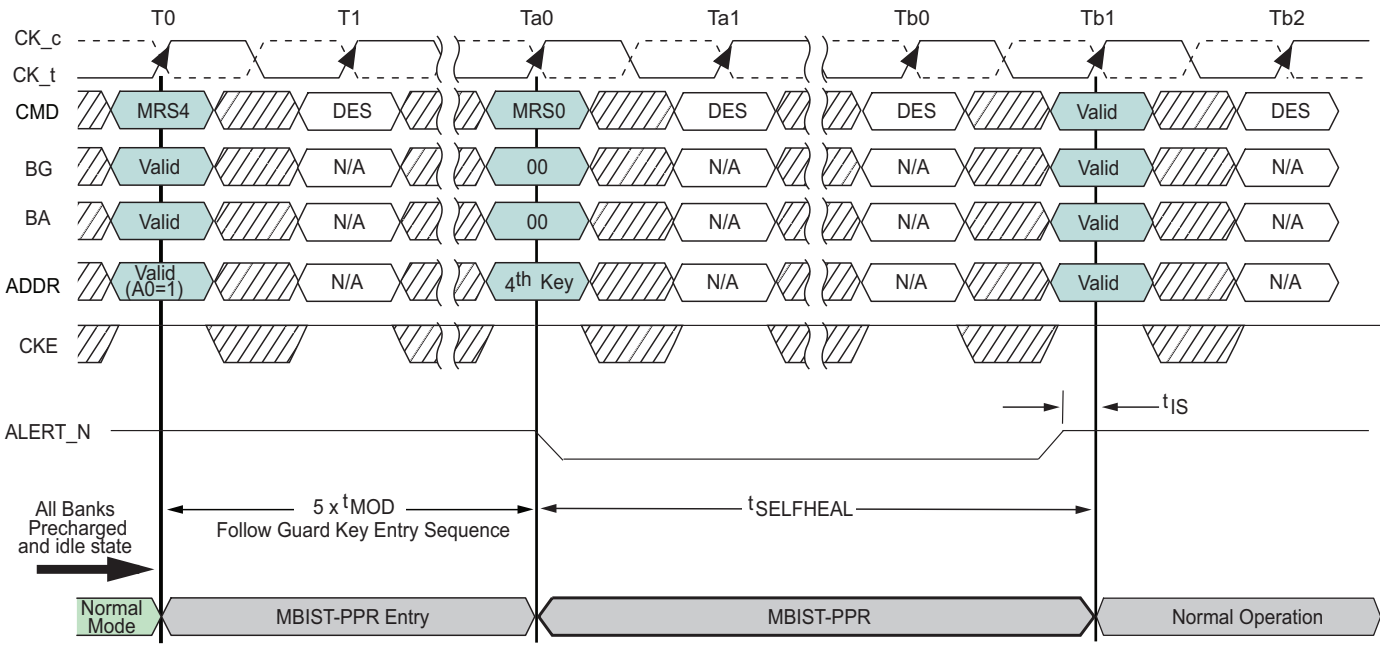


Table 47: MPR Page3 Configuration for MBIST-PPR

Address	MPR Location	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]	Note
BA[1:0]	00 = MPR0	DC	DC	DC	DC	DC	DC	DC	DC	Read-only
	01 = MPR1	DC	DC	DC	DC	DC	DC	DC	DC	
	10 = MPR2	DC	DC	DC	DC	DC	DC	DC	DC	
	11 = MPR3	MBIST-PPR Support	DC	MBIST-PPR Transparency		MAC	MAC	MAC	MAC	

MPR Location	Address Bit	Function	Data	Notes
11 = MPR3	7	MBIST-PPR Support	0: Don't Support 1: Support	1
11 = MPR3	5:4	MBIST-PPR Transparency	00_B: MBIST-PPR hasn't run since init OR no fails found during most recent MBIST-PPR	1, 2
			01_B: Repaired all found fails during most recent run	1
			10_B: Unrepairable fails found during most recent run	1
			11_B: MBIST-PPR should be run again	1, 3

Notes: 1. MPR bits are cleared either by a power-up sequence or re-initialization by RESET_n signal
2. The host should track whether MBIST-PPR has run since INIT. If MBIST-PPR is performed and it finds no fails, this transparency state will remain set to 00_B
3. This state does not imply that MBIST-PPR is required to run again. This implies that additional repairable fails were found during the most recent MBIST-PPR beyond what could be repaired in the tSELFHEAL window.

hPPR/sPPR/MBIST-PPR Support Identifier

Table 48: DDR4 Repair Mode Support Identifier

MPR Page 2	A7	A6	A5	A4	A3	A2	A1	A0
	UI0	UI1	UI2	UI3	UI4	UI5	UI6	UI7
MPR0	hPPR[1]	sPPR[2]	R_{TT_WR}	Temp sensor		CRC	R_{TT_WR}	

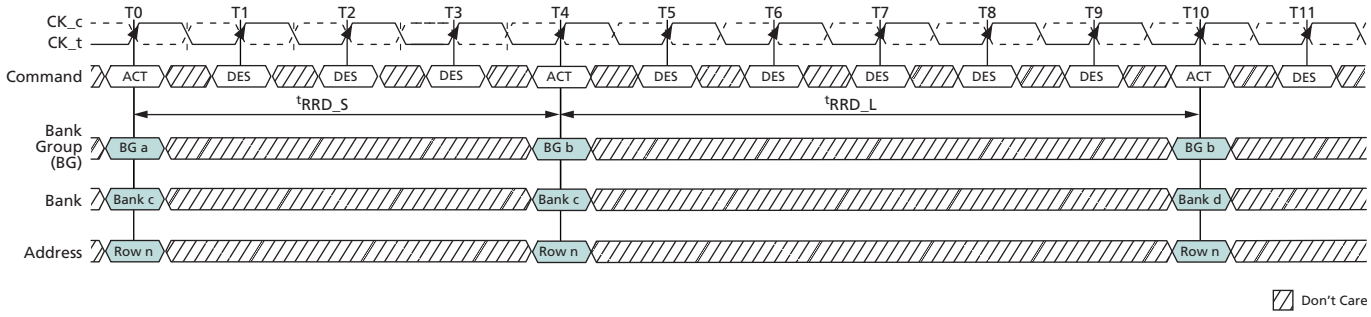
MPR Page 3	A7	A6	A5	A4	A3	A2	A1	A0
	UI0	UI1	UI2	UI3	UI4	UI5	UI6	UI7
MPR3	MBIST-PPR Support[3]	Don't Care	MBIST-PPR Transparency		MAC	MAC	MAC	MAC

Notes: 1. 0 = hPPR mode is not available, 1 = hPPR mode is available.
2. 0 = sPPR mode is not available, 1 = sPPR mode is available.
3. 0 = MBIST-PPR mode is not available, 1 = MBIST-PPR mode is available.
4. Gray shaded areas are for reference only.

ACTIVATE Command

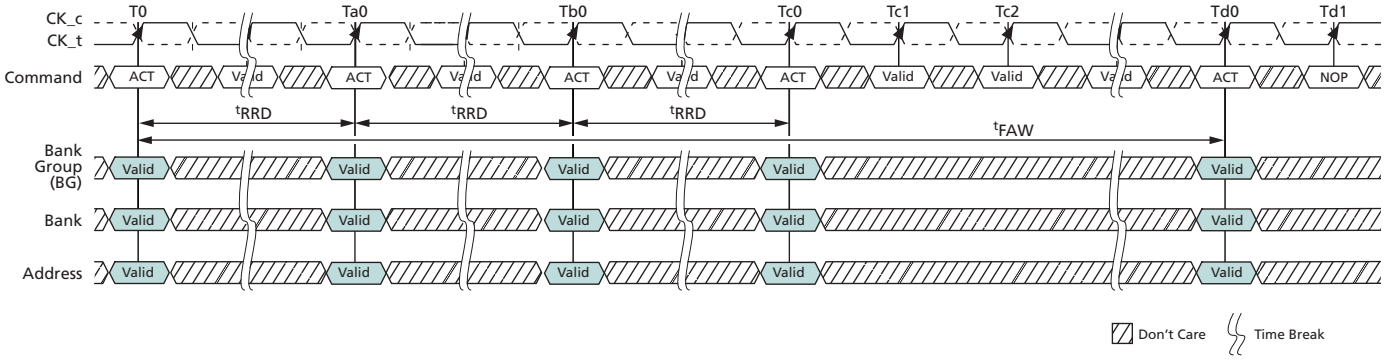
The ACTIVATE command is used to open (activate) a row in a particular bank for subsequent access. The values on the BG[1:0] inputs select the bank group, the BA[1:0] inputs select the bank within the bank group, and the address provided on inputs A[17:0] selects the row within the bank. This row remains active (open) for accesses until a PRECHARGE command is issued to that bank. A PRECHARGE command must be issued before opening a different row in the same bank.
Bank-to-bank command timing for ACTIVATE commands uses two different timing parameters, depending on whether the banks are in the same or different bank group. ^{t}RRD_S (short) is used for timing between banks located in different bank groups. ^{t}RRD_L (long) is used for timing between banks located in the same bank group. Another timing restriction for consecutive ACTIVATE commands [issued at ^{t}RRD (MIN)] is ^{t}FAW (four activate window). Because there is a maximum of four banks in a bank group, the ^{t}FAW parameter applies across different bank groups (five ACTIVATE commands issued at ^{t}RRD_L (MIN) to the same bank group would be limited by ^{t}RC).

Figure 82: ^{t}RRD Timing



Notes: 1. ^{t}RRD_S; ACTIVATE-to-ACTIVATE command period (short); applies to consecutive ACTIVATE commands to different bank groups (that is, T0 and T4).
2. ^{t}RRD_L; ACTIVATE-to-ACTIVATE command period (long); applies to consecutive ACTIVATE commands to the different banks in the same bank group (that is, T4 and T10).

Figure 83: tFAW Timing



Note: 1. tFAW; four activate windows.

PRECHARGE Command

The PRECHARGE command is used to deactivate the open row in a particular bank or the open row in all banks. The bank(s) will be available for a subsequent row activation for a specified time (tRP) after the PRECHARGE command is issued. An exception to this is the case of concurrent auto precharge, where a READ or WRITE command to a different bank is allowed as long as it does not interrupt the data transfer in the current bank and does not violate any other timing parameters.

After a bank is precharged, it is in the idle state and must be activated prior to any READ or WRITE commands being issued to that bank. A PRECHARGE command is allowed if there is no open row in that bank (idle state) or if the previously open row is already in the process of precharging. However, the precharge period will be determined by the last PRECHARGE command issued to the bank.

The auto precharge feature is engaged when a READ or WRITE command is issued with A10 HIGH. The auto precharge feature uses the RAS lockout circuit to internally delay the PRECHARGE operation until the ARRAY RESTORE operation has completed. The RAS lockout circuit feature allows the PRECHARGE operation to be partially or completely hidden during burst READ cycles when the auto precharge feature is engaged. The PRECHARGE operation will not begin until after the last data of the burst write sequence is properly stored in the memory array.

REFRESH Command

The REFRESH command (REF) is used during normal operation of the device. This command is nonpersistent, so it must be issued each time a refresh is required. The device requires REFRESH cycles at an average periodic interval of tREFI. When CS_n, RAS_n/A16, and CAS_n/A15 are held LOW and WE_n/A14 HIGH at the rising edge of the clock, the device enters a REFRESH cycle. All banks of the SDRAM must be precharged and idle for a minimum of the precharge time, tRP (MIN), before the REFRESH command can be applied. The refresh addressing is generated by the internal DRAM refresh controller. This makes the address bits "Don't Care" during a REFRESH command. An internal address counter supplies the addresses during the REFRESH cycle. No control of the external address bus is required once this cycle has started. When the REFRESH cycle has completed, all banks of the SDRAM will be in the precharged (idle) state. A delay between the REFRESH command and the next valid command, except DES, must be greater than or equal to the minimum REFRESH cycle time tRFC (MIN), as shown in .

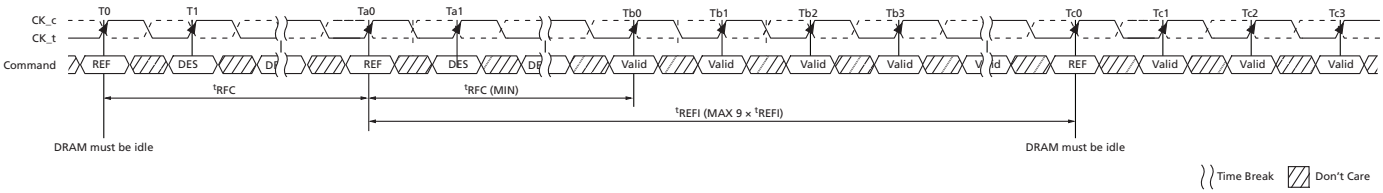
NOTE: The tRFC timing parameter depends on memory density.

In general, a REFRESH command needs to be issued to the device regularly every ^{t}REFI interval. To allow for improved efficiency in scheduling and switching between tasks, some flexibility in the absolute refresh interval is provided for postponing and pulling-in the REFRESH command. A limited number REFRESH commands can be postponed depending on refresh mode: a maximum of 8 REFRESH commands can be postponed when the device is in 1X refresh mode; a maximum of 16 REFRESH commands can be postponed when the device is in 2X refresh mode; and a maximum of 32 REFRESH commands can be postponed when the device is in 4X refresh mode.

When 8 consecutive REFRESH commands are postponed, the resulting maximum interval between the surrounding REFRESH commands is limited to 9 × ^{t}REFI (see). For both the 2X and 4X refresh modes, the maximum interval between surrounding REFRESH commands allowed is limited to 17 × ^{t}REFI2 and 33 × ^{t}REFI4, respectively.

A limited number REFRESH commands can be pulled-in as well. A maximum of 8 additional REFRESH commands can be issued in advance or "pulled-in" in 1X refresh mode, a maximum of 16 additional REFRESH commands can be issued when in advance in 2X refresh mode, and a maximum of 32 additional REFRESH commands can be issued in advance when in 4X refresh mode. Each of these REFRESH commands reduces the number of regular REFRESH commands required later by one. The resulting maximum interval between two surrounding REFRESH commands is limited to 9 × ^{t}REFI ( ), 17 × ^{t}RFEI2, or 33 × ^{t}REFI4. At any given time, a maximum of 16 REF commands can be issued within 2 × ^{t}REFI, 32 REF2 commands can be issued within 4 × ^{t}REFI2, and 64 REF4 commands can be issued within 8 × ^{t}REFI4 (larger densities are limited by tRFC1, tRFC2, and tRFC4, respectively, which must still be met).

Figure 84: REFRESH Command Timing



Notes: 1. Only DES commands are allowed after a REFRESH command is registered until ^{t}RFC (MIN) expires.
2. Time interval between two REFRESH commands may be extended to a maximum of 9 × ^{t}REFI.

Figure 85: Postponing REFRESH Commands (Example)

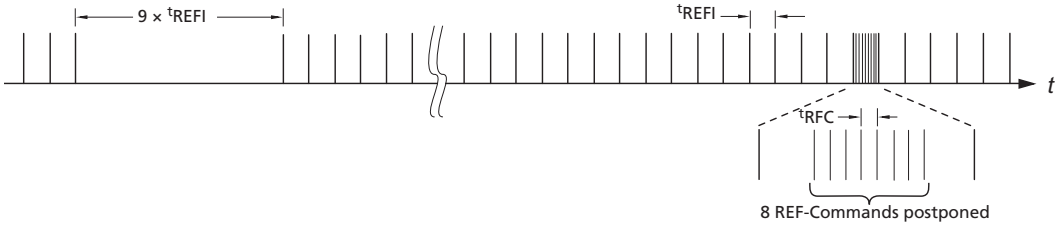
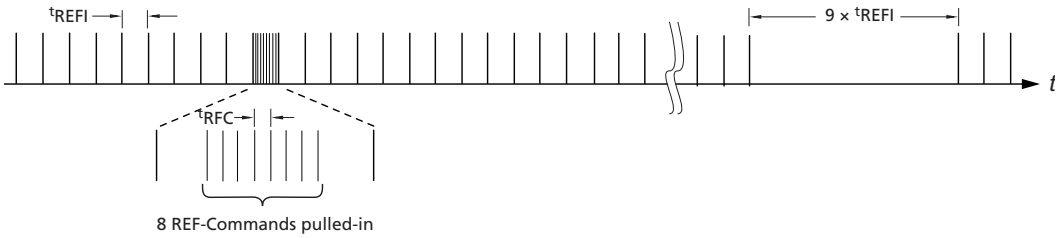


Figure 86: Pulling In REFRESH Commands (Example)



Temperature-Controlled Refresh Mode

During normal operation, temperature-controlled refresh (TCR) mode disabled, the device must have a REFRESH command issued once every ^{t}REFI, except for what is allowed by posting (see REFRESH Command section). This means a REFRESH command must be issued once every 7.8µs if T_C is less than or equal to 85°C, once every 3.9µs if T_C is greater than 85°C, once every 1.95µs if T_C is greater than 95°C, regardless of which Temperature Mode is selected (MR4[2]). TCR mode is disabled by setting MR4[3] = 0 while TCR mode is enabled by setting MR4[3] = 1. When TCR mode is enabled (MR4[3] = 1), the Temperature Mode must be selected where MR4[2] = 0 enables the Normal Temperature Mode while MR4[2] = 1 enables the Extended Temperature Mode.

When TCR mode is enabled, the device will register the externally supplied REFRESH command and adjust the internal refresh period to be longer than ^{t}REFI of the normal temperature range, when allowed, by skipping REFRESH commands with the proper gear ratio. TCR mode has two Temperature Modes to select between the normal temperature range and the extended temperature range; the correct Temperature Mode must be selected so the internal control operates correctly. The DRAM must have the correct refresh rate applied externally; the internal refresh rate is determined by the DRAM based upon the temperature.

Normal Temperature Mode

REFRESH commands should be issued to the device with the refresh period equal to ^{t}REFI of normal temperature range (-40°C to 85°C). The system must guarantee that the T_C does not exceed 85°C when ^{t}REFI of the normal temperature range is used. The device may adjust the internal refresh period to be longer than ^{t}REFI of the normal temperature range by skipping external REFRESH commands with the proper gear ratio when T_C is below 85°C. The internal refresh period is automatically adjusted inside the DRAM, and the DRAM controller does not need to provide any additional control.

Extended Temperature Mode

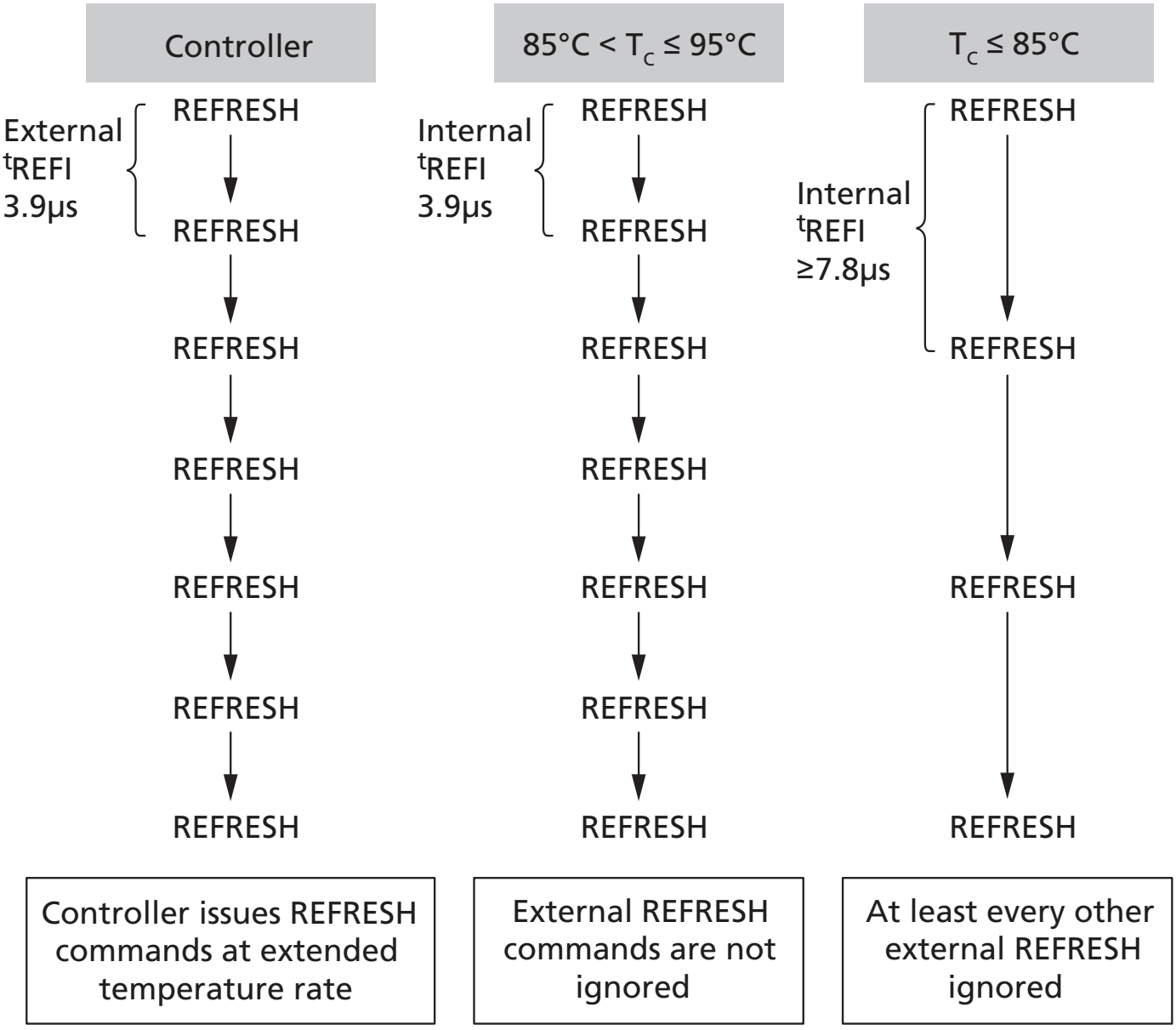
REFRESH commands should be issued to the device with the refresh period equal to ^{t}REFI of extended temperature range (85°C to 95°C, or 95°C to 105°C) . The system must guarantee that the T_C does not exceed 95°C, or 105°C. Even though the external refresh supports the extended temperature range, the device may adjust its internal refresh period to be equal to or longer than ^{t}REFI of the normal temperature range (-40°C to 85°C) by skipping external REFRESH commands with the proper gear ratio when T_C is equal to or below 85°C. The internal refresh period is automatically adjusted inside the DRAM, and the DRAM controller does not need to provide any additional control.

Table 49: Normal ^{t}REFI Refresh (TCR Enabled)

	Normal Temperature Mode		Extended Temperature Mode	
Temperature	External Refresh Period	Internal Refresh Period	External Refresh Period	Internal Refresh Period
$T_C \le 85°C$	7.8µs	≥7.8µs	3.9µs[1]	≥7.8µs
$85°C < T_C \le 95°C$	3.9µs			
$95°C < T_C \le 105°C$	1.95µs			

Notes: 1. If the external refresh period is slower than 3.9µs, the device will refresh internally at too slow of a refresh rate and will violate refresh specifications.

Figure 87: TCR Mode Example[1]



Note: 1. TCR enabled with Extended Temperature Mode selected.

Fine Granularity Refresh Mode

Mode Register and Command Truth Table

The REFRESH cycle time (^{t}RFC) and the average refresh interval (^{t}REFI) can be programmed by the MRS command. The appropriate setting in the mode register will set a single set of REFRESH cycle times and average refresh interval for the device (fixed mode), or allow the dynamic selection of one of two sets of REFRESH cycle times and average refresh interval for the device (on-the-fly mode [OTF]). OTF mode must be enabled by MRS before any OTF REFRESH command can be issued.

Table 50: MRS Definition

MR3[8]	MR3[7]	MR3[6]	Refresh Rate Mode
0	0	0	Normal mode (fixed 1x)
0	0	1	Fixed 2x
0	1	0	Fixed 4x
0	1	1	Reserved
1	0	0	Reserved
1	0	1	On-the-fly 1x/2x
1	1	0	On-the-fly 1x/4x
1	1	1	Reserved

There are two types of OTF modes (1x/2x and 1x/4x modes) that are selectable by programming the appropriate values into the mode register MR3 [8:6]. When either of the two OTF modes is selected, the device evaluates the BG0 bit when a REFRESH command is issued, and depending on the status of BG0, it dynamically switches its internal refresh configuration between 1x and 2x (or 1x and 4x) modes, and then executes the corresponding REFRESH operation.

Table 51: REFRESH Command Truth Table

Refresh	CS_n	ACT_n	RAS_n/A15	CAS_n/A14	WE_n/A13	BG1	BG0	A10/AP	A[9:0], A[12:11], A[20:16]	MR3[8:6]
Fixed rate	L	H	L	L	H	V	V	V	V	0vv
OTF: 1x	L	H	L	L	H	V	L	V	V	1vv
OTF: 2x	L	H	L	L	H	V	H	V	V	101
OTF: 4x	L	H	L	L	H	V	H	V	V	110

^{t}REFI and ^{t}RFC Parameters

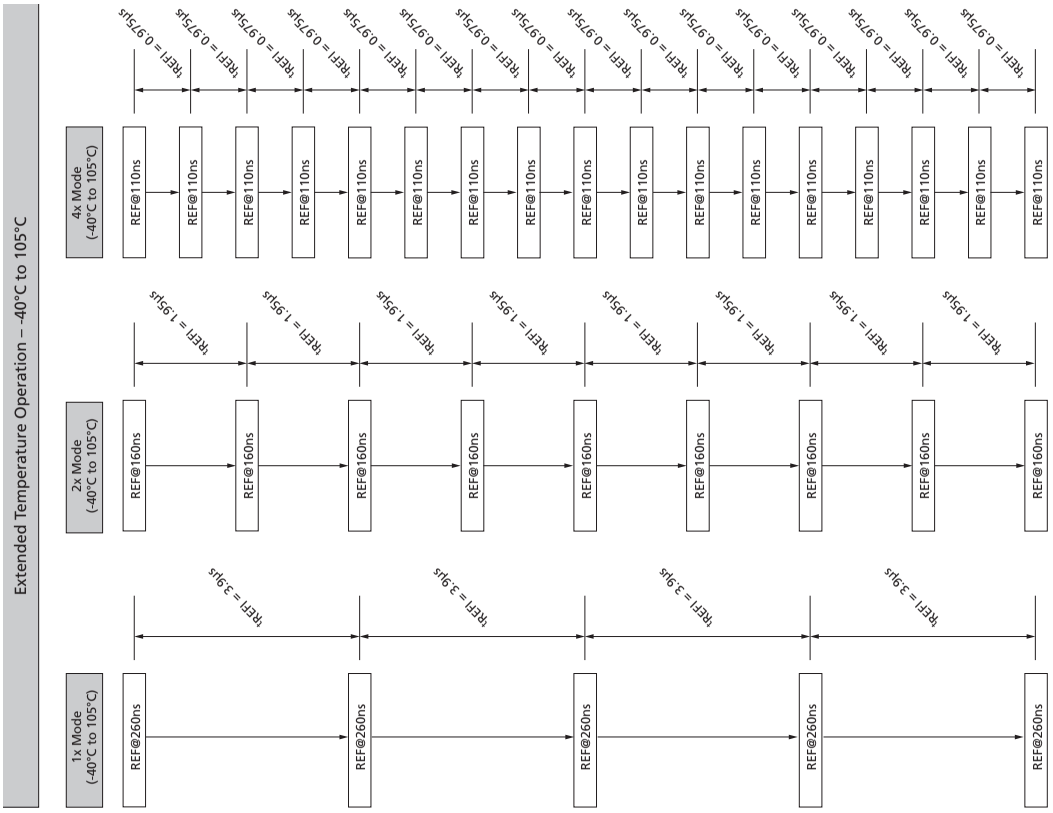
The default refresh rate mode is fixed 1x mode where REFRESH commands should be issued with the normal rate; that is, ^{t}REFI1 = ^{t}REFI(base) (for $T_C \leq 85°C$), and the duration of each REFRESH command is the normal REFRESH cycle time (^{t}RFC1). In 2x mode (either fixed 2x or OTF 2x mode), REFRESH commands should be issued to the device at the double frequency (^{t}REFI2 = ^{t}REFI(base)/2) of the normal refresh rate. In 4x mode, the REFRESH command rate should be quadrupled (^{t}REFI4 = ^{t}REFI(base)/4). Per each mode and command type, the ^{t}RFC parameter has different values as defined in the following table.

For discussion purposes, the REFRESH command that should be issued at the normal refresh rate and has the normal REFRESH cycle duration may be referred to as an REF1x command. The REFRESH command that should be issued at the double frequency (^{t}REFI2 = ^{t}REFI(base)/2) may be referred to as a REF2x command. Finally, the REFRESH command that should be issued at the quadruple rate (^{t}REFI4 = ^{t}REFI(base)/4) may be referred to as a REF4x command.

In the fixed 1x refresh rate mode, only REF1x commands are permitted. In the fixed 2x refresh rate mode, only REF2x commands are permitted. In the fixed 4x refresh rate mode, only REF4x commands are permitted. When the on-the-fly 1x/2x refresh rate mode is enabled, both REF1x and REF2x commands are permitted. When the OTF 1x/4x refresh rate mode is enabled, both REF1x and REF4x commands are permitted.

Table 52: ^{t}REFI and ^{t}RFC Parameters

Refresh Mode	Parameter		2Gb	4Gb	8Gb	16Gb	Units
	^{t}REFI (base)		7.8	7.8	7.8	7.8	µs
1x mode	^{t}REFI1	-40°C ≤ T_C ≤ 85°C	^{t}REFI(base)	^{t}REFI(base)	^{t}REFI(base)	^{t}REFI(base)	µs
		85°C ≤ T_C ≤ 95°C	^{t}REFI(base)/2	^{t}REFI(base)/2	^{t}REFI(base)/2	^{t}REFI(base)/2	µs
		95°C ≤ T_C ≤ 105°C	^{t}REFI(base)/4	^{t}REFI(base)/4	^{t}REFI(base)/4	^{t}REFI(base)/4	µs
	^{t}RFC1		160	260	350	350	ns
2x mode	^{t}REFI2	-40°C ≤ T_C ≤ 85°C	^{t}REFI(base)/2	^{t}REFI(base)/2	^{t}REFI(base)/2	^{t}REFI(base)/2	µs
		85°C ≤ T_C ≤ 95°C	^{t}REFI(base)/4	^{t}REFI(base)/4	^{t}REFI(base)/4	^{t}REFI(base)/4	µs
		95°C ≤ T_C ≤ 105°C	^{t}REFI(base)/8	^{t}REFI(base)/8	^{t}REFI(base)/8	^{t}REFI(base)/8	µs
	^{t}RFC2		110	160	260	260	ns
4x mode	^{t}REFI4	-40°C ≤ T_C ≤ 85°C	^{t}REFI(base)/4	^{t}REFI(base)/4	^{t}REFI(base)/4	^{t}REFI(base)/4	µs
		85°C ≤ T_C ≤ 95°C	^{t}REFI(base)/8	^{t}REFI(base)/8	^{t}REFI(base)/8	^{t}REFI(base)/8	µs
		95°C ≤ T_C ≤ 105°C	^{t}REFI(base)/16	^{t}REFI(base)/16	^{t}REFI(base)/16	^{t}REFI(base)/16	µs
	^{t}RFC4		90	110	160	160	ns



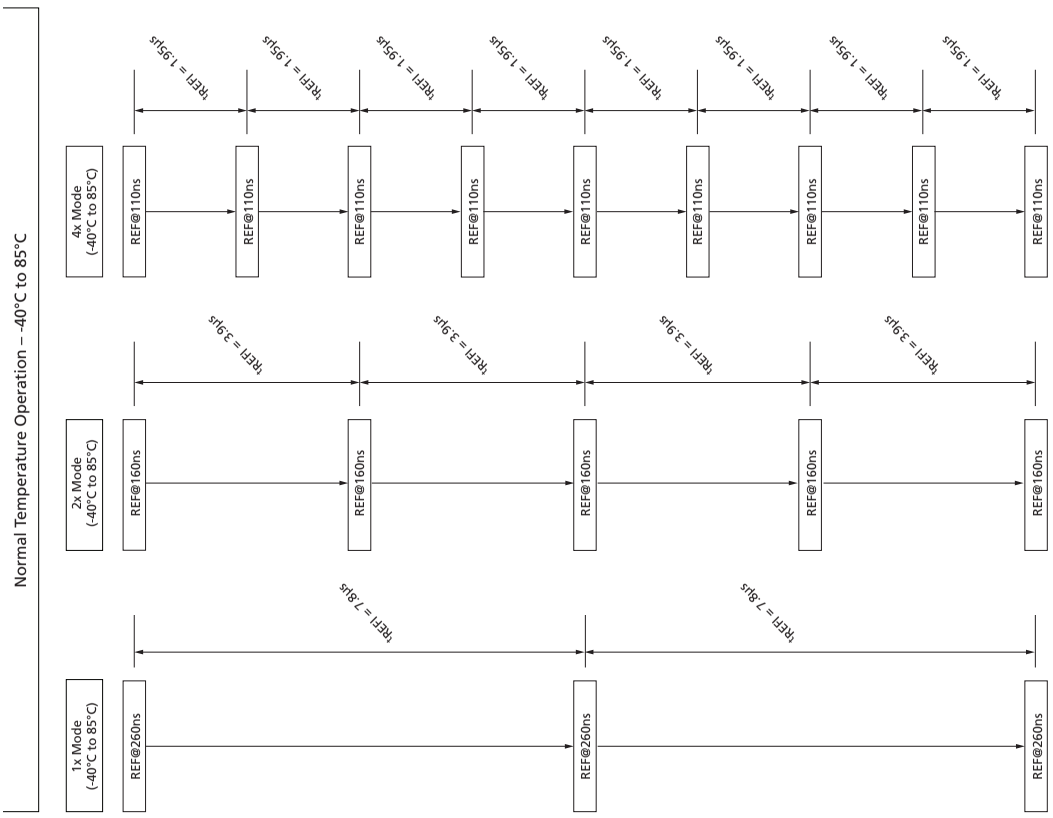
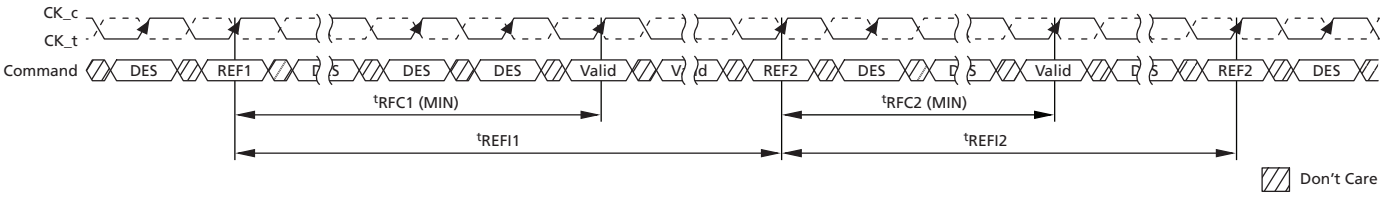


Figure 88: 4Gb with Fine Granularity Refresh Mode Example

Changing Refresh Rate

If the refresh rate is changed by either MRS or OTF. New ^{t}REFI and ^{t}RFC parameters will be applied from the moment of the rate change. When the REF1x command is issued to the DRAM, ^{t}REF1 and ^{t}RFC1 are applied from the time that the command was issued; when the REF2x command is issued, ^{t}REF2 and ^{t}RFC2 should be satisfied.

Figure 89: OTF REFRESH Command Timing



The following conditions must be satisfied before the refresh rate can be changed. Otherwise, data retention cannot be guaranteed.

- In the fixed 2x refresh rate mode or the OTF 1x/2x refresh mode, an even number of REF2x commands must be issued because the last change of the refresh rate mode with an MRS command before the refresh rate can be changed by another MRS command.
- In the OTF1x/2x refresh rate mode, an even number of REF2x commands must be issued between any two REF1x commands.
- In the fixed 4x refresh rate mode or the OTF 1x/4x refresh mode, a multiple-of-four number of REF4x commands must be issued because the last change of the refresh rate with an MRS command before the refresh rate can be changed by another MRS command.
- In the OTF1x/4x refresh rate mode, a multiple-of-four number of REF4x commands must be issued between any two REF1x commands.

There are no special restrictions for the fixed 1x refresh rate mode. Switching between fixed and OTF modes keeping the same rate is not regarded as a refresh rate change.

Usage with TCR Mode

If the temperature controlled refresh mode is enabled, only the normal mode (fixed 1x mode, MR3[8:6] = 000) is allowed. If any other refresh mode than the normal mode is selected, the temperature controlled refresh mode must be disabled.

Self Refresh Entry and Exit

The device can enter self refresh mode anytime in 1x, 2x, and 4x mode without any restriction on the number of REFRESH commands that have been issued during the mode before the self refresh entry. However, upon self refresh exit, extra REFRESH command(s) may be required, depending on the condition of the self refresh entry.

The conditions and requirements for the extra REFRESH command(s) are defined as follows:

- In the fixed 2x refresh rate mode or the enable-OTF 1x/2x refresh rate mode, it is recommended there be an even number of REF2x commands before entry into self refresh after the last self refresh exit, REF1x command, or MRS command that set the refresh mode. If this condition is met, no additional REFRESH commands are required upon self refresh exit. In the case that this condition is not met, either one extra REF1x command or two extra REF2x commands must be issued upon self refresh exit. These extra REFRESH commands are not counted toward the computation of the average refresh interval (^{t}REFI).

- In the fixed 4x refresh rate mode or the enable-OTF 1x/4x refresh rate mode, it is recommended there be a multiple-of-four number of REF4x commands before entry into self refresh after the last self refresh exit, REF1x command, or MRS command that set the refresh mode. If this condition is met, no additional refresh commands are required upon self refresh exit. When this condition is not met, either one extra REF1x command or four extra REF4x commands must be issued upon self refresh exit. These extra REFRESH commands are not counted toward the computation of the average refresh interval (^{t}REFI).

There are no special restrictions on the fixed 1x refresh rate mode.

This section does not change the requirement regarding postponed REFRESH commands. The requirement for the additional REFRESH command(s) described above is independent of the requirement for the postponed REFRESH commands.

SELF REFRESH Operation

The SELF REFRESH command can be used to retain data in the device, even if the rest of the system is powered down. When in self refresh mode, the device retains data without external clocking. The device has a built-in timer to accommodate SELF REFRESH operation. The SELF REFRESH command is defined by having CS_n, RAS_n, CAS_n, and CKE held LOW with WE_n and ACT_n HIGH at the rising edge of the clock.

Before issuing the SELF REFRESH ENTRY command, the device must be idle with all banks in the precharge state and ^{t}RP satisfied. Idle state is defined as: All banks are closed (^{t}RP, ^{t}DAL, and so on, satisfied), no data bursts are in progress, CKE is HIGH, and all timings from previous operations are satisfied (^{t}MRD, ^{t}MOD, ^{t}RFC, ^{t}ZQinit, ^{t}ZQoper, ^{t}ZQCS, and so on). After the SELF REFRESH ENTRY command is registered, CKE must be held LOW to keep the device in self refresh mode. The DRAM automatically disables ODT termination, regardless of the ODT pin, when it enters self refresh mode and automatically enables ODT upon exiting self refresh. During normal operation (DLL_on), the DLL is automatically disabled upon entering self refresh and is automatically enabled (including a DLL reset) upon exiting self refresh.

When the device has entered self refresh mode, all of the external control signals, except CKE and RESET_n, are "Don't Care." For proper SELF REFRESH operation, all power supply and reference pins (V_{DD}, V_{DDQ}, V_{SS}, V_{SSQ}, V_{PP}, and V_{REFCA}) must be at valid levels. The DRAM internal V_{REFDQ} generator circuitry may remain on or be turned off depending on the MR6 bit 7 setting. If the internal V_{REFDQ} circuit is on in self refresh, the first WRITE operation or first write-leveling activity may occur after ^{t}XS time after self refresh exit. If the DRAM internal V_{REFDQ} circuitry is turned off in self refresh, it ensures that the V_{REFDQ} generator circuitry is powered up and stable within the ^{t}XSDLL period when the DRAM exits the self refresh state. The first WRITE operation or first write-leveling activity may not occur earlier than ^{t}XSDLL after exiting self refresh. The device initiates a minimum of one REFRESH command internally within the ^{t}CKE period once it enters self refresh mode.

The clock is internally disabled during a SELF REFRESH operation to save power. The minimum time that the device must remain in self refresh mode is ^{t}CKESR/^{t}CKESR_PAR. The user may change the external clock frequency or halt the external clock ^{t}CKSRE/^{t}CKSRE_PAR after self refresh entry is registered; however, the clock must be restarted and ^{t}CKSRX must be stable before the device can exit SELF REFRESH operation.

The procedure for exiting self refresh requires a sequence of events. First, the clock must be stable prior to CKE going back HIGH. Once a SELF REFRESH EXIT command (SRX, combination of CKE going HIGH and DESELECT on the command bus) is registered, the following timing delay must be satisfied:

Commands that do not require locked DLL:

- ^{t}XS = ACT, PRE, PREA, REF, SRE, and PDE.
- ^{t}XS_FAST = ZQCL, ZQCS, and MRS commands. For an MRS command, only DRAM CL, WR/RTP register, and DLL reset in MR0; $R_{TT(NOM)}$ register in MR1; the CWL and $R_{TT(WR)}$ registers in MR2; and gear-down mode register in MR3; WRITE and READ preamble registers in MR4; $R_{TT(PARK)}$ register in MR5; Data rate and V_{REFDQ} calibration value registers in MR6 may be accessed provided the DRAM is not in per-DRAM mode. Access to other DRAM mode registers must satisfy ^{t}XS timing. WRITE commands (WR, WRS4, WRS8, WRA, WRAS4, and WRAS8) that require synchronous ODT and dynamic ODT controlled by the WRITE command require a locked DLL.

Commands that require locked DLL in the normal operating range:

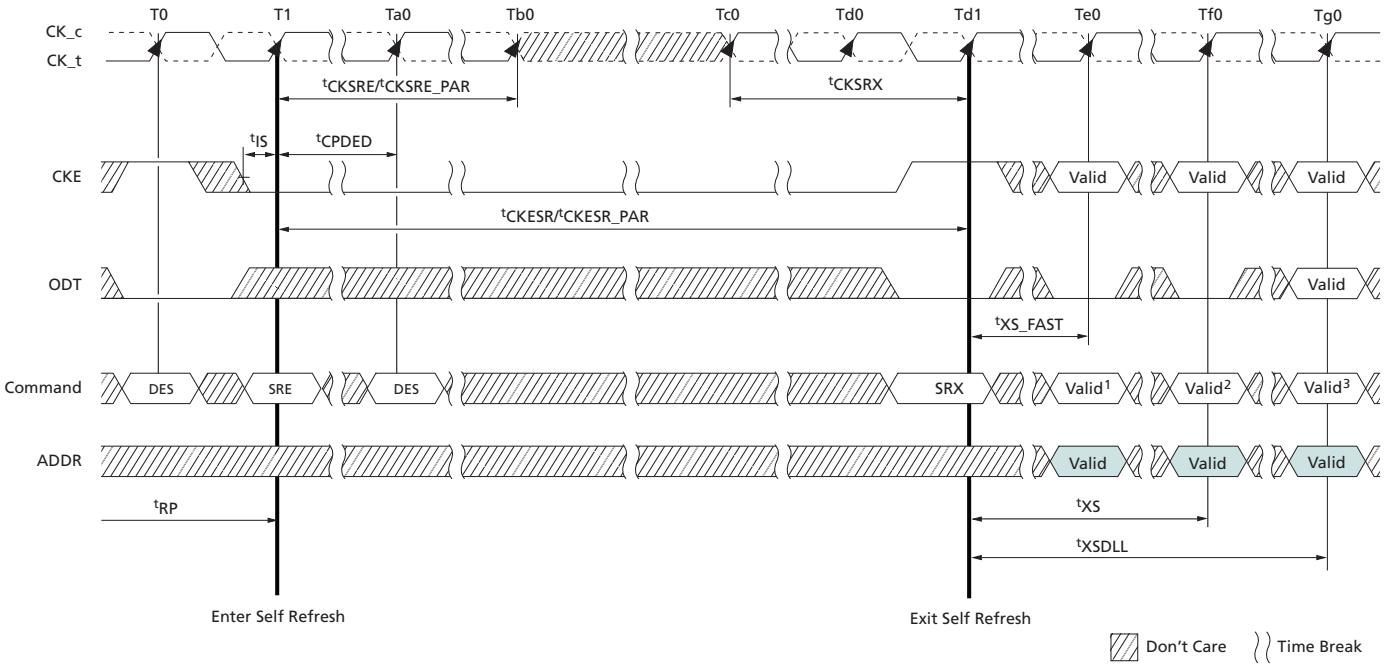
- ^{t}XSDLL – RD, RDS4, RDS8, RDA, RDAS4, and RDAS8 (unlike DDR3, WR, WRS4, WRS8, WRA, WRAS4, and WRAS8 because synchronous ODT is required).

Depending on the system environment and the amount of time spent in self refresh, ZQ CALIBRATION commands may be required to compensate for the voltage and temperature drift described in the ZQ CALIBRATION Commands section. To issue ZQ CALIBRATION commands, applicable timing requirements must be satisfied (see the ZQ Calibration Timing figure).

CKE must remain HIGH for the entire self refresh exit period ^{t}XSDLL for proper operation except for self refresh re-entry. Upon exit from self refresh, the device can be put back into self refresh mode or power-down mode after waiting at least ^{t}XS period and issuing one REFRESH command (refresh period of ^{t}RFC). The DESELECT command must be registered on each positive clock edge during the self refresh exit interval ^{t}XS. ODT must be turned off during ^{t}XSDLL.

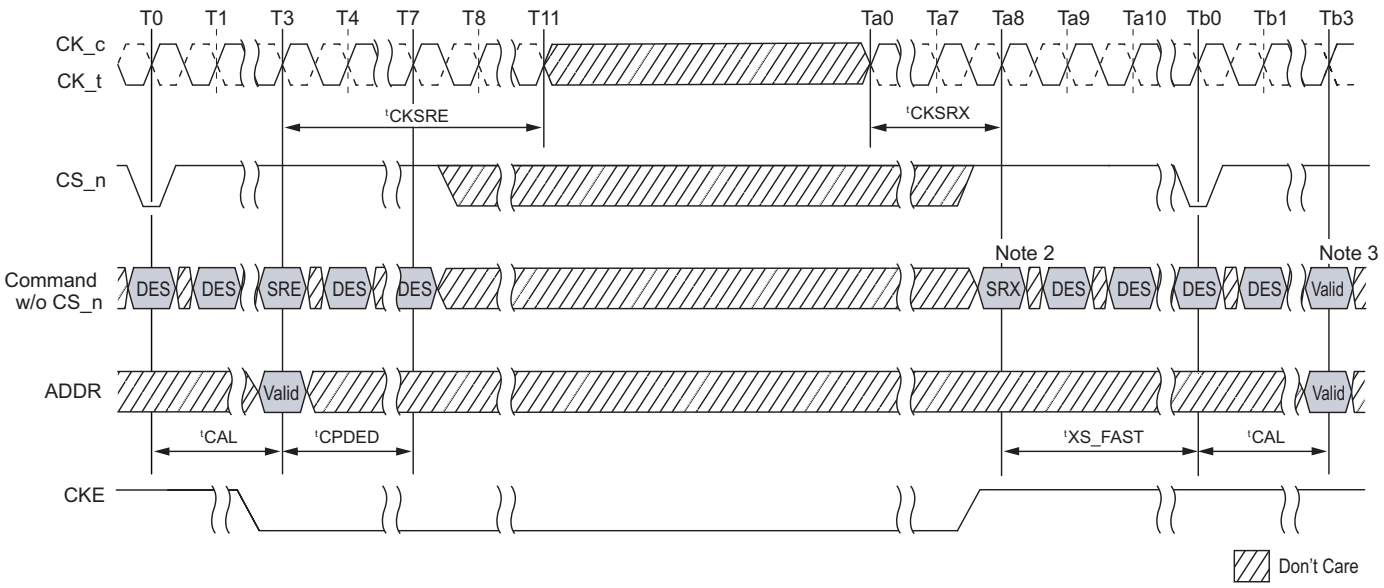
The use of self refresh mode introduces the possibility that an internally timed refresh event can be missed when CKE is raised for exit from self refresh mode. Upon exit from self refresh, the device requires a minimum of one extra REFRESH command before it is put back into self refresh mode.

Figure 90: Self Refresh Entry/Exit Timing



Notes: 1. Only MRS (limited to those described in the SELF REFRESH Operation section), ZQCS, or ZQCL commands are allowed.
2. Valid commands not requiring a locked DLL.
3. Valid commands requiring a locked DLL.

Figure 91: Self Refresh Entry/Exit Timing with CAL Mode

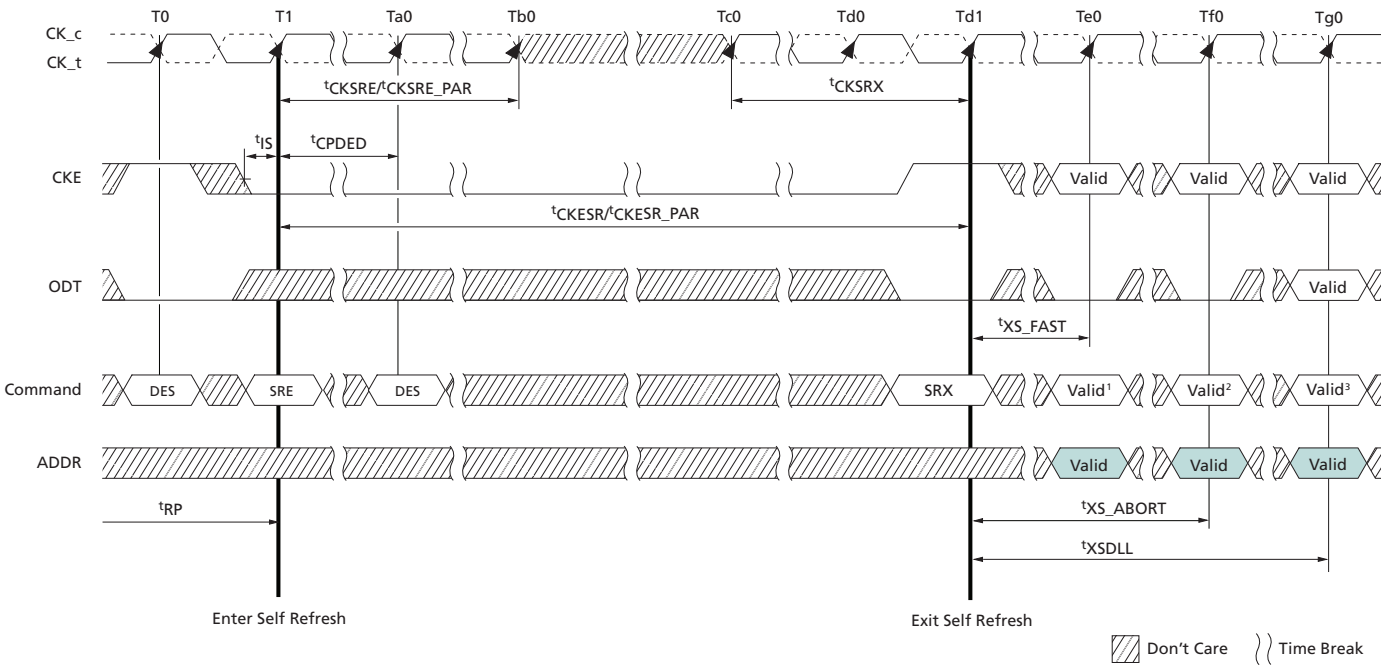


Notes: 1. ^{t}CAL = 3*n*CK, ^{t}CPDED = 4*n*CK, ^{t}CKSRE/^{t}CKSRE_PAR = 8*n*CK, ^{t}CKSRX = 8*n*CK, ^{t}XS_FAST = ^{t}REFC4 (MIN) + 10ns.
2. CS_n = HIGH, ACT_n = "Don't Care," RAS_n/A16 = "Don't Care," CAS_n/A15 = "Don't Care," WE_n/A14 = "Don't Care."
3. Only MRS (limited to those described in the SELF REFRESH Operations section), ZQCS, or ZQCL commands are allowed.
4. The figure only displays ^{t}XS_FAST timing, but ^{t}CAL must also be added to any ^{t}XS and ^{t}XSDLL associated commands during CAL mode.

Self Refresh Abort

The exit timing from self refresh exit to the first valid command not requiring a locked DLL is ^{t}XS. The value of ^{t}XS is (^{t}RFC1 + 10ns). This delay allows any refreshes started by the device time to complete. ^{t}RFC continues to grow with higher density devices, so ^{t}XS will grow as well. An MRS bit enables the self refresh abort mode. If the bit is disabled, the controller uses ^{t}XS timings (location MR4, bit 9). If the bit is enabled, the device aborts any ongoing refresh and does not increment the refresh counter. The controller can issue a valid command not requiring a locked DLL after a delay of ^{t}XS_ABORT. Upon exit from self refresh, the device requires a minimum of one extra REFRESH command before it is put back into self refresh mode. This requirement remains the same irrespective of the setting of the MRS bit for self refresh abort.

Figure 92: Self Refresh Abort



Notes: 1. Only MRS (limited to those described in the SELF REFRESH Operation section), ZQCS, or ZQCL commands are allowed.
2. Valid commands not requiring a locked DLL with self refresh abort mode enabled in the mode register.
3. Valid commands requiring a locked DLL.

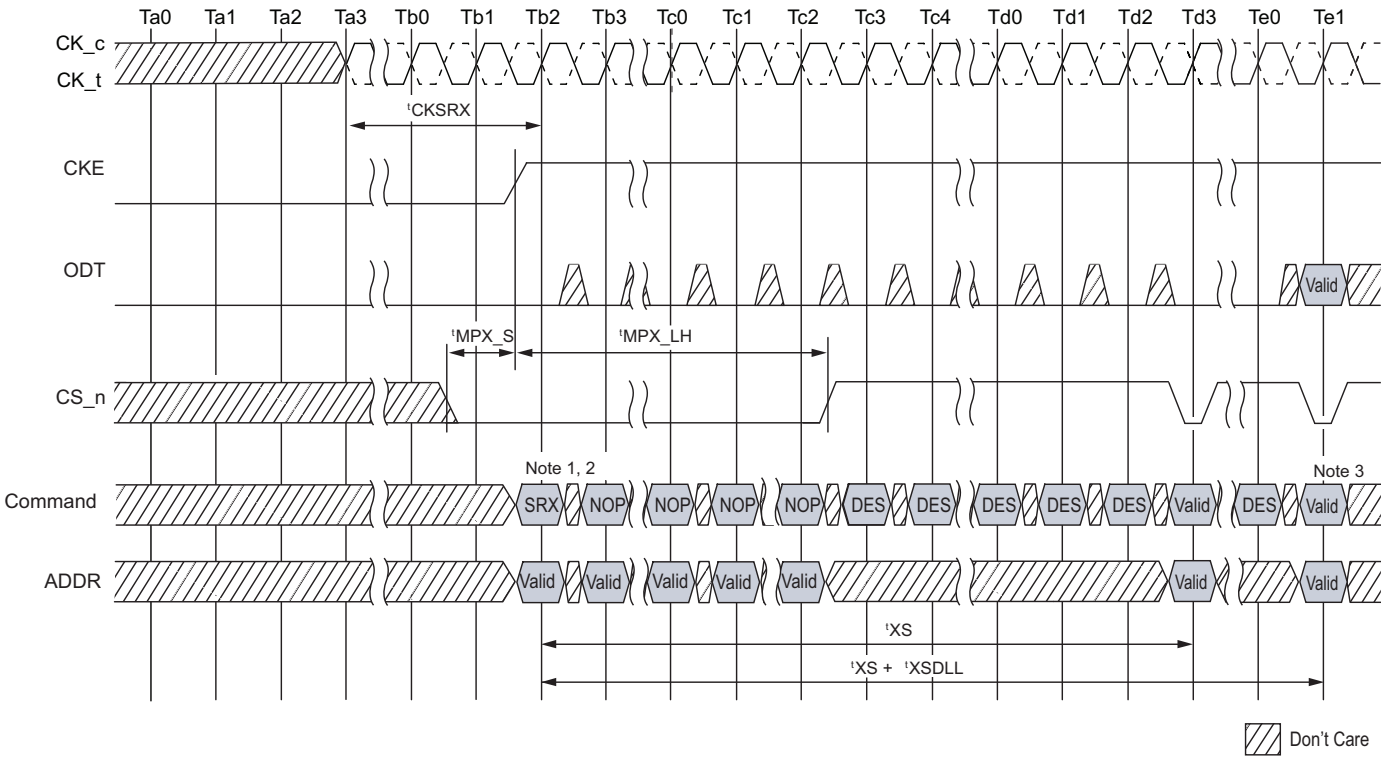
Self Refresh Exit with NOP Command

Exiting self refresh mode using the NO OPERATION command (NOP) is allowed under a specific system application. This special use of NOP allows for a common command/address bus between active DRAM devices and DRAM(s) in maximum power saving mode. Self refresh mode may exit with NOP commands provided:

- The device entered self refresh mode with CA parity, CAL, and gear-down disabled.
- ^{t}MPX_S and ^{t}MPX_LH are satisfied.
- NOP commands are only issued during ^{t}MPX_LH window.

No other command is allowed during the ^{t}MPX_LH window after an SELF REFRESH EXIT (SRX) command is issued.

Figure 93: Self Refresh Exit with NOP Command



Power-Down Mode

Power-down is synchronously entered when CKE is registered LOW (along with a DESELECT command). CKE is not allowed to go LOW when the following operations are in progress: MRS command, MPR operations, ZQCAL operations, DLL locking, or READ/WRITE operations. CKE is allowed to go LOW while any other operations, such as ROW ACTIVATION, PRECHARGE or auto precharge, or REFRESH, are in progress, but the power-down I_{DD} specification will not be applied until those operations are complete. The timing diagrams that follow illustrate power-down entry and exit.

For the fastest power-down exit timing, the DLL should be in a locked state when power-down is entered. If the DLL is not locked during power-down entry, the DLL must be reset after exiting power-down mode for proper READ operation and synchronous ODT operation. DRAM design provides all AC and DC timing and voltage specification as well as proper DLL operation with any CKE intensive operations as long as the controller complies with DRAM specifications.

During power-down, if all banks are closed after any in-progress commands are completed, the device will be in precharge power-down mode; if any bank is open after in-progress commands are completed, the device will be in active power-down mode.

Entering power-down deactivates the input and output buffers, excluding CK, CKE, and RESET_n. In power-down mode, DRAM ODT input buffer deactivation is based on Mode Register 5, bit 5 (MR5[5]). If it is configured to 0b, the ODT input buffer remains on and the ODT input signal must be at valid logic level. If it is configured to 1b, the ODT input buffer is deactivated and the DRAM ODT input signal may be floating and the device does not provide $R_{TT(NOM)}$ termination. Note that the device continues to provide $R_{TT(Park)}$ termination if it is enabled in MR5[8:6]. To protect internal delay on the CKE line to block the input signals, multiple DES commands are needed during the CKE switch off and on cycle(s); this timing period is defined as ^{t}CPDED. CKE LOW will result in deactivation of command and address receivers after ^{t}CPDED has expired.

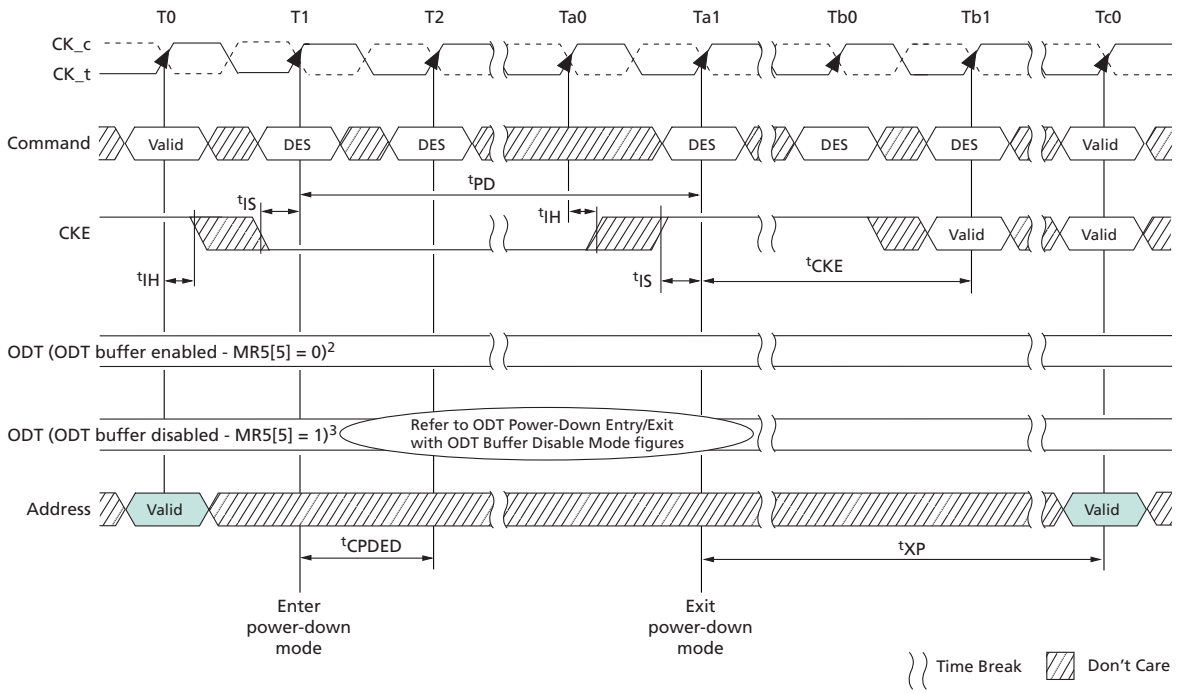
Table 53: Power-Down Entry Definitions

DRAM Status	DLL	Power-Down Exit	Relevant Parameters
Active (a bank or more open)	On	Fast	^{t}XP to any valid command.
Precharged (all banks precharged)	On	Fast	^{t}XP to any valid command.

The DLL is kept enabled during precharge power-down or active power-down. In power-down mode, CKE is LOW, RESET_n is HIGH, and a stable clock signal must be maintained at the inputs of the device. ODT should be in a valid state, but all other input signals are "Don't Care." (If RESET_n goes LOW during power-down, the device will be out of power-down mode and in the reset state.) CKE LOW must be maintained until ^{t}CKE has been satisfied. Power-down duration is limited by 9 × ^{t}REFI.

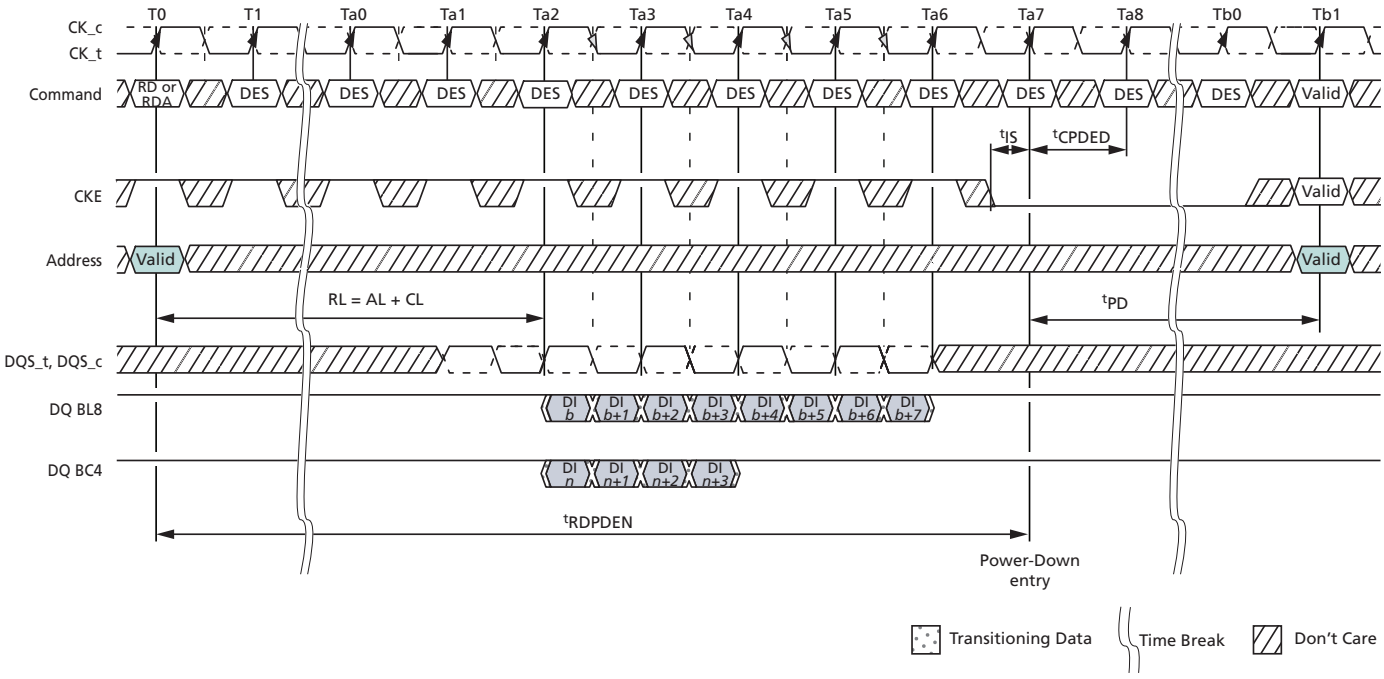
The power-down state is synchronously exited when CKE is registered HIGH (along with DES command). CKE HIGH must be maintained until ^{t}CKE has been satisfied. The ODT input signal must be at a valid level when the device exits from power-down mode, independent of MR1 bit [10:8] if $R_{TT(NOM)}$ is enabled in the mode register. If $R_{TT(NOM)}$ is disabled, the ODT input signal may remain floating. A valid, executable command can be applied with power-down exit latency, ^{t}XP, after CKE goes HIGH. Power-down exit latency is defined in the AC Specifications table.

Figure 94: Active Power-Down Entry and Exit



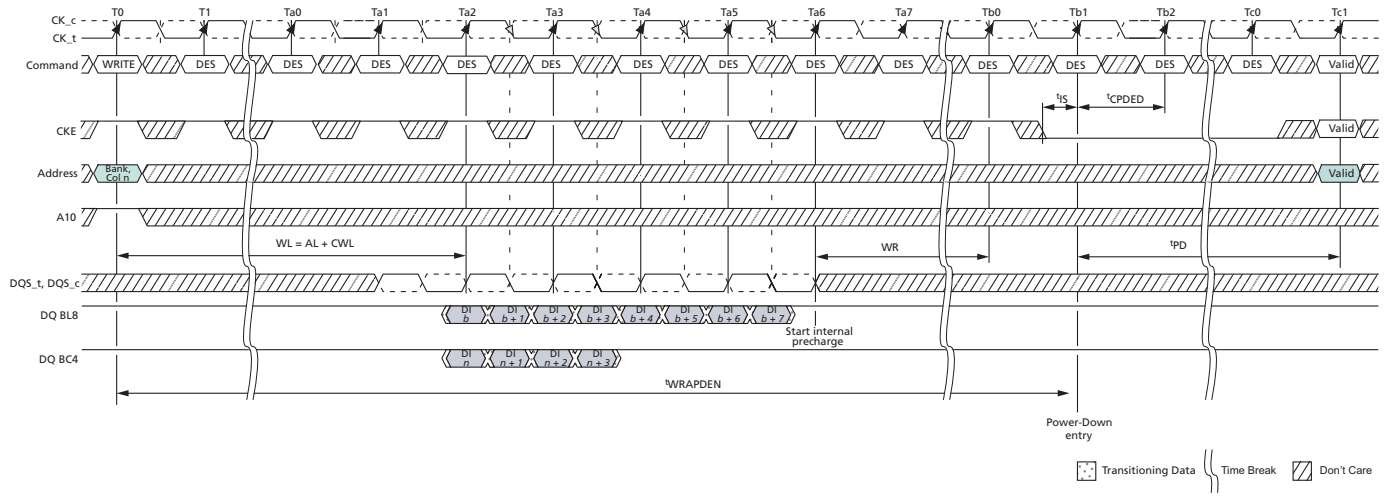
Notes: 1. Valid commands at T0 are ACT, DES, or PRE with one bank remaining open after completion of the PRECHARGE command.
2. ODT pin driven to a valid state; MR5[5] = 0 (normal setting).
3. ODT pin drive/float timing requirements for the ODT input buffer disable option (for additional power savings during active power-down) is described in the section for ODT Input Buffer Disable Mode for Power-Down; MR5[5] = 1.

Figure 95: Power-Down Entry After Read and Read with Auto Precharge



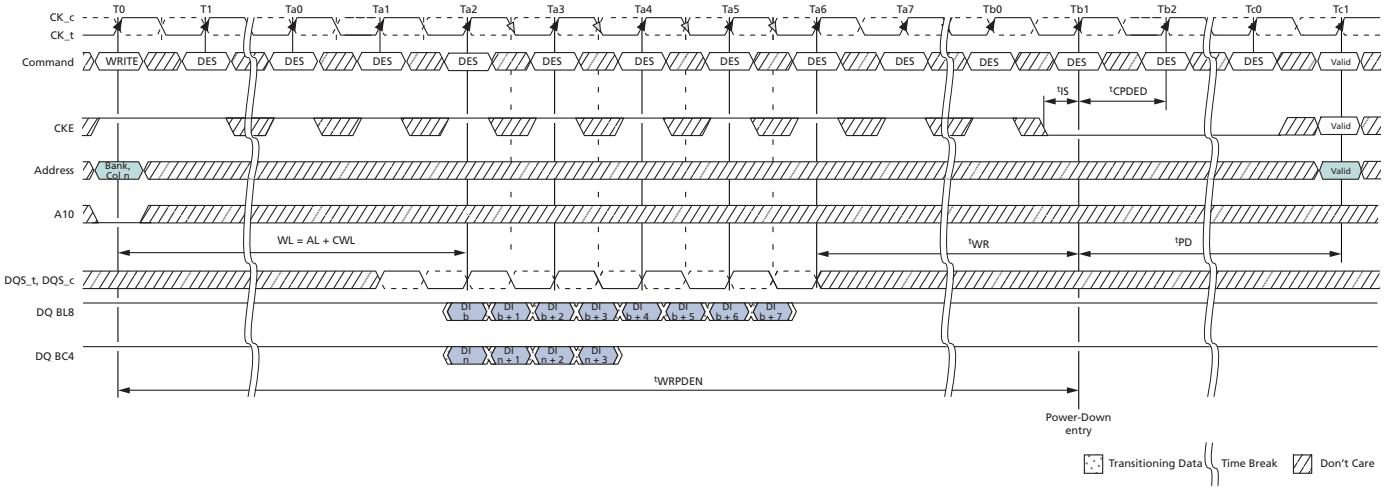
Note: 1. DI n (or b) = data-in from column n (or b).

Figure 96: Power-Down Entry After Write and Write with Auto Precharge



Notes: 1. DI n (or b) = data-in from column n (or b).
2. Valid commands at T0 are ACT, DES, or PRE with one bank remaining open after completion of the PRECHARGE command.

Figure 97: Power-Down Entry After Write



Note: 1. DI n (or b) = data-in from column n (or b).

Figure 98: Precharge Power-Down Entry and Exit

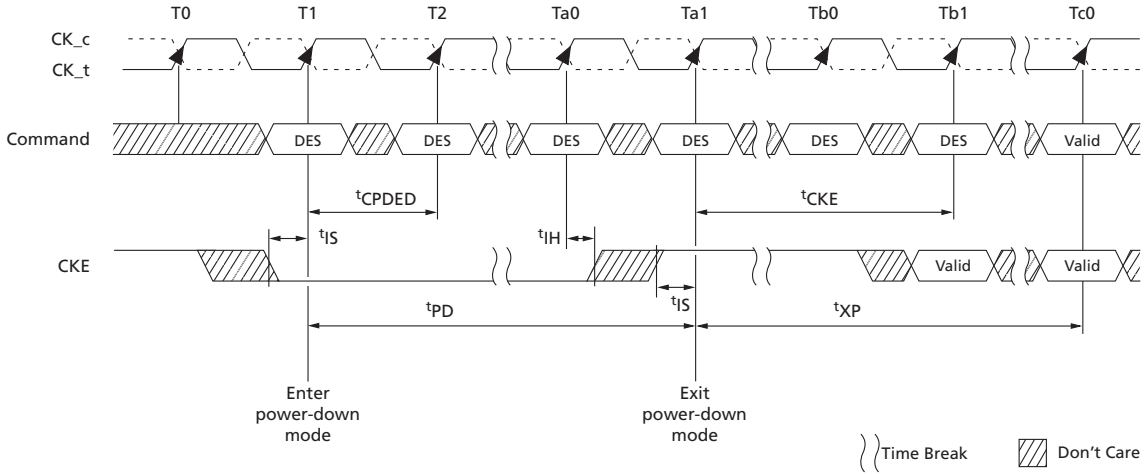


Figure 99: REFRESH Command to Power-Down Entry

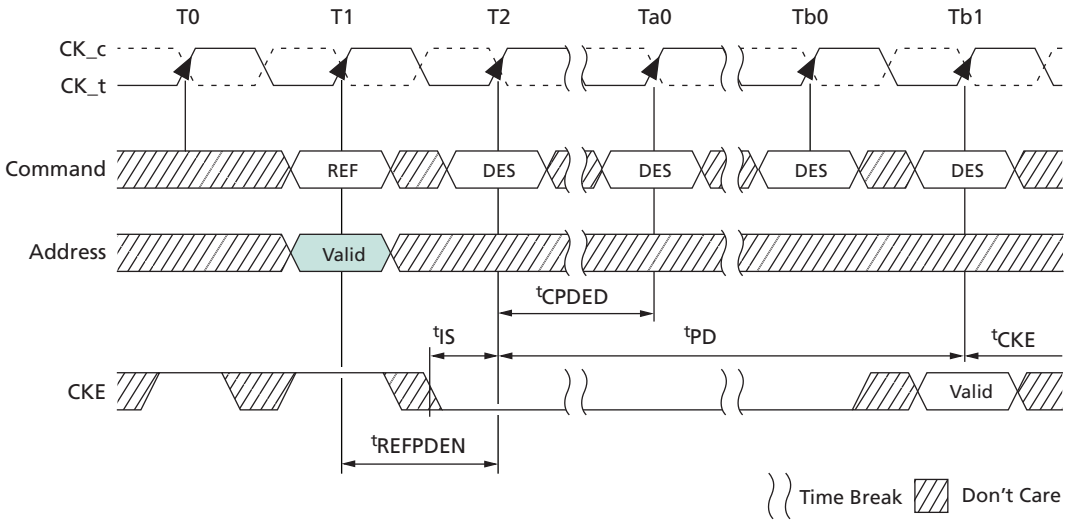


Figure 100: Active Command to Power-Down Entry

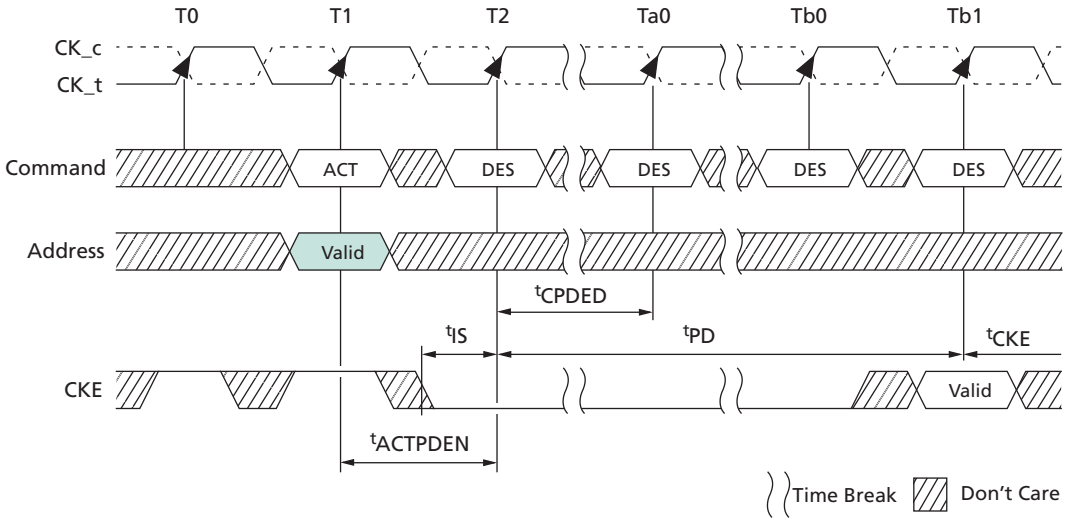


Figure 101: PRECHARGE/PRECHARGE ALL Command to Power-Down Entry

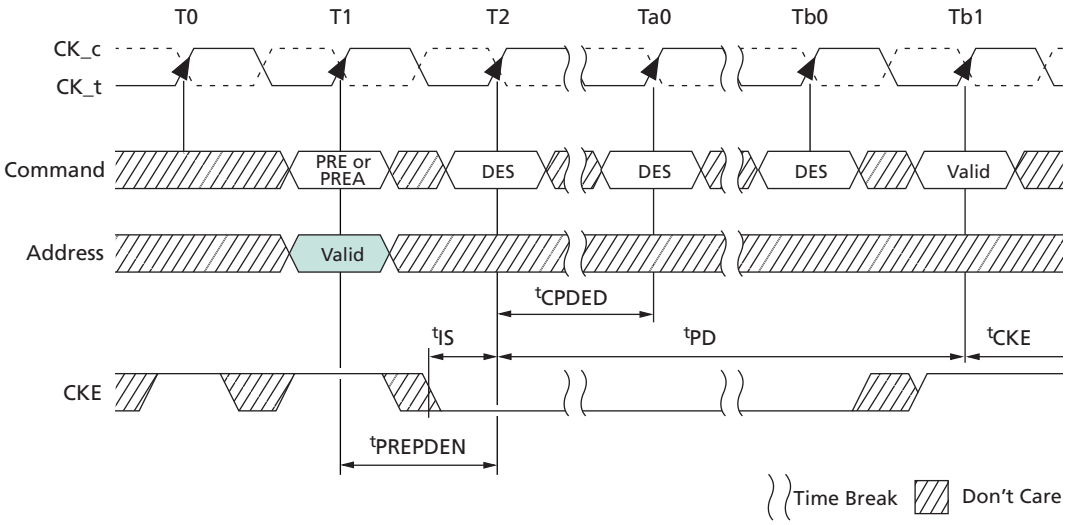
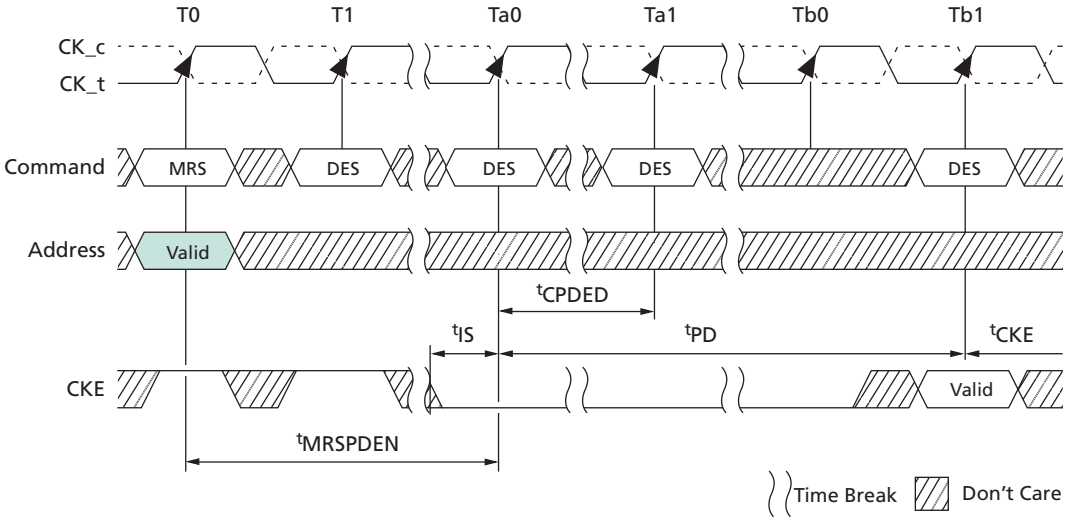


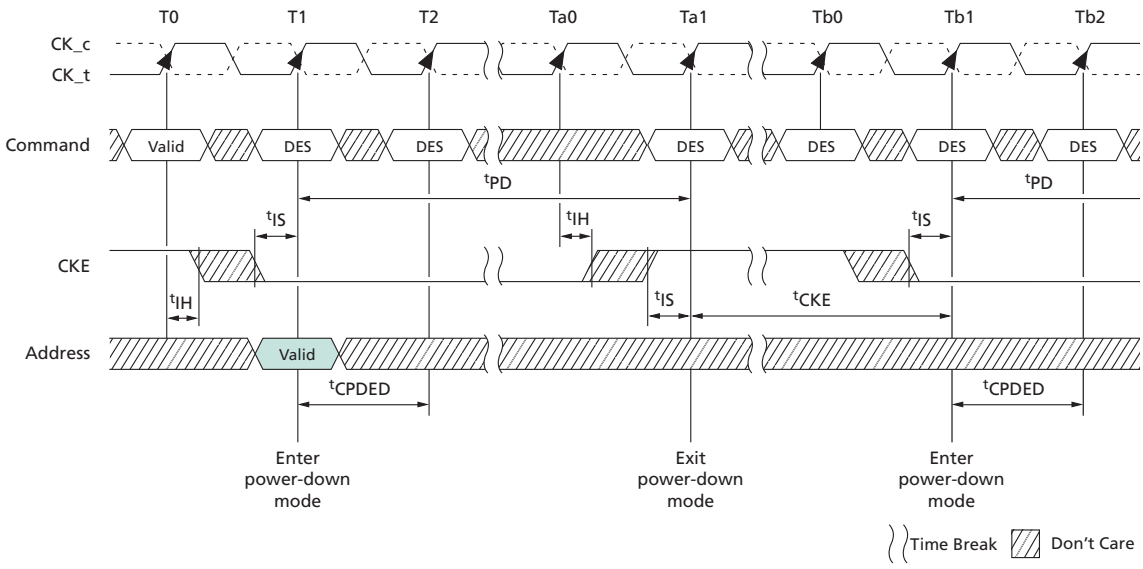
Figure 102: MRS Command to Power-Down Entry



Power-Down Clarifications – Case 1

When CKE is registered LOW for power-down entry, ^{t}PD (MIN) must be satisfied before CKE can be registered HIGH for power-down exit. The minimum value of parameter ^{t}PD (MIN) is equal to the minimum value of parameter ^{t}CKE (MIN) as shown in the Timing Parameters by Speed Bin table. A detailed example of Case 1 follows.

Figure 103: Power-Down Entry/Exit Clarifications – Case 1



Power-Down Entry, Exit Timing with CAL

Command/Address latency is used and additional timing restrictions are required when entering power-down, as noted in the following figures.

Figure 104: Active Power-Down Entry and Exit Timing with CAL

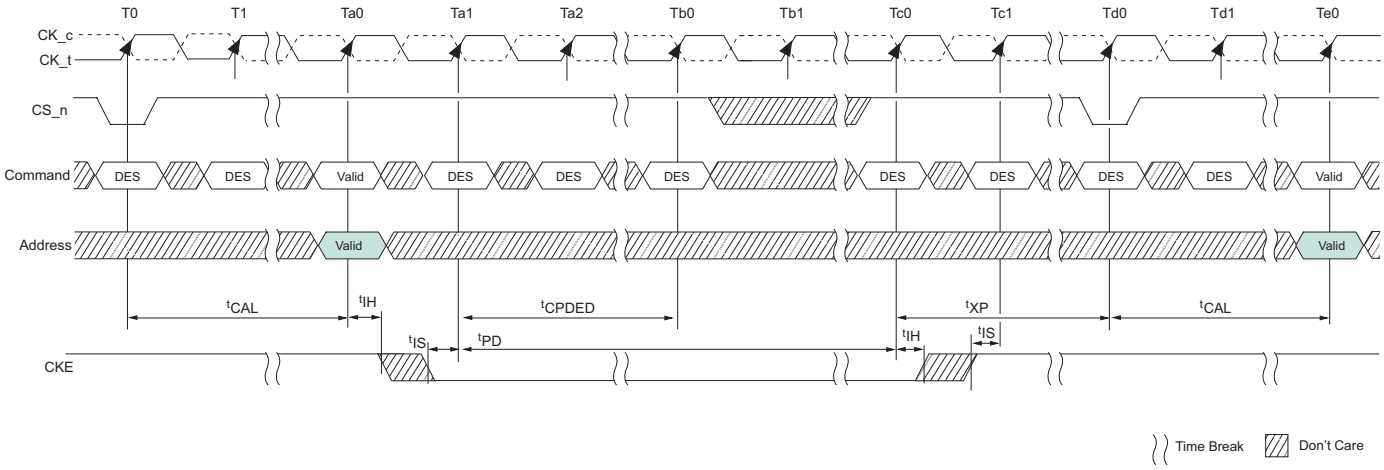
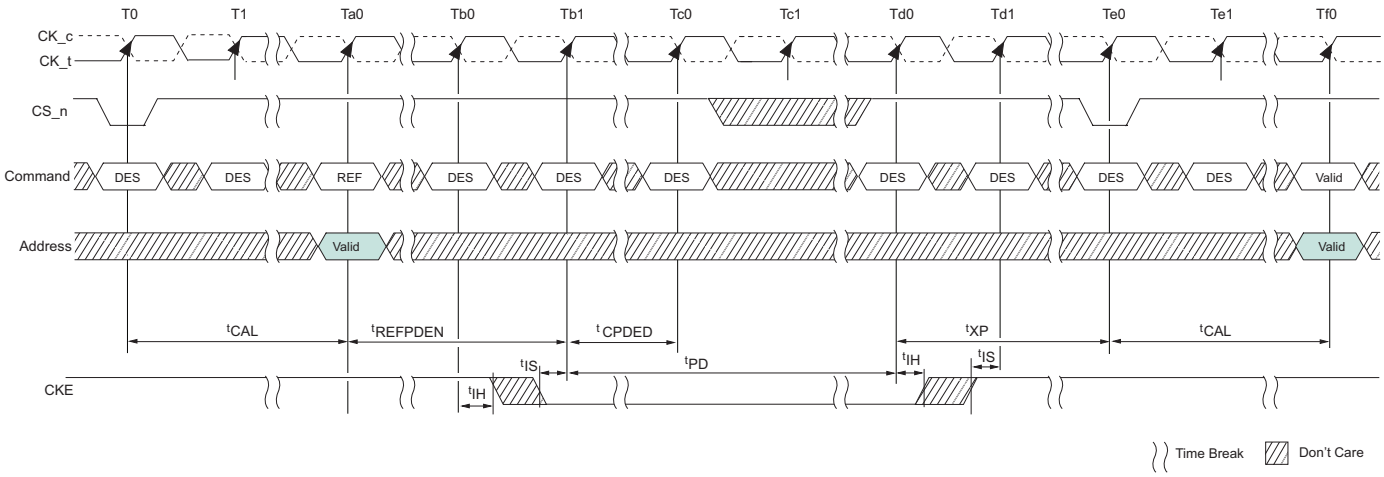


Figure 105: REFRESH Command to Power-Down Entry with CAL



ODT Input Buffer Disable Mode for Power-Down

DRAM does not provide R_{TT_NOM} termination during power-down when ODT input buffer deactivation mode is enabled in MR5 bit A5.

To account for DRAM internal delay on CKE line to disable the ODT buffer and block the sampled output, the host controller must continuously drive ODT to either low or high when entering power down (from ^{t}DODTLoff+1 prior to CKE low till ^{t}CPDED after CKE low).

The ODT signal is allowed to float after ^{t}CPDEDmin has expired. In this mode, R_{TT_NOM} termination corresponding to sampled ODT at the input when CKE is registered low (and ^{t}ANPD before that) may be either R_{TT_NOM} or R_{TT_PARK}. ^{t}ANPD is equal to (WL-1) and is counted backwards from PDE.

Figure 106: ODT Power-Down Entry with ODT Buffer Disable Mode

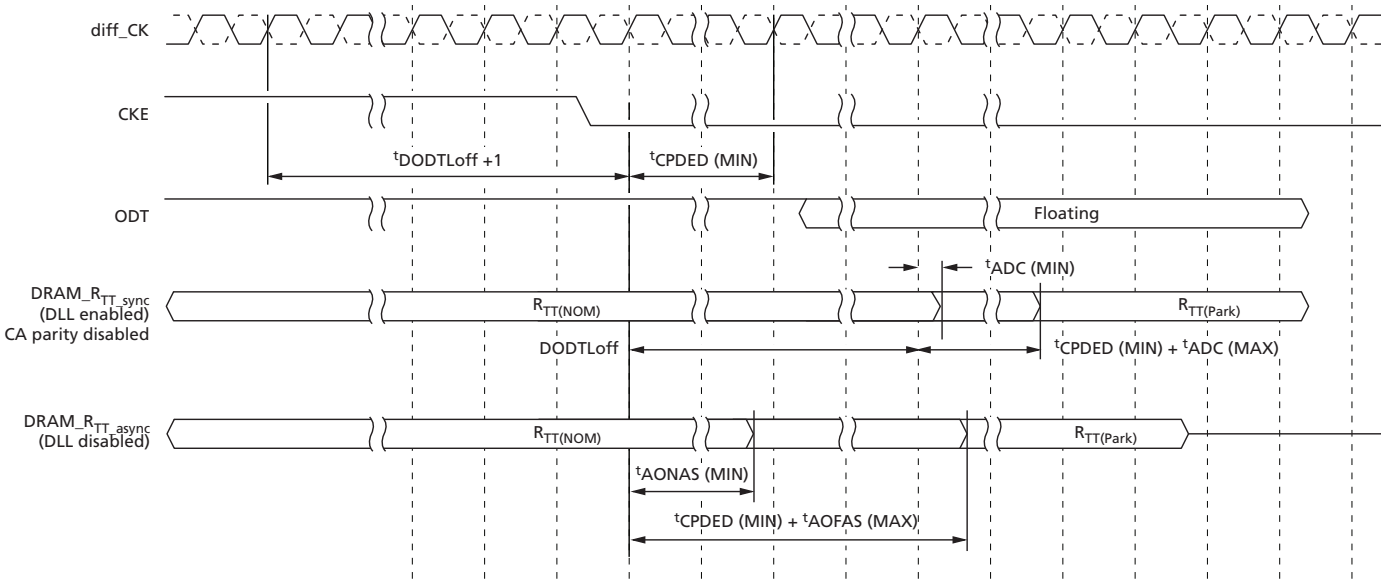
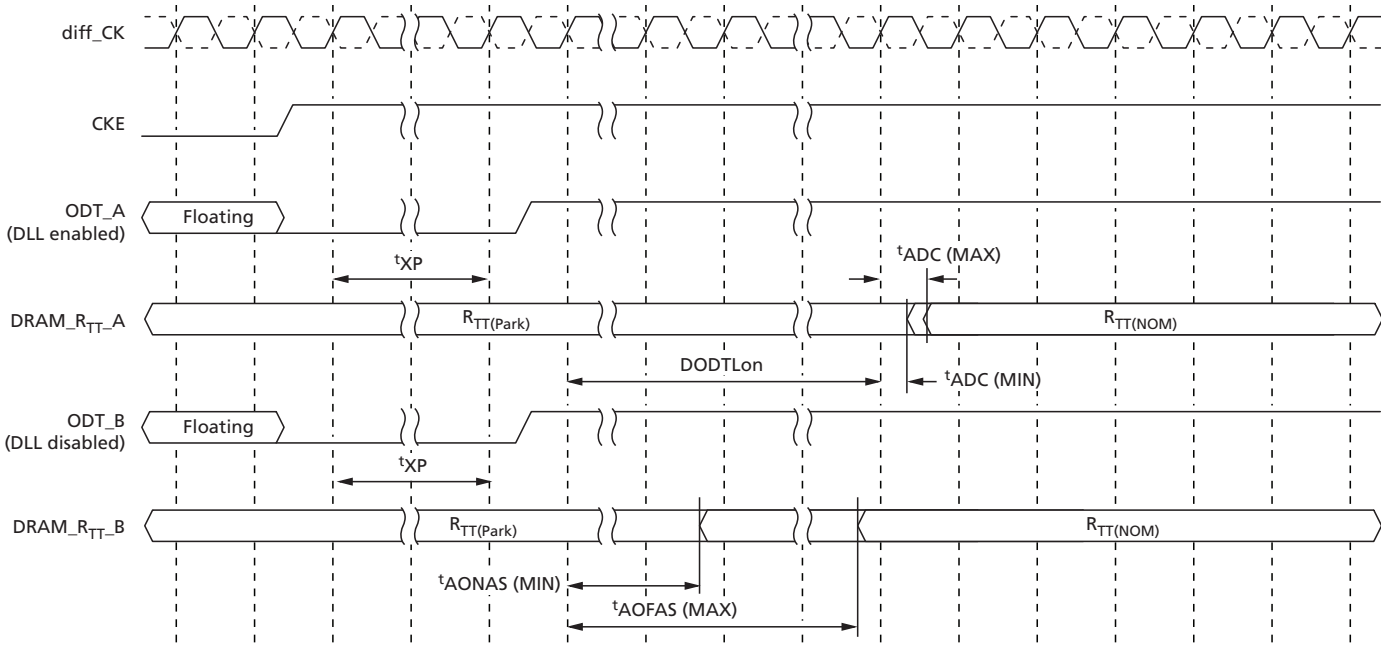


Figure 107: ODT Power-Down Exit with ODT Buffer Disable Mode

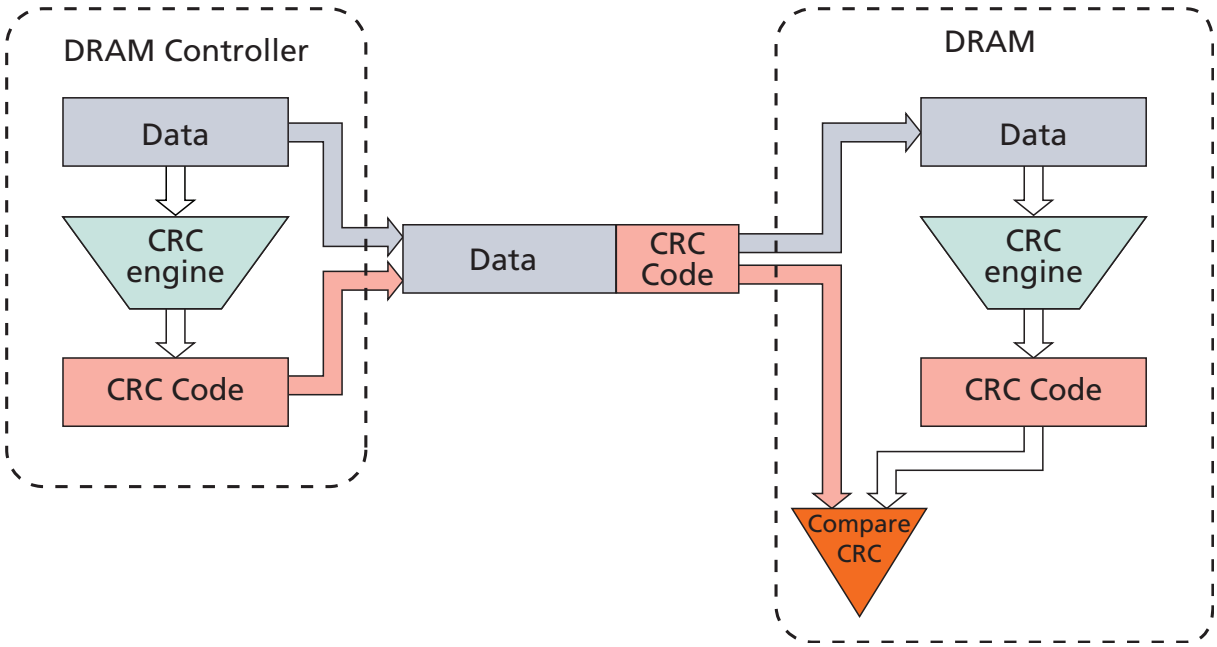


CRC Write Data Feature

CRC Write Data

The CRC write data feature takes the CRC generated data from the DRAM controller and compares it to the internally CRC generated data and determines whether the two match (no CRC error) or do not match (CRC error).

Figure 108: CRC Write Data Operation



WRITE CRC DATA Operation

A DRAM controller generates a CRC checksum using a 72-bit CRC tree and forms the write data frames, as shown in the following CRC data mapping tables for the x4, x8, and x16 configurations. A x4 device has a CRC tree with 32 input data bits used, and the remaining upper 40 bits D[71:32] being 1s. A x8 device has a CRC tree with 64 input data bits used, and the remaining upper 8 bits dependant upon whether DM_n/DBI_n is used (1s are sent when not used). A x16 device has two identical CRC trees each, one for the lower byte and one for the upper byte, with 64 input data bits used by each, and the remaining upper 8 bits on each byte dependant upon whether DM_n/DBI_n is used (1s are sent when not used). For a x8 and x16 DRAMs, the DRAM memory controller must send 1s in transfer 9 location whether or not DM_n/DBI_n is used.

The DRAM checks for an error in a received code word D[71:0] by comparing the received checksum against the computed checksum and reports errors using the ALERT_n signal if there is a mismatch. The DRAM can write data to the DRAM core without waiting for the CRC check for full writes when DM is disabled. If bad data is written to the DRAM core, the DRAM memory controller will try to overwrite the bad data with good data; this means the DRAM controller is responsible for data coherency when DM is disabled. However, in the case where both CRC and DM are enabled via MRS (that is, persistent mode), the DRAM will not write bad data to the core when a CRC error is detected.

DBI_n and CRC Both Enabled

The DRAM computes the CRC for received written data D[71:0]. Data is not inverted back based on DBI before it is used for computing CRC. The data is inverted back based on DBI before it is written to the DRAM core.

DM_n and CRC Both Enabled

When both DM and write CRC are enabled in the DRAM mode register, the DRAM calculates CRC before sending the write data into the array. If there is a CRC error, the DRAM blocks the WRITE operation and discards the data. If a CRC error is encountered from a WRITE with auto precharge (WRA), the DRAM will not block the precharge. The *Nonconsecutive WRITE (BL8/BC4-OTF) with 2tCK Preamble and Write CRC in Same or Different Bank Group* and the *WRITE (BL8/BC4-OTF/Fixed) with 1tCK Preamble and Write CRC in Same or Different BankGroup* figures in the WRITE Operation section show timing differences when DM is enabled.

DM_n and DBI_n Conflict During Writes with CRC Enabled

Both write DBI_n and DM_n can not be enabled at the same time; read DBI_n and DM_n can be enabled at the same time.

CRC and Write Preamble Restrictions

When write CRC is enabled:

- And 1tCK WRITE preamble mode is enabled, a tCCD_S or tCCD_L of 4 clocks is not allowed.
- And 2tCK WRITE preamble mode is enabled, a tCCD_S or tCCD_L of 6 clocks is not allowed.

CRC Simultaneous Operation Restrictions

When write CRC is enabled, neither MPR writes nor per-DRAM mode is allowed.

CRC Polynomial

The CRC polynomial used by DDR4 is the ATM-8 HEC, $X^8 + X^2 + X^1 + 1$.

A combinatorial logic block implementation of this 8-bit CRC for 72 bits of data includes 272 two-input XOR gates contained in eight 6-XOR-gate-deep trees.

The CRC polynomial and combinatorial logic used by DDR4 is the same as used on GDDR5.

The error coverage from the DDR4 polynomial used is shown in the following table.

Table 54: CRC Error Detection Coverage

Error Type	Detection Capability
Random single-bit errors	100%
Random double-bit errors	100%
Random odd count errors	100%
Random multibit UI vertical column error detection excluding DBI bits	100%

CRC Combinatorial Logic Equations

```
module CRC8_D72;
// polynomial: (0 1 2 8)
// data width: 72
// convention: the first serial data bit is D[71]
//initial condition all 0 implied
// "^" = XOR
function [7:0]
nextCRC8_D72;
input [71:0] Data;
input [71:0] D;
reg [7:0] CRC;
begin
D = Data;

CRC[0] =
D[69]^D[68]^D[67]^D[66]^D[64]^D[63]^D[60]^D[56]^D[54]^D[53]^D[52]^D[50]^D[49]^D[48]^D[45]
^D[43]^D[40]^D[39]^D[35]^D[34]^D[31]^D[30]^D[28]^D[23]^D[21]^D[19]^D[18]^D[16]^D[14]^D[1
2]^D[8]^D[7]^D[6]^D[0];


CRC[1] =
D[70]^D[66]^D[65]^D[63]^D[61]^D[60]^D[57]^D[56]^D[55]^D[52]^D[51]^D[48]^D[46]^D[45]^D[44]
^D[43]^D[41]^D[39]^D[36]^D[34]^D[32]^D[30]^D[29]^D[28]^D[24]^D[23]^D[22]^D[21]^D[20]^D[1
8]^D[17]^D[16]^D[15]^D[14]^D[13]^D[12]^D[9]^D[6]^D[1]^D[0];


CRC[2] =
D[71]^D[69]^D[68]^D[63]^D[62]^D[61]^D[60]^D[58]^D[57]^D[54]^D[50]^D[48]^D[47]^D[46]^D[44]
^D[43]^D[42]^D[39]^D[37]^D[34]^D[33]^D[29]^D[28]^D[25]^D[24]^D[22]^D[17]^D[15]^D[13]^D[1
2]^D[10]^D[8]^D[6]^D[2]^D[1]^D[0];


CRC[3] =
D[70]^D[69]^D[64]^D[63]^D[62]^D[61]^D[59]^D[58]^D[55]^D[51]^D[49]^D[48]^D[47]^D[45]^D[44]
^D[43]^D[40]^D[38]^D[35]^D[34]^D[30]^D[29]^D[26]^D[25]^D[23]^D[18]^D[16]^D[14]^D[13]^D[1
1]^D[9]^D[7]^D[3]^D[2]^D[1];


CRC[4] =
D[71]^D[70]^D[65]^D[64]^D[63]^D[62]^D[60]^D[59]^D[56]^D[52]^D[50]^D[49]^D[48]^D[46]^D[45]
^D[44]^D[41]^D[39]^D[36]^D[35]^D[31]^D[30]^D[27]^D[26]^D[24]^D[19]^D[17]^D[15]^D[14]^D[1
2]^D[10]^D[8]^D[4]^D[3]^D[2];
```

CRC[5] =
D[71]^D[66]^D[65]^D[64]^D[63]^D[61]^D[60]^D[57]^D[53]^D[51]^D[50]^D[49]^D[47]^D[46]^D[45]^D[42]^D[40]^D[37]^D[36]^D[32]^D[31]^D[28]^D[27]^D[25]^D[20]^D[18]^D[16]^D[15]^D[13]^D[11]^D[9]^D[5]^D[4]^D[3];

CRC[6] =
D[67]^D[66]^D[65]^D[64]^D[62]^D[61]^D[58]^D[54]^D[52]^D[51]^D[50]^D[48]^D[47]^D[46]^D[43]^D[41]^D[38]^D[37]^D[33]^D[32]^D[29]^D[28]^D[26]^D[21]^D[19]^D[17]^D[16]^D[14]^D[12]^D[10]^D[6]^D[5]^D[4];

CRC[7] =
D[68]^D[67]^D[66]^D[65]^D[63]^D[62]^D[59]^D[55]^D[53]^D[52]^D[51]^D[49]^D[48]^D[47]^D[44]^D[42]^D[39]^D[38]^D[34]^D[33]^D[30]^D[29]^D[27]^D[22]^D[20]^D[18]^D[17]^D[15]^D[13]^D[11]^D[7]^D[6]^D[5];

nextCRC8_D72 = CRC;

Burst Ordering for BL8

DDR4 supports fixed WRITE burst ordering [A2:A1:A0 = 0:0:0] when write CRC is enabled in BL8 (fixed).

CRC Data Bit Mapping

Table 55: CRC Data Mapping for x4 Devices, BL8

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
DQ0	D0	D1	D2	D3	D4	D5	D6	D7	CRC0	CRC4
DQ1	D8	D9	D10	D11	D12	D13	D14	D15	CRC1	CRC5
DQ2	D16	D17	D18	D19	D20	D21	D22	D23	CRC2	CRC6
DQ3	D24	D25	D26	D27	D28	D29	D30	D31	CRC3	CRC7

Table 56: CRC Data Mapping for x8 Devices, BL8

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
DQ0	D0	D1	D2	D3	D4	D5	D6	D7	CRC0	1
DQ1	D8	D9	D10	D11	D12	D13	D14	D15	CRC1	1
DQ2	D16	D17	D18	D19	D20	D21	D22	D23	CRC2	1
DQ3	D24	D25	D26	D27	D28	D29	D30	D31	CRC3	1
DQ4	D32	D33	D34	D35	D36	D37	D38	D39	CRC4	1

Table 56: CRC Data Mapping for x8 Devices, BL8 (Continued)

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
DQ5	D40	D41	D42	D43	D44	D45	D46	D47	CRC5	1
DQ6	D48	D49	D50	D51	D52	D53	D54	D55	CRC6	1
DQ7	D56	D57	D58	D59	D60	D61	D62	D63	CRC7	1
DM_n/DBI_n	D64	D65	D66	D67	D68	D69	D70	D71	1	1

A x16 device is treated as two x8 devices; a x16 device will have two identical CRC trees implemented. CRC[7:0] covers data bits D[71:0], and CRC[15:8] covers data bits D[143:72].

Table 57: CRC Data Mapping for x16 Devices, BL8

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
DQ0	D0	D1	D2	D3	D4	D5	D6	D7	CRC0	1
DQ1	D8	D9	D10	D11	D12	D13	D14	D15	CRC1	1
DQ2	D16	D17	D18	D19	D20	D21	D22	D23	CRC2	1
DQ3	D24	D25	D26	D27	D28	D29	D30	D31	CRC3	1
DQ4	D32	D33	D34	D35	D36	D37	D38	D39	CRC4	1
DQ5	D40	D41	D42	D43	D44	D45	D46	D47	CRC5	1
DQ6	D48	D49	D50	D51	D52	D53	D54	D55	CRC6	1
DQ7	D56	D57	D58	D59	D60	D61	D62	D63	CRC7	1
LDM_n/LDBI_n	D64	D65	D66	D67	D68	D69	D70	D71	1	1
DQ8	D72	D73	D74	D75	D76	D77	D78	D79	CRC8	1
DQ9	D80	D81	D82	D83	D84	D85	D86	D87	CRC9	1
DQ10	D88	D89	D90	D91	D92	D93	D94	D95	CRC10	1
DQ11	D96	D97	D98	D99	D100	D101	D102	D103	CRC11	1
DQ12	D104	D105	D106	D107	D108	D109	D110	D111	CRC12	1
DQ13	D112	D113	D114	D115	D116	D117	D118	D119	CRC13	1
DQ14	D120	D121	D122	D123	D124	D125	D126	D127	CRC14	1
DQ15	D128	D129	D130	D131	D132	D133	D134	D135	CRC15	1
UDM_n/UDBI_n	D136	D137	D138	D139	D140	D141	D142	D143	1	1

CRC Enabled With BC4

If CRC and BC4 are both enabled, then address bit A2 is used to transfer critical data first for BC4 writes.

CRC with BC4 Data Bit Mapping

For a x4 device, the CRC tree inputs are 16 data bits, and the inputs for the remaining bits are 1.

When A2 = 1, data bits D[7:4] are used as inputs for D[3:0], D[15:12] are used as inputs to D[11:8], and so forth, for the CRC tree.

Table 58: CRC Data Mapping for x4 Devices, BC4

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
A2 = 0										
DQ0	D0	D1	D2	D3	1	1	1	1	CRC0	CRC4
DQ1	D8	D9	D10	D11	1	1	1	1	CRC1	CRC5
DQ2	D16	D17	D18	D19	1	1	1	1	CRC2	CRC6
DQ3	D24	D25	D26	D27	1	1	1	1	CRC3	CRC7
A2 = 1										
DQ0	D4	D5	D6	D7	1	1	1	1	CRC0	CRC4
DQ1	D12	D13	D14	D15	1	1	1	1	CRC1	CRC5
DQ2	D20	D21	D22	D23	1	1	1	1	CRC2	CRC6
DQ3	D28	D29	D30	D31	1	1	1	1	CRC3	CRC7

For a x8 device, the CRC tree inputs are 36 data bits.

When A2 = 0, the input bits D[67:64]) are used if DBI_n or DM_n functions are enabled; if DBI_n and DM_n are disabled, then D[67:64]) are 1.

When A2 = 1, data bits D[7:4] are used as inputs for D[3:0], D[15:12] are used as inputs to D[11:8], and so forth, for the CRC tree. The input bits D[71:68]) are used if DBI_n or DM_n functions are enabled; if DBI_n and DM_n are disabled, then D[71:68]) are 1.

Table 59: CRC Data Mapping for x8 Devices, BC4

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
A2 = 0										
DQ0	D0	D1	D2	D3	1	1	1	1	CRC0	1
DQ1	D8	D9	D10	D11	1	1	1	1	CRC1	1
DQ2	D16	D17	D18	D19	1	1	1	1	CRC2	1
DQ3	D24	D25	D26	D27	1	1	1	1	CRC3	1
DQ4	D32	D33	D34	D35	1	1	1	1	CRC4	1
DQ5	D40	D41	D42	D43	1	1	1	1	CRC5	1
DQ6	D48	D49	D50	D51	1	1	1	1	CRC6	1
DQ7	D56	D57	D58	D59	1	1	1	1	CRC7	1
DM_n/DBI_n	D64	D65	D66	D67	1	1	1	1	1	1
A2 = 1										
DQ0	D4	D5	D6	D7	1	1	1	1	CRC0	1
DQ1	D12	D13	D14	D15	1	1	1	1	CRC1	1
DQ2	D20	D21	D22	D23	1	1	1	1	CRC2	1
DQ3	D28	D29	D30	D31	1	1	1	1	CRC3	1

Table 59: CRC Data Mapping for x8 Devices, BC4 (Continued)

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
DQ4	D36	D37	D38	D39	1	1	1	1	CRC4	1
DQ5	D44	D45	D46	D47	1	1	1	1	CRC5	1
DQ6	D52	D53	D54	D55	1	1	1	1	CRC6	1
DQ7	D60	D61	D62	D63	1	1	1	1	CRC7	1
DM_n/DBI_n	D68	D69	D70	D71	1	1	1	1	1	1

There are two identical CRC trees for x16 devices, each have CRC tree inputs of 36 bits.

When A2 = 0, input bits D[67:64] are used if DBI_n or DM_n functions are enabled; if DBI_n and DM_n are disabled, then D[67:64] are 1s. The input bits D[139:136] are used if DBI_n or DM_n functions are enabled; if DBI_n and DM_n are disabled, then D[139:136] are 1s.

When A2 = 1, data bits D[7:4] are used as inputs for D[3:0], D[15:12] are used as inputs for D[11:8], and so forth, for the CRC tree. Input bits D[71:68] are used if DBI_n or DM_n functions are enabled; if DBI_n and DM_n are disabled, then D[71:68] are 1s. The input bits D[143:140] are used if DBI_n or DM_n functions are enabled; if DBI_n and DM_n are disabled, then D[143:140] are 1s.

Table 60: CRC Data Mapping for x16 Devices, BC4

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
A2 = 0										
DQ0	D0	D1	D2	D3	1	1	1	1	CRC0	1
DQ1	D8	D9	D10	D11	1	1	1	1	CRC1	1
DQ2	D16	D17	D18	D19	1	1	1	1	CRC2	1
DQ3	D24	D25	D26	D27	1	1	1	1	CRC3	1
DQ4	D32	D33	D34	D35	1	1	1	1	CRC4	1
DQ5	D40	D41	D42	D43	1	1	1	1	CRC5	1
DQ6	D48	D49	D50	D51	1	1	1	1	CRC6	1
DQ7	D56	D57	D58	D59	1	1	1	1	CRC7	1
LDM_n/LDBI_n	D64	D65	D66	D67	1	1	1	1	1	1
DQ8	D72	D73	D74	D75	1	1	1	1	CRC8	1
DQ9	D80	D81	D82	D83	1	1	1	1	CRC9	1
DQ10	D88	D89	D90	D91	1	1	1	1	CRC10	1
DQ11	D96	D97	D98	D99	1	1	1	1	CRC11	1
DQ12	D104	D105	D106	D107	1	1	1	1	CRC12	1
DQ13	D112	D113	D114	D115	1	1	1	1	CRC13	1
DQ14	D120	D121	D122	D123	1	1	1	1	CRC14	1
DQ15	D128	D129	D130	D131	1	1	1	1	CRC15	1
UDM_n/UDBI_n	D136	D137	D138	D139	1	1	1	1	1	1

Table 60: CRC Data Mapping for x16 Devices, BC4 (Continued)

Function	Transfer									
	0	1	2	3	4	5	6	7	8	9
A2 = 1										
DQ0	D4	D5	D6	D7	1	1	1	1	CRC0	1
DQ1	D12	D13	D14	D15	1	1	1	1	CRC1	1
DQ2	D20	D21	D22	D23	1	1	1	1	CRC2	1
DQ3	D28	D29	D30	D31	1	1	1	1	CRC3	1
DQ4	D36	D37	D38	D39	1	1	1	1	CRC4	1
DQ5	D44	D45	D46	D47	1	1	1	1	CRC5	1
DQ6	D52	D53	D54	D55	1	1	1	1	CRC6	1
DQ7	D60	D61	D62	D63	1	1	1	1	CRC7	1
LDM_n/LDBI_n	D68	D69	D70	D71	1	1	1	1	1	1
DQ8	D76	D77	D78	D79	1	1	1	1	CRC8	1
DQ9	D84	D85	D86	D87	1	1	1	1	CRC9	1
DQ10	D92	D93	D94	D95	1	1	1	1	CRC10	1
DQ11	D100	D101	D102	D103	1	1	1	1	CRC11	1
DQ12	D108	D109	D110	D111	1	1	1	1	CRC12	1
DQ13	D116	D117	D118	D119	1	1	1	1	CRC13	1
DQ14	D124	D125	D126	D127	1	1	1	1	CRC14	1
DQ15	D132	D133	D134	D135	1	1	1	1	CRC15	1
UDM_n/UDBI_n	D140	D141	D142	D143	1	1	1	1	1	1

CRC Equations for x8 Device in BC4 Mode with A2 = 0 and A2 = 1

The following example is of a CRC tree when x8 is used in BC4 mode (x4 and x16 CRC trees have similar differences).

CRC[0], A2=0 =
1^1^D[67]^D[66]^D[64]^1^1^D[56]^1^1^1^D[50]^D[49]^D[48]^1^D[43]^D[40]^1^D[35]^D[34]^1^1^1^1^1^D[19]^D[18]^D[16]^1^1^D[8] ^1^1^ D[0] ;

CRC[0], A2=1=
1^1^D[71]^D[70]^D[68]^1^1^D[60]^1^1^1^D[54]^D[53]^D[52]^1^D[47]^D[44]^1^D[39]^D[38]^1^1^1^1^1^D[23]^D[22]^D[20]^1^1^D[12]^1^1^D[4] ;

CRC[1], A2=0 =
1^D[66]^D[65]^1^1^1^D[57]^D[56]^1^1^D[51]^D[48]^1^1^1^D[43]^D[41]^1^1^D[34]^D[32]^1^1^1^D[24]^1^1^1^1^D[18]^D[17]^D[16]^1^1^1^1^D[9] ^1^ D[1]^D[0];

CRC[1], A2=1 =
1^D[70]^D[69]^1^1^1^D[61]^D[60]^1^1^D[55]^D[52]^1^1^1^D[47]^D[45]^1^1^D[38]^D[36]^1^1^1^D[28]^1^1^1^1^D[22]^D[21]^D[20]^1^1^1^1^D[13]^1^D[5]^D[4];

CRC[2], A2=0=
1^1^1^1^1^1^1^D[58]^D[57]^1^D[50]^D[48]^1^1^1^D[43]^D[42]^1^1^D[34]^D[33]^1^1^D[25]^D[24]^1^D[17]^1^1^1^D[10]^D[8] ^1^D[2]^D[1]^D[0];

CRC[2], A2=1=
1^1^1^1^1^1^1^D[62]^D[61]^1^D[54]^D[52]^1^1^1^D[47]^D[46]^1^1^D[38]^D[37]^1^1^D[29]^D[28]^1^D[21]^1^1^1^D[14]^D12]^1^D[6]^D[5]^D[4];

CRC[3], A2=0 =
1^1^D[64]^1^1^1^D[59]^D[58]^1^D[51]^D[49]^D[48]^1^1^1^D[43]^D[40]^1^D[35]^D[34]^1^1^D[26]^D[25]^1^D[18]^D[16]^1^1^D[11]^D[9] ^1^D[3]^D[2]^D[1];

CRC[3], A2=1 =
1^1^D[68]^1^1^1^D[63]^D[62]^1^D[55]^D[53]^D[52]^1^1^1^D[47]^D[44]^1^D[39]^D[38]^1^1^D[30]^D[29]^1^D[22]^D[20]^1^1^D[15]^D[13]^1^D[7]^D[6]^D[5];

CRC[4], A2=0 =
1^1^D[65]^D[64]^1^1^1^D[59]^D[56]^1^D[50]^D[49]^D[48]^1^1^1^D[41]^1^1^D[35]^1^1^D[27]^D[26]^D[24]^D[19]^D[17]^1^1^1^D[10]^D[8] ^1^D[3]^D[2];

CRC[4], A2=1 =
1^1^D[69]^D[68]^1^1^1^D[63]^D[60]^1^D[54]^D[53]^D[52]^1^1^1^D[45]^1^1^D[39]^1^1^D[31]^D[30]^D[28]^D[23]^D[21]^1^1^1^D[14]^D[12]^1^D[7]^D[6];

CRC[5], A2=0 =
1^D[66]^D[65]^D[64]^1^1^1^D[57]^1^D[51]^D[50]^D[49]^1^1^1^D[42]^D[40]^1^1^D[32]^1^1^D[27]^D[25]^1^D[18]^D[16]^1^1^D[11]^D[9] ^1^1^D[3];

CRC[5], A2=1 =
1^D[70]^D[69]^D[68]^1^1^1^D[61]^1^D[55]^D[54]^D[53]^1^1^1^D[46]^D[44]^1^1^D[36]^1^1^D[31]^D[29]^1^D[22]^D[20]^1^1^D[15]^D[13]^1^1^D[7];

CRC[6], A2=0 =
D[67]^D[66]^D[65]^D[64]^1^1^D[58]^1^1^D[51]^D[50]^D[48]^1^1^D[43]^D[41]^1^1^D[33]^D[32]^1^1^D[26]^1^D[19]^D[17]^D[16]^1^1^D[10]^1^1^1;

CRC[6], A2=1 =
D[71]^D[70]^D[69]^D[68]^1^1^D[62]^1^1^D[55]^D[54]^D[52]^1^1^D[47]^D[45]^1^1^D[37]^D[36]^1^1^D[30]^1^D[23]^D[21]^D[20]^1^1^D[14]^1^1^1;

CRC[7], A2=0=
1^D[67]^D[66]^D[65]^1^1^D[59]^1^1^1^D[51]^D[49]^D[48]^1^1^D[42]^1^1^D[34]^D[33]^1^1^D[27]^1^1^D[18]^D[17]^1^1^D[11]^1^1^1;

CRC[7], A2=1 =
1^D[71]^D[70]^D[69]^1^1^D[63]^1^1^1^D[55]^D[53]^D[52]^1^1^D[46]^1^1^D[38]^D[37]^1^1^D[31]^1^1^D[22]^D[21]^1^1^D[15]^1^1^1;

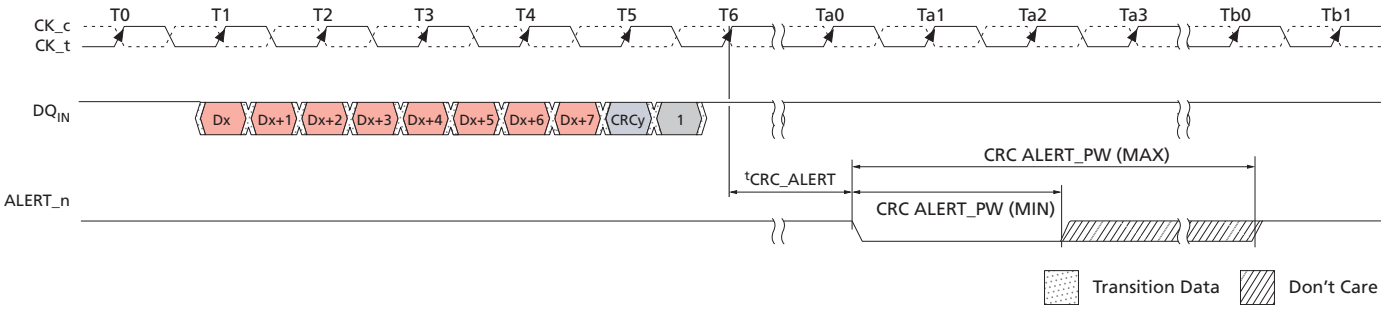
CRC Error Handling

The CRC error mechanism shares the same ALERT_n signal as CA parity for reporting write errors to the DRAM. The controller has two ways to distinguish between CRC errors and CA parity errors: 1) Read DRAM mode/MPR registers, and 2) Measure time ALERT_n is LOW. To speed up recovery for CRC errors, CRC errors are only sent back as a "short" pulse; the maximum pulse width is roughly ten clocks (unlike CA parity where ALERT_n is LOW longer than 45 clocks). The ALERT_n LOW could be longer than the maximum limit at the controller if there are multiple CRC errors as the ALERT_n signals are connected by a daisy chain bus. The latency to ALERT_n signal is defined as ^{t}CRC_ALERT in the following figure.

The DRAM will set the error status bit located at MR5[3] to a 1 upon detecting a CRC error, which will subsequently set the CRC error status flag in the MPR error log HIGH (MPR Page1, MPR3[7]). The CRC error status bit (and CRC error status flag) remains set at 1 until the DRAM controller clears the CRC error status bit using an MRS command to set MR5[3] to a 0. The DRAM controller, upon seeing an error as a pulse width, will retry the write transactions. The controller should consider the worst-case

delay for ALERT_n (during initialization) and backup the transactions accordingly. The DRAM controller may also be made more intelligent and correlate the write CRC error to a specific rank or a transaction.

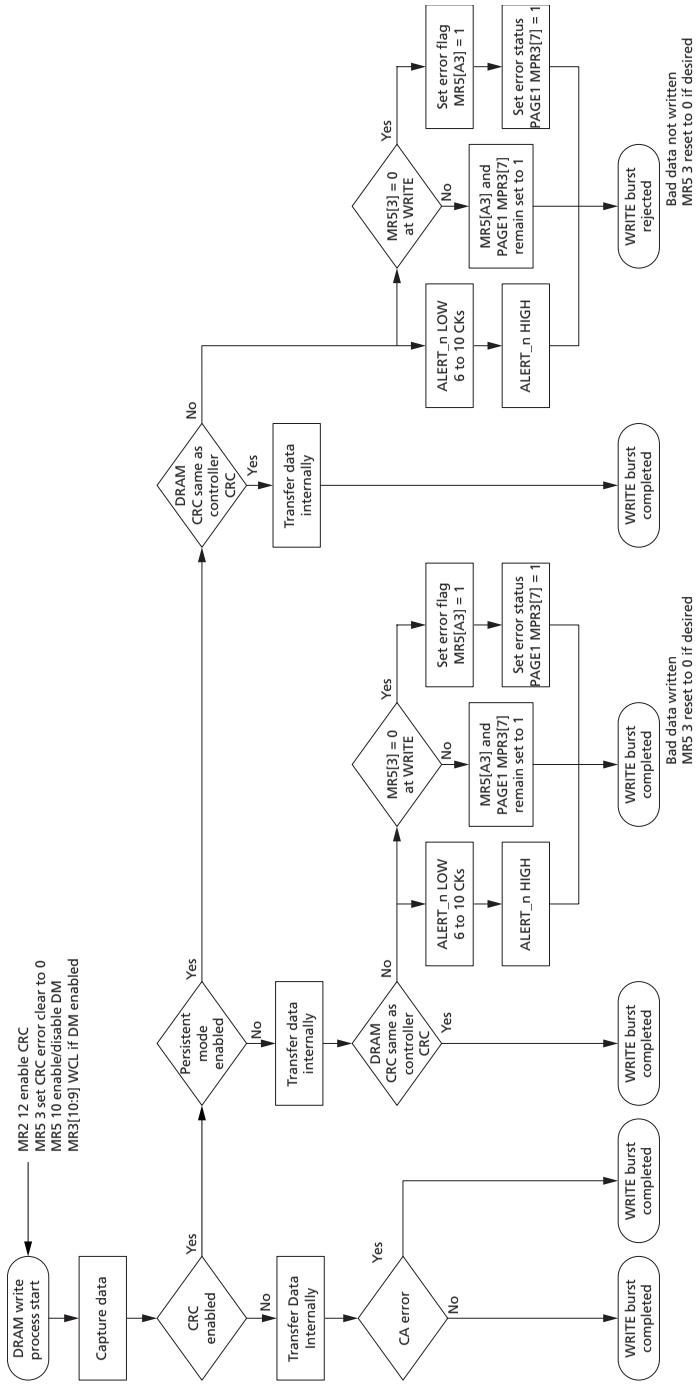
Figure 109: CRC Error Reporting



Notes: 1. D[71:1] CRC computed by DRAM did not match CRC[7:0] at T5 and started error generating process at T6.
2. CRC ALERT_PW is specified from the point where the DRAM starts to drive the signal LOW to the point where the DRAM driver releases and the controller starts to pull the signal up.
3. Timing diagram applies to x4, x8, and x16 devices.

CRC Write Data Flow Diagram

Figure 110: CA Parity Flow Diagram



Data Bus Inversion

The DATA BUS INVERSION (DBI) function is supported only for x8 and x16 configurations (it is not supported on x4 devices). DBI opportunistically inverts data bits, and in conjunction with the DBI_n I/O, less than half of the DQs will switch LOW for a given DQS strobe edge. The DBI function shares a common pin with the DATA MASK (DM) and TDQS functions. The DBI function applies to either or both READ and WRITE operations: Write DBI cannot be enabled at the same time the DM function is enabled, and DBI is not allowed during MPR READ operation. Valid configurations for TDQS, DM, and DBI functions are shown below.

Table 61: DBI vs. DM vs. TDQS Function Matrix

Read DBI	Write DBI	Data Mask (DM)	TDQS (x8 only)
Enabled (or Disabled) **MR5[12]=1** (or MR5[12] = 0)	Disabled MR5[11] = 0	Disabled MR5[10] = 0	Disabled MR1[11] = 0
	Enabled MR5[11] = 1	Disabled MR5[10] = 0	Disabled MR1[11] = 0
	Disabled MR5[11] = 0	**Enabled** MR5[10] = 1	Disabled MR1[11] = 0
Disabled MR5[12] = 0	Disabled MR5[11] = 0	Disabled MR5[10] = 0	**Enabled** MR1[11] = 1

DBI During a WRITE Operation

If DBI_n is sampled LOW on a given byte lane during a WRITE operation, the DRAM inverts write data received on the DQ inputs prior to writing the internal memory array. If DBI_n is sampled HIGH on a given byte lane, the DRAM leaves the data received on the DQ inputs noninverted. The write DQ frame format is shown below for x8 and x16 configurations (the x4 configuration does not support the DBI function).

Table 62: DBI Write, DQ Frame Format (x8)

	Transfer							
Function	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
DQ[7:0]	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
DM_n or DBI_n	DM0 or DBI0	DM1 or DBI1	DM2 or DBI2	DM3 or DBI3	DM4 or DBI4	DM5 or DBI5	DM6 or DBI6	DM7 or DBI7

Table 63: DBI Write, DQ Frame Format (x16)

	Transfer, Lower (L) and Upper(U)							
Function	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
DQ[7:0]	LByte 0	LByte 1	LByte 2	LByte 3	LByte 4	LByte 5	LByte 6	LByte 7
LDM_n or LDBI_n	LDM0 or LDBI0	LDM1 or LDBI1	LDM2 or LDBI2	LDM3 or LDBI3	LDM4 or LDBI4	LDM5 or LDBI5	LDM6 or LDBI6	LDM7 or LDBI7
DQ[15:8]	UByte 0	UByte 1	UByte 2	UByte 3	UByte 4	UByte 5	UByte 6	UByte 7
UDM_n or UDBI_n	UDM0 or UDBI0	UDM1 or UDBI1	UDM2 or UDBI2	UDM3 or UDBI3	UDM4 or UDBI4	UDM5 or UDBI5	UDM6 or UDBI6	UDM7 or UDBI7

DBI During a READ Operation

If the number of 0 data bits within a given byte lane is greater than four during a READ operation, the DRAM inverts read data on its DQ outputs and drives the DBI_n pin LOW; otherwise, the DRAM does not invert the read data and drives the DBI_n pin HIGH. The read DQ frame format is shown below for x8 and x16 configurations (the x4 configuration does not support the DBI function).

Table 64: DBI Read, DQ Frame Format (x8)

Function	Transfer Byte							
	0	1	2	3	4	5	6	7
DQ[7:0]	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
DBI_n	DBI0	DBI1	DBI2	DBI3	DBI4	DBI5	DBI6	DBI7

Table 65: DBI Read, DQ Frame Format (x16)

Function	Transfer Byte, Lower (L) and Upper(U)							
	0	1	2	3	4	5	6	7
DQ[7:0]	LByte 0	LByte 1	LByte 2	LByte 3	LByte 4	LByte 5	LByte 6	LByte 7
LDBI_n	LDBI0	LDBI1	LDBI2	LDBI3	LDBI4	LDBI5	LDBI6	LDBI7
DQ[15:8]	UByte 0	UByte 1	UByte 2	UByte 3	UByte 4	UByte 5	UByte 6	UByte 7
UDBI_n	UDBI0	UDBI1	UDBI2	UDBI3	UDBI4	UDBI5	UDBI6	UDBI7

Data Mask

The DATA MASK (DM) function, also described as PARTIAL WRITE, is supported only for x8 and x16 configurations (it is not supported on x4 devices). The DM function shares a common pin with the DBI_n and TDQS functions. The DM function applies only to WRITE operations and cannot be enabled at the same time the WRITE DBI function is enabled. The valid configurations for the TDQS, DM, and DBI functions are shown here.

Table 66: DM vs. TDQS vs. DBI Function Matrix

Data Mask (DM)	TDQS (x8 only)	Write DBI	Read DBI
Enabled MR5[10] = 1	Disabled MR1[11] = 0	Disabled MR5[11] = 0	**Enabled** or Disabled MR5[12] = 1 or MR5[12] = 0
Disabled MR5[10] = 0	**Enabled** MR1[11] = 1	Disabled MR5[11] = 0	Disabled MR5[12] = 0
	Disabled MR1[11] = 0	**Enabled** MR5[11] = 1	**Enabled** or Disabled MR5[12] = 1 or MR5[12] = 0
	Disabled MR1[11] = 0	Disabled MR5[11] = 0	**Enabled** (or Disabled) MR5[12] = 1 (or MR5[12] = 0)

When enabled, the DM function applies during a WRITE operation. If DM_n is sampled LOW on a given byte lane, the DRAM masks the write data received on the DQ inputs. If DM_n is sampled HIGH on a given byte lane, the DRAM does not mask the data and writes this data into the DRAM core. The DQ frame format for x8 and x16 configurations is shown below. If both CRC write and DM are enabled (via MRS), the CRC will be checked and valid prior to the DRAM writing data into the DRAM core. If a CRC error occurs while the DM feature is enabled, CRC write persistent mode will be enabled and data will not be written into the DRAM core. In the case of CRC write enabled and DM disabled (via MRS), that is, CRC write nonpersistent mode, data is written to the DRAM core even if a CRC error occurs.

Table 67: Data Mask, DQ Frame Format (x8)

Function	Transfer							
	0	1	2	3	4	5	6	7
DQ[7:0]	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
DM_n or DBI_n	DM0 or DBI0	DM1 or DBI1	DM2 or DBI2	DM3 or DBI3	DM4 or DBI4	DM5 or DBI5	DM6 or DBI6	DM7 or DBI7

Table 68: Data Mask, DQ Frame Format (x16)

Function	Transfer, Lower (L) and Upper (U)							
	0	1	2	3	4	5	6	7
DQ[7:0]	LByte 0	LByte 1	LByte 2	LByte 3	LByte 4	LByte 5	LByte 6	LByte 7
LDM_n or LDBI_n	LDM0 or LDBI0	LDM1 or LDBI1	LDM2 or LDBI2	LDM3 or LDBI3	LDM4 or LDBI4	LDM5 or LDBI5	LDM6 or LDBI6	LDM7 or LDBI7
DQ[15:8]	UByte 0	UByte 1	UByte 2	UByte 3	UByte 4	UByte 5	UByte 6	UByte 7
UDM_n or UDBI_n	UDM0 or UDBI0	UDM1 or UDBI1	UDM2 or UDBI2	UDM3 or UDBI3	UDM4 or UDBI4	UDM5 or UDBI5	UDM6 or UDBI6	UDM7 or UDBI7

Programmable Preamble Modes and DQS Postambles

The device supports programmable WRITE and READ preamble modes, either the normal 1^{t}CK preamble mode or special 2^{t}CK preamble mode. The 2^{t}CK preamble mode places special timing constraints on many operational features as well as being supported for data rates of DDR4-2400 and faster. The WRITE preamble 1^{t}CK or 2^{t}CK mode can be selected independently from READ preamble 1^{t}CK or 2^{t}CK mode.

READ preamble training is also supported; this mode can be used by the DRAM controller to train or "read level" the DQS receivers.

There are ^{t}CCD restrictions under some circumstances:

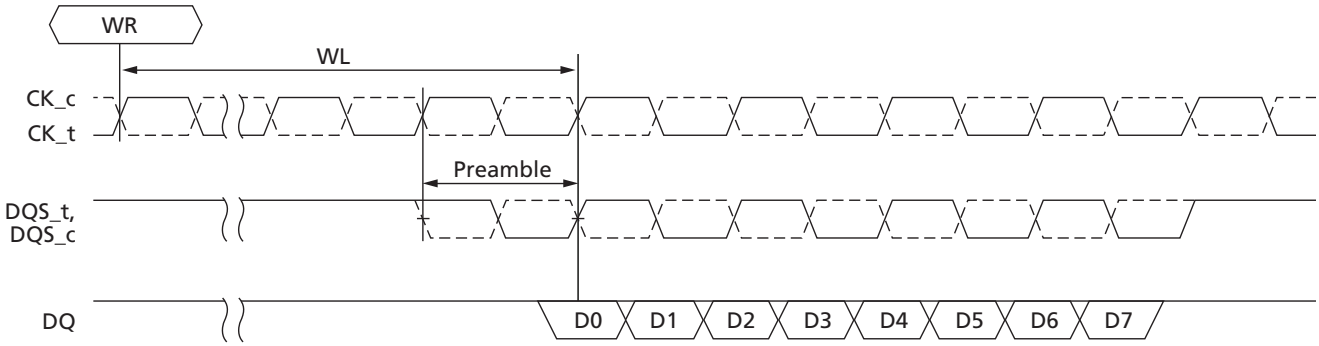
- When 2^{t}CK READ preamble mode is enabled, a ^{t}CCD_S or ^{t}CCD_L of 5 clocks is not allowed.
- When 2^{t}CK WRITE preamble mode is enabled and write CRC is *not* enabled, a ^{t}CCD_S or ^{t}CCD_L of 5 clocks is not allowed.
- When 2^{t}CK WRITE preamble mode is enabled and write CRC is enabled, a ^{t}CCD_S or ^{t}CCD_L of 6 clocks is not allowed.

WRITE Preamble Mode

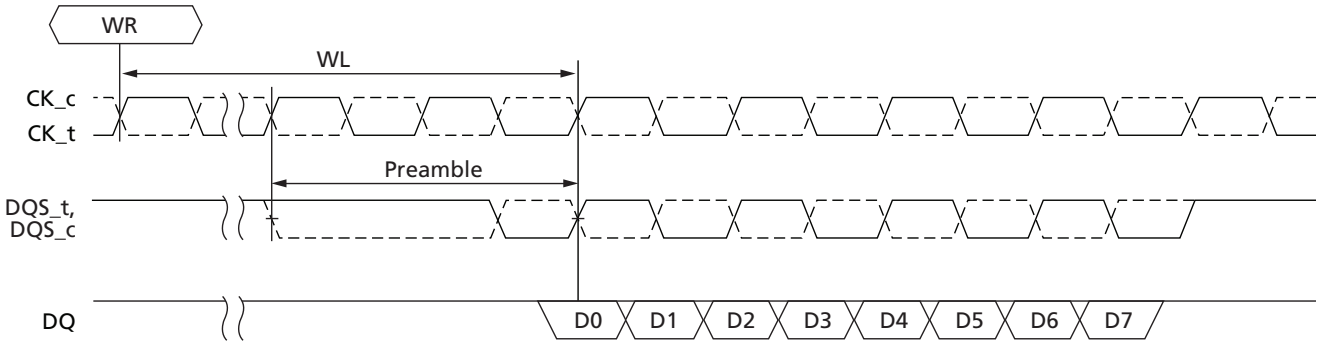
MR4[12] = 0 selects 1^{t}CK WRITE preamble mode while MR4[12] = 1 selects 2^{t}CK WRITE preamble mode. Examples are shown in the figures below.

Figure 111: 1^{t}CK vs. 2^{t}CK WRITE Preamble Mode

1^{t}CK Mode



2^{t}CK Mode



CWL has special considerations when in the 2tCK WRITE preamble mode. The CWL value selected in MR2[5:3], as seen in table below, requires at least one additional clock when the primary CWL value and 2tCK WRITE preamble mode are used; no additional clocks are required when the alternate CWL value and 2tCK WRITE preamble mode are used.

Table 69: CWL Selection

	CWL - Primary Choice		CWL - Alternate Choice	
Speed Bin	**1tCK Preamble**	**2tCK Preamble**	**1tCK Preamble**	**2tCK Preamble**
DDR4-1600	9	N/A	11	N/A
DDR4-1866	10	N/A	12	N/A
DDR4-2133	11	N/A	14	N/A
DDR4-2400	12	14	16	16
DDR4-2666	14	16	18	18
DDR4-2933	16	18	20	20
DDR4-3200	16	18	20	20

Note: 1. CWL programmable requirement for MR2[5:3].

When operating in 2tCK WRITE preamble mode, tWTR (command based) and tWR (MR0[11:9]) must be programmed to a value 1 clock greater than the tWTR and tWR setting normally required for the applicable speed bin to be JEDEC compliant; however, Micron's DDR4 DRAMs do not require these additional tWTR and tWR clocks. The CAS_n-to-CAS_n command delay to either a different bank group (tCCD_S) or the same bank group (tCCD_L) have minimum timing requirements that must be satisfied between WRITE commands and are stated in the Timing Parameters by Speed Bin tables.

Figure 112: 1tCK vs. 2tCK WRITE Preamble Mode, tCCD = 4

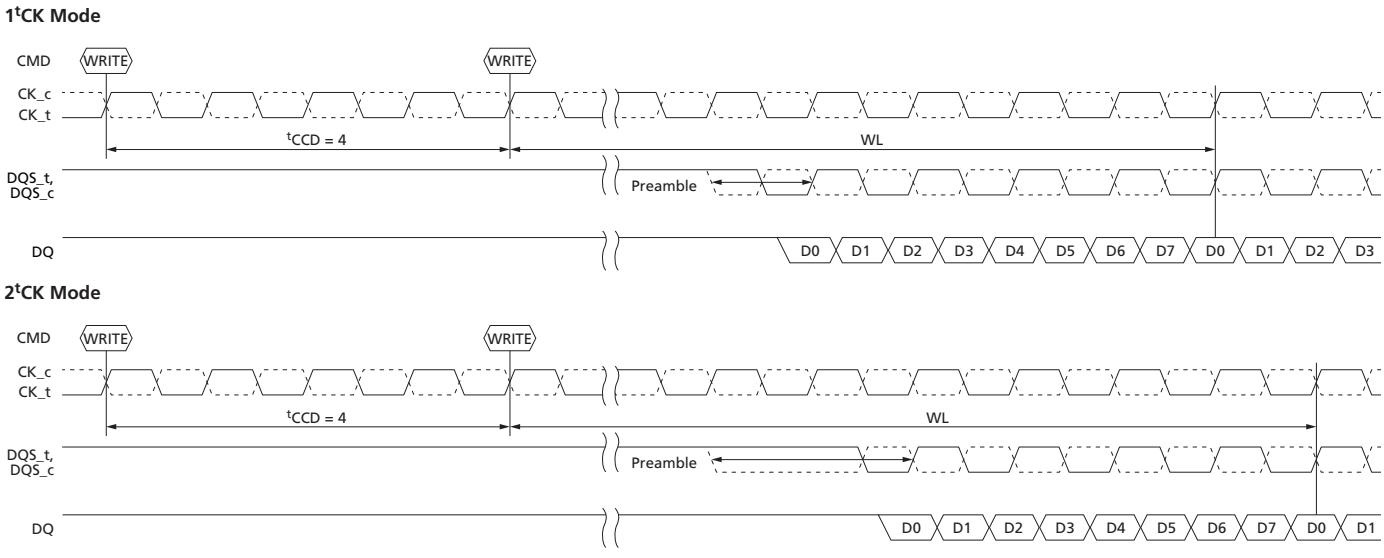
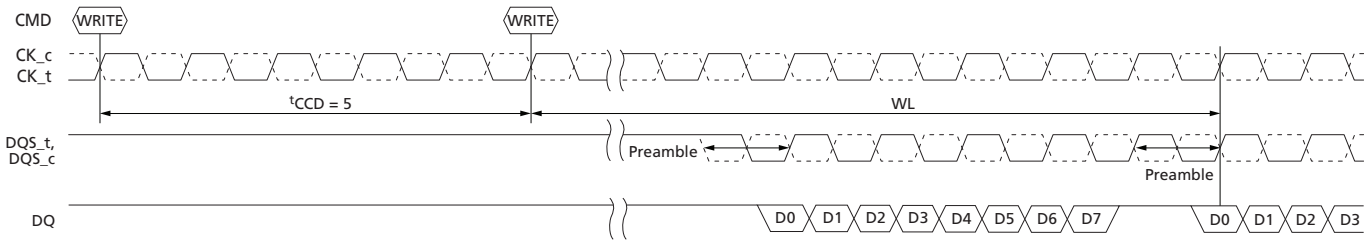


Figure 113: 1tCK vs. 2tCK WRITE Preamble Mode, tCCD = 5

1tCK Mode

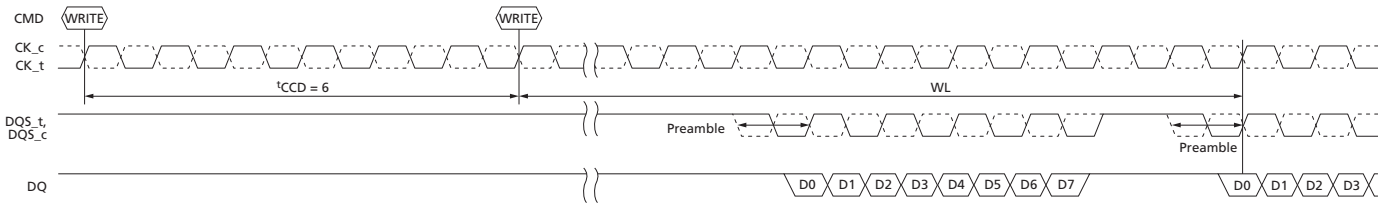


2tCK Mode: tCCD = 5 is not allowed in 2tCK mode.

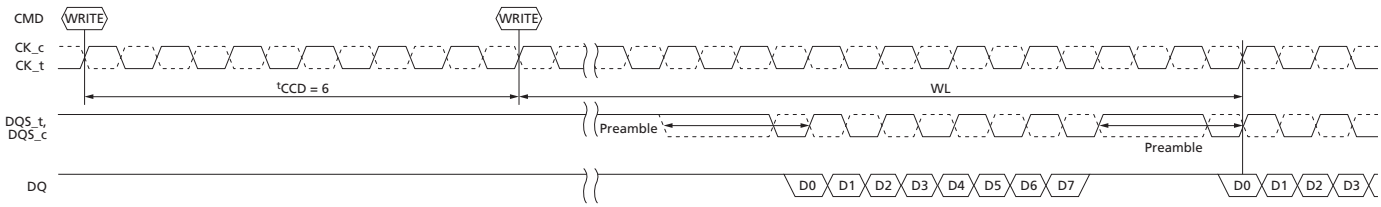
Note: 1. tCCD_S and tCCD_L = 5 tCKs is not allowed when in 2tCK WRITE preamble mode.

Figure 114: 1tCK vs. 2tCK WRITE Preamble Mode, tCCD = 6

1tCK Mode



2tCK Mode

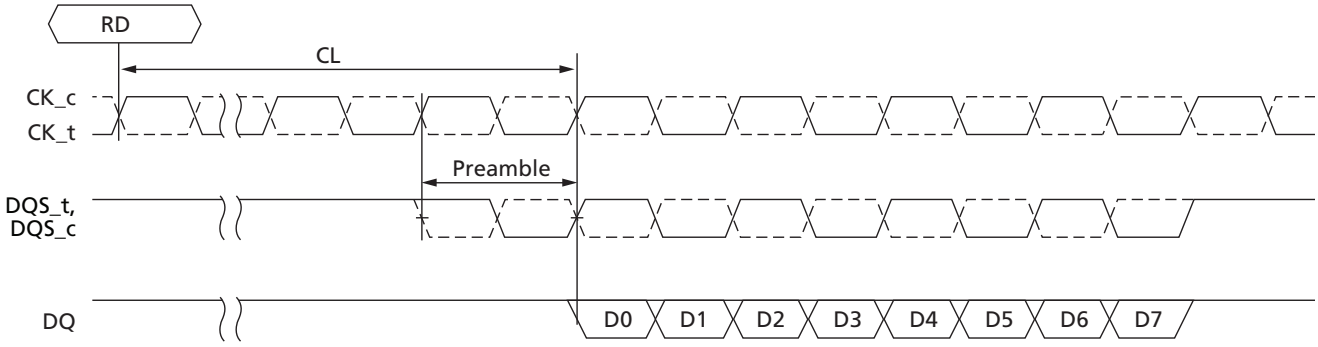


READ Preamble Mode

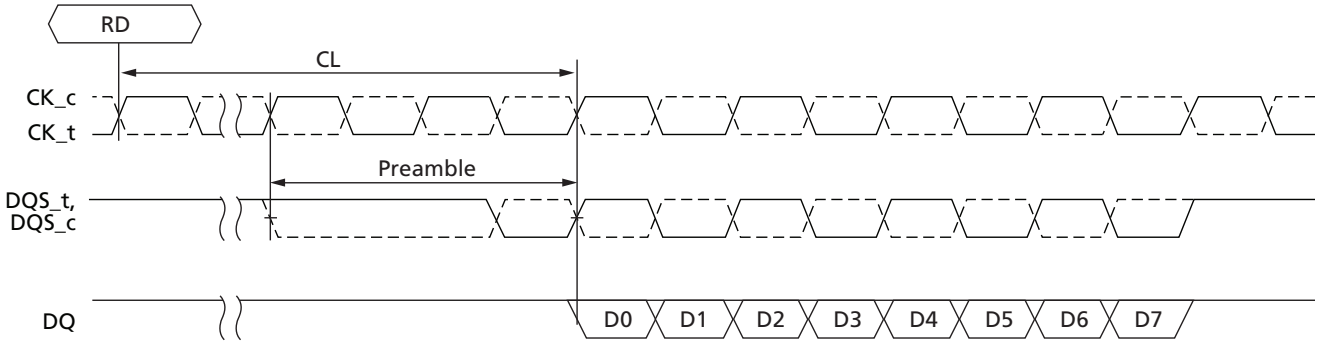
MR4[11] = 0 selects 1^{t}CK READ preamble mode and MR4[11] = 1 selects 2^{t}CK READ preamble mode. Examples are shown in the following figure.

Figure 115: 1^{t}CK vs. 2^{t}CK READ Preamble Mode

1^{t}CK Mode



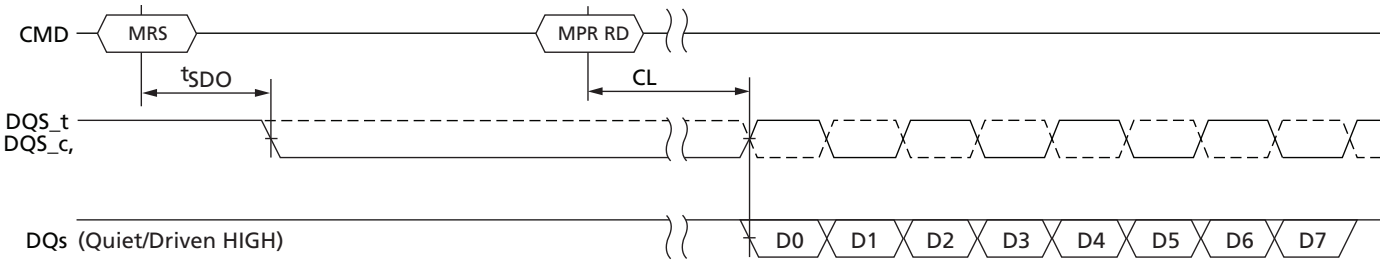
2^{t}CK Mode



READ Preamble Training

DDR4 supports READ preamble training via MPR reads; that is, READ preamble training is allowed only when the DRAM is in the MPR access mode. The READ preamble training mode can be used by the DRAM controller to train or "read level" its DQS receivers. READ preamble training is entered via an MRS command (MR4[10] = 1 is enabled and MR4[10] = 0 is disabled). After the MRS command is issued to enable READ preamble training, the DRAM DQS signals are driven to a valid level by the time ^{t}SDO is satisfied. During this time, the data bus DQ signals are held quiet, that is, driven HIGH. The DQS_t signal remains driven LOW and the DQS_c signal remains driven HIGH until an MPR Page0 READ command is issued (MPR0 through MPR3 determine which pattern is used), and when CAS latency (CL) has expired, the DQS signals will toggle normally depending on the burst length setting. To exit READ preamble training mode, an MRS command must be issued, MR4[10] = 0.

Figure 116: READ Preamble Training

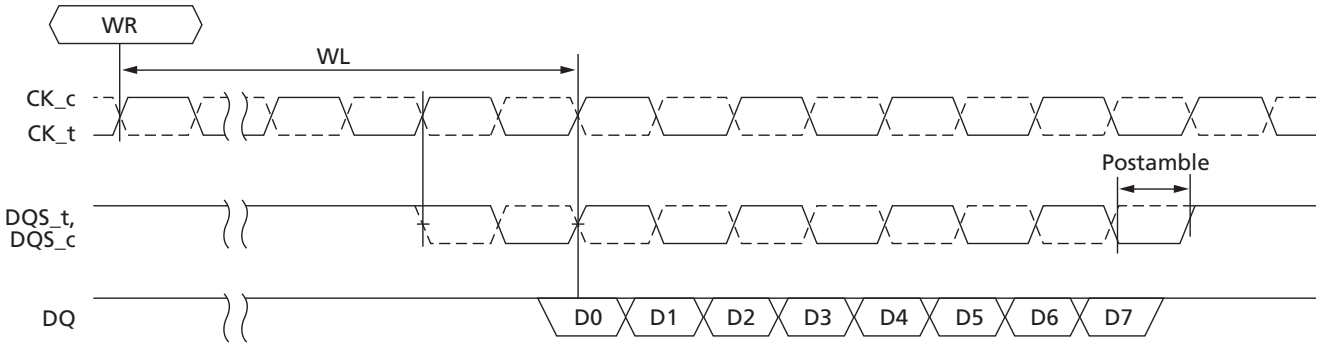


WRITE Postamble

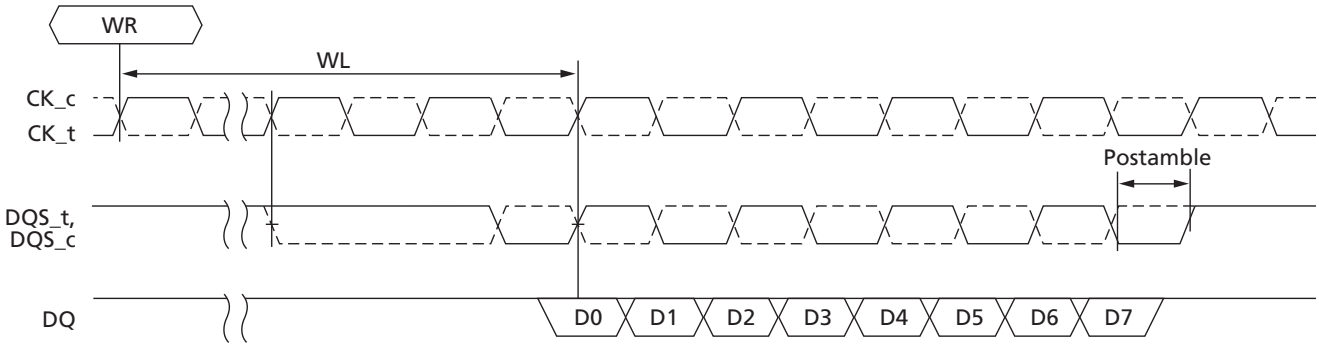
Whether the 1tCK or 2tCK WRITE preamble mode is selected, the WRITE postamble remains the same at ½tCK.

Figure 117: WRITE Postamble

1tCK Mode



2tCK Mode

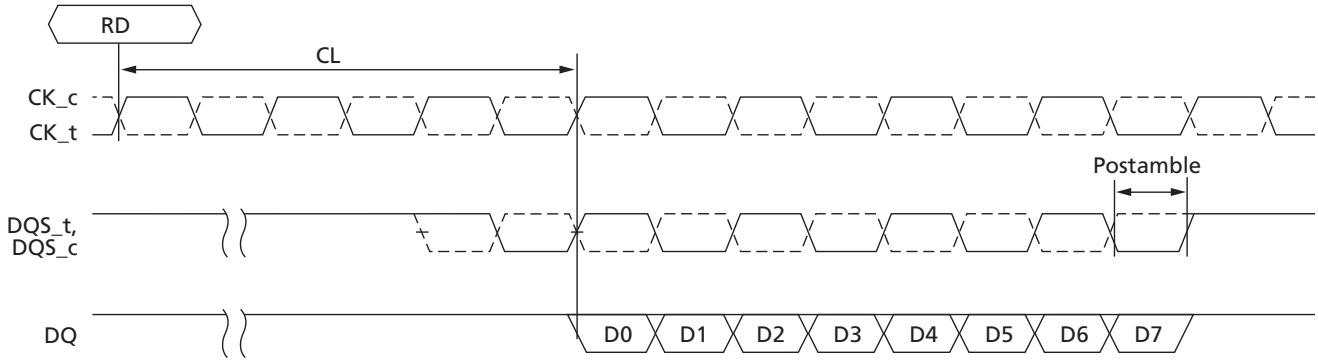


READ Postamble

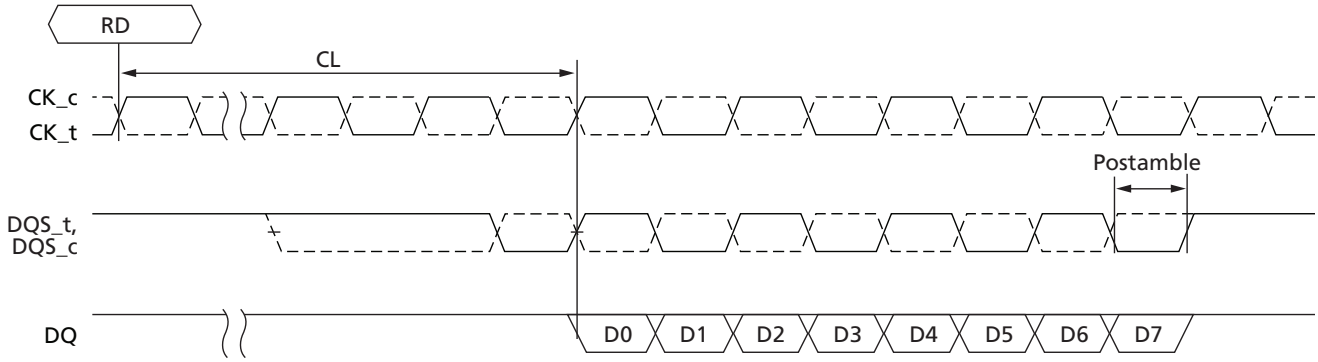
Whether the 1tCK or 2tCK READ preamble mode is selected, the READ postamble remains the same at ½tCK.

Figure 118: READ Postamble

1tCK Mode



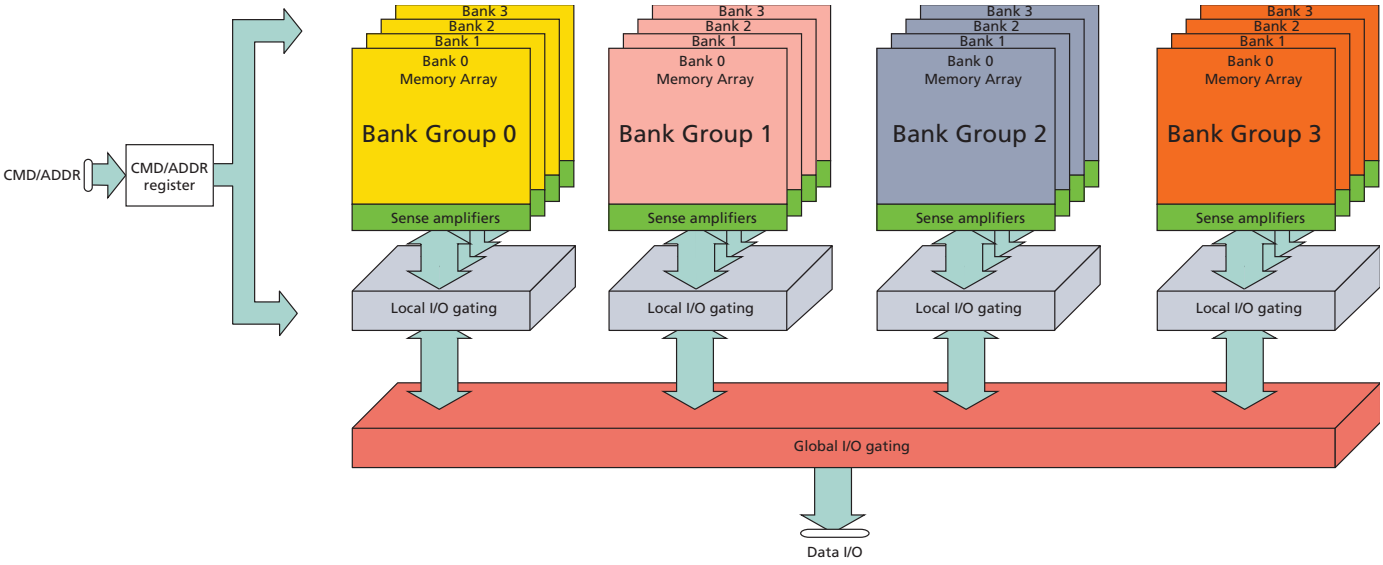
2tCK Mode



Bank Access Operation

DDR4 supports bank grouping: x4/x8 DRAMs have four bank groups (BG[1:0]), and each bank group is comprised of four subbanks (BA[1:0]); x16 DRAMs have two bank groups (BG[0]), and each bank group is comprised of four subbanks. Bank accesses to different banks' groups require less time delay between accesses than bank accesses to within the same bank's group. Bank accesses to different bank groups require ^{t}CCD_S (or short) delay between commands while bank accesses within the same bank group require ^{t}CCD_L (or long) delay between commands.

Figure 119: Bank Group x4/x8 Block Diagram



Notes: 1. Bank accesses to different bank groups require ^{t}CCD_S.
2. Bank accesses within the same bank group require ^{t}CCD_L.

Splitting the banks into bank groups with subbanks improved some bank access timings and increased others. However, considering DDR4 did not increase the prefetch from 8*n* to 16*n*, the penalty for staying 8*n* prefetch was significantly mitigated by using bank groups. The table below summarizes the timings affected (values listed as x*n*CK or yns means the larger of the two values).

Table 70: DDR4 Bank Group Timing Examples

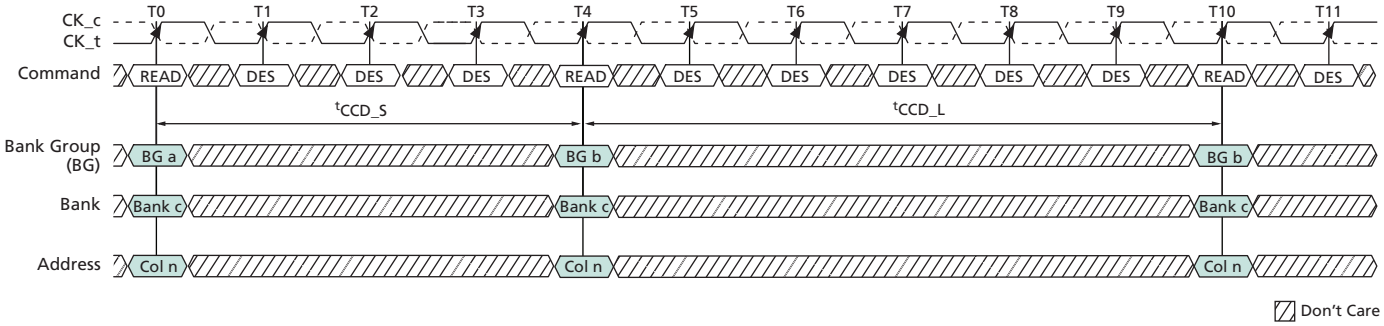
Parameter	DDR4-1600	DDR4-2133	DDR4-2400
^{t}CCD_S	4*n*CK	4*n*CK	4*n*CK
^{t}CCD_L	4*n*CK or 6.25ns	4*n*CK or 5.355ns	4*n*CK or 5ns
^{t}RRD_S (½K)	4*n*CK or 5ns	4*n*CK or 3.7ns	4*n*CK or 3.3ns
^{t}RRD_L (½K)	4*n*CK or 6ns	4*n*CK or 5.3ns	4*n*CK or 4.9ns
^{t}RRD_S (1K)	4*n*CK or 5ns	4*n*CK or 3.7ns	4*n*CK or 3.3ns
^{t}RRD_L (1K)	4*n*CK or 6ns	4*n*CK or 5.3ns	4*n*CK or 4.9ns

Table 70: DDR4 Bank Group Timing Examples (Continued)

Parameter	DDR4-1600	DDR4-2133	DDR4-2400
^{t}RRD_S (2K)	4*n*CK or 6ns	4*n*CK or 5.3ns	4*n*CK or 5.3ns
^{t}RRD_L (2K)	4*n*CK or 7.5ns	4*n*CK or 6.4ns	4*n*CK or 6.4ns
^{t}WTR_S	2*n*CK or 2.5ns	2*n*CK or 2.5ns	2*n*CK or 2.5ns
^{t}WTR_L	4*n*CK or 7.5ns	4*n*CK or 7.5ns	4*n*CK or 7.5ns

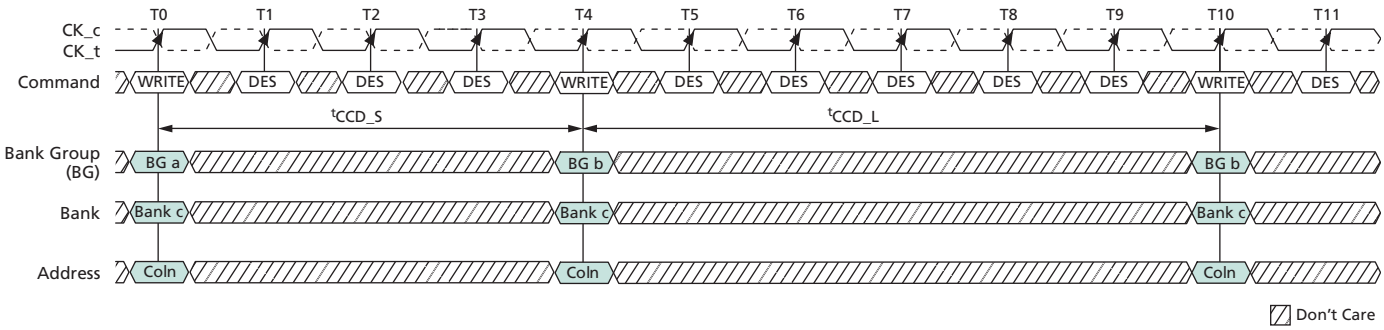
Notes: 1. Refer to Timing Tables for actual specification values, these values are shown for reference only and are not verified for accuracy.
2. Timings with both *n*CK and ns require both to be satisfied; that is, the larger time of the two cases must be satisfied.

Figure 120: READ Burst ^{t}CCD_S and ^{t}CCD_L Examples



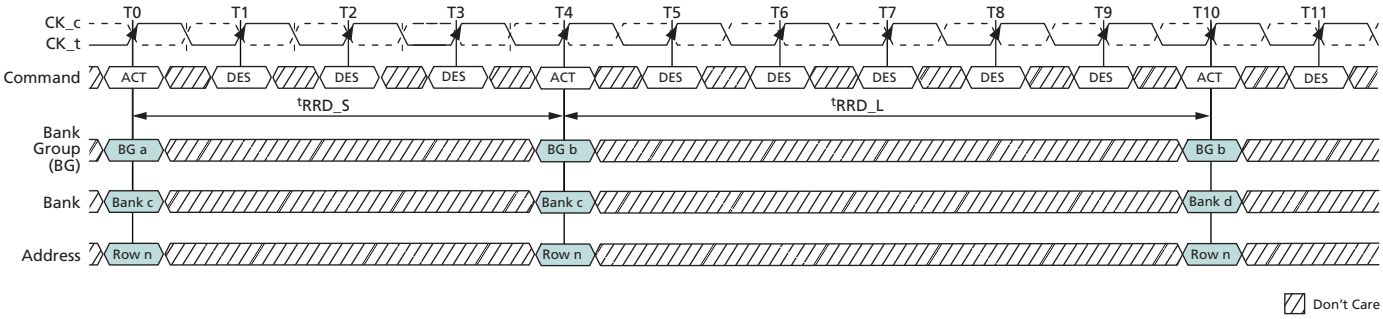
Notes: 1. ^{t}CCD_S; CAS_n-to-CAS_n delay (short). Applies to consecutive CAS_n to different bank groups (T0 to T4).
2. ^{t}CCD_L; CAS_n-to-CAS_n delay (long). Applies to consecutive CAS_n to the same bank group (T4 to T10).

Figure 121: Write Burst ^{t}CCD_S and ^{t}CCD_L Examples



Notes: 1. ^{t}CCD_S; CAS_n-to-CAS_n delay (short). Applies to consecutive CAS_n to different bank groups (T0 to T4).
2. ^{t}CCD_L; CAS_n-to-CAS_n delay (long). Applies to consecutive CAS_n to the same bank group (T4 to T10).

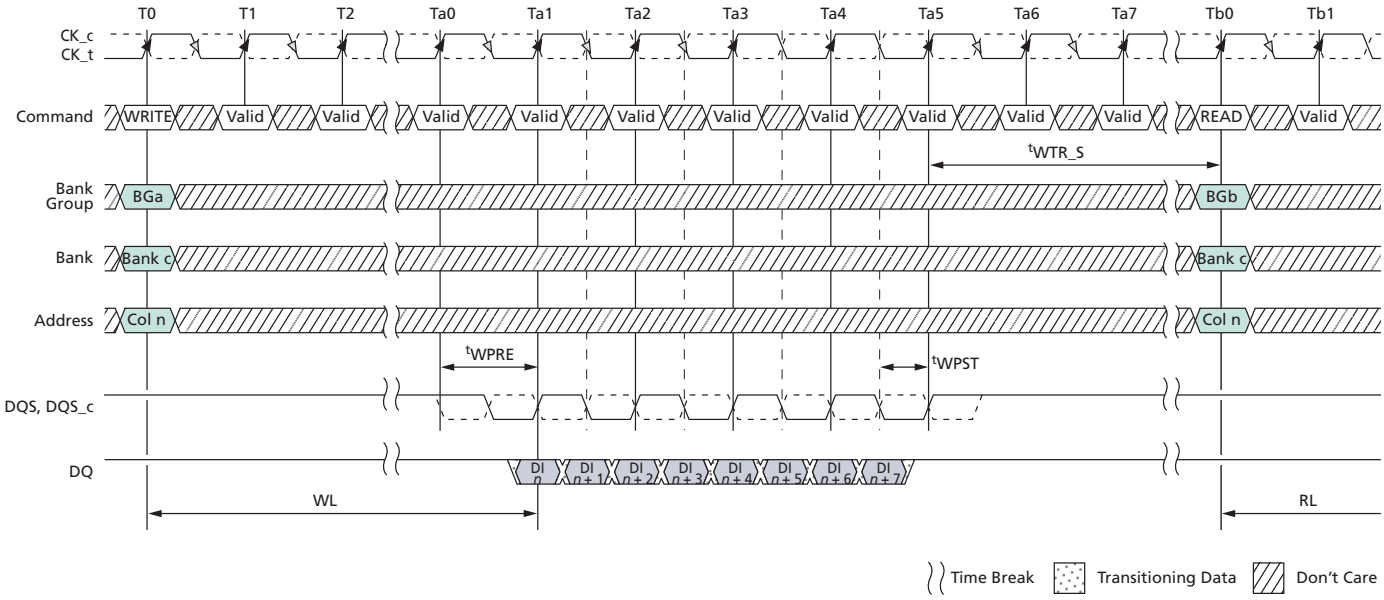
Figure 122: tRRD Timing



Notes: 1. tRRD_S; ACTIVATE-to-ACTIVATE command period (short); applies to consecutive ACTIVATE commands to different bank groups (T0 and T4).

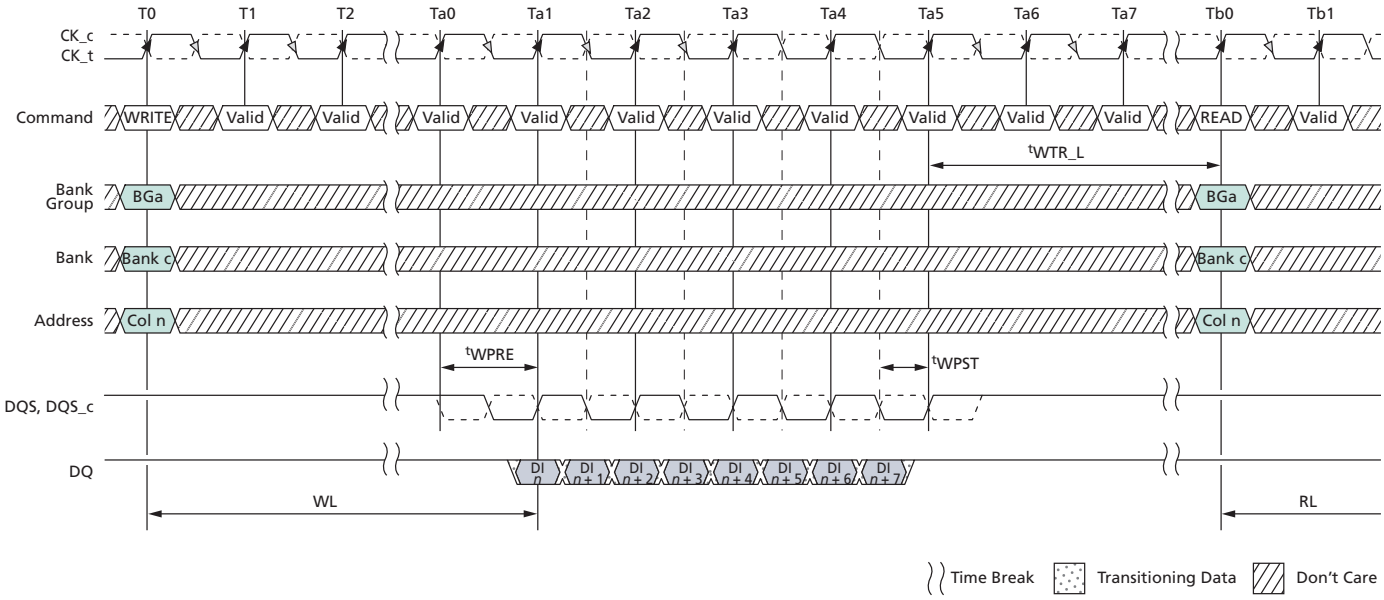
2. tRRD_L; ACTIVATE-to-ACTIVATE command period (long); applies to consecutive ACTIVATE commands to the different banks in the same bank group (T4 and T10).

Figure 123: ^{t}WTR_S Timing (WRITE-to-READ, Different Bank Group, CRC and DM Disabled)



Note: 1. ^{t}WTR_S: delay from start of internal write transaction to internal READ command to a different bank group.

Figure 124: ^{t}WTR_L Timing (WRITE-to-READ, Same Bank Group, CRC and DM Disabled)



Note: 1. ^{t}WTR_L: delay from start of internal write transaction to internal READ command to the same bank group.

READ Operation

Read Timing Definitions

The read timings shown below are applicable in normal operation mode, that is, when the DLL is enabled and locked.

Note: ^{t}DQSQ = both rising/falling edges of DQS; no ^{t}AC defined.

Rising data strobe edge parameters:

- ^{t}DQSCK (MIN)/(MAX) describes the allowed range for a rising data strobe edge relative to CK.
- ^{t}DQSCK is the actual position of a rising strobe edge relative to CK.
- ^{t}QSH describes the DQS differential output HIGH time.
- ^{t}DQSQ describes the latest valid transition of the associated DQ pins.
- ^{t}QH describes the earliest invalid transition of the associated DQ pins.

Falling data strobe edge parameters:

- ^{t}QSL describes the DQS differential output LOW time.
- ^{t}DQSQ describes the latest valid transition of the associated DQ pins.
- ^{t}QH describes the earliest invalid transition of the associated DQ pins.

Figure 125: Read Timing Definition

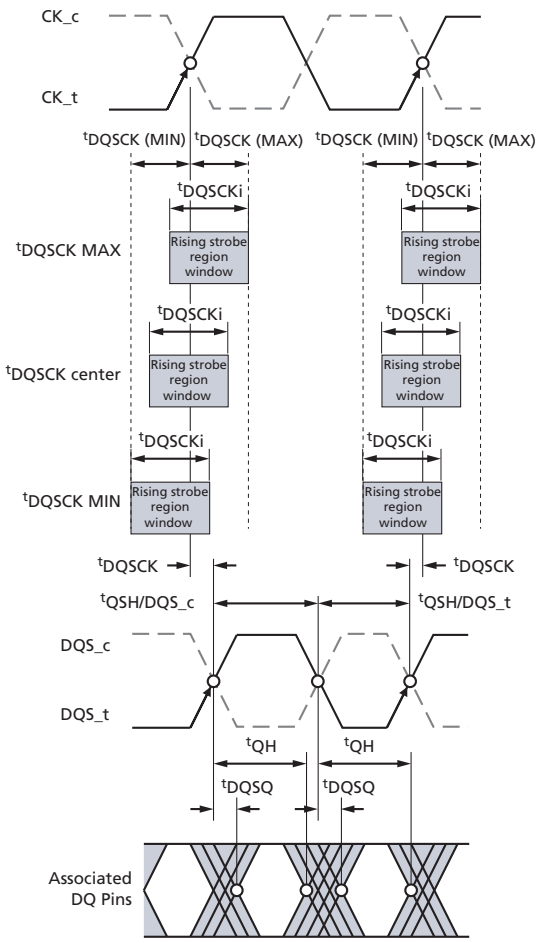


Table 71: Read-to-Write and Write-to-Read Command Intervals

Access Type	Bank Group	Timing Parameters	Note
Read-to-Write, minimum	Same	CL - CWL + RBL/2 + 1^{t}CK + ^{t}WPRE	1, 2
	Different	CL - CWL + RBL/2 + 1^{t}CK + ^{t}WPRE	1, 2
Write-to-Read, minimum	Same	CWL + WBL/2 + ^{t}WTR_L	1, 3
	Different	CWL + WBL/2 + ^{t}WTR_S	1, 3

Notes: 1. These timings require extended calibrations times ^{t}ZQinit and ^{t}ZQCS.
2. RBL: READ burst length associated with READ command, RBL = 8 for fixed 8 and on-the-fly mode 8 and RBL = 4 for fixed BC4 and on-the-fly mode BC4.
3. WBL: WRITE burst length associated with WRITE command, WBL = 8 for fixed 8 and on-the-fly mode 8 or BC4 and WBL = 4 for fixed BC4 only.

Read Timing – Clock-to-Data Strobe Relationship

The clock-to-data strobe relationship shown below is applicable in normal operation mode, that is, when the DLL is enabled and locked.

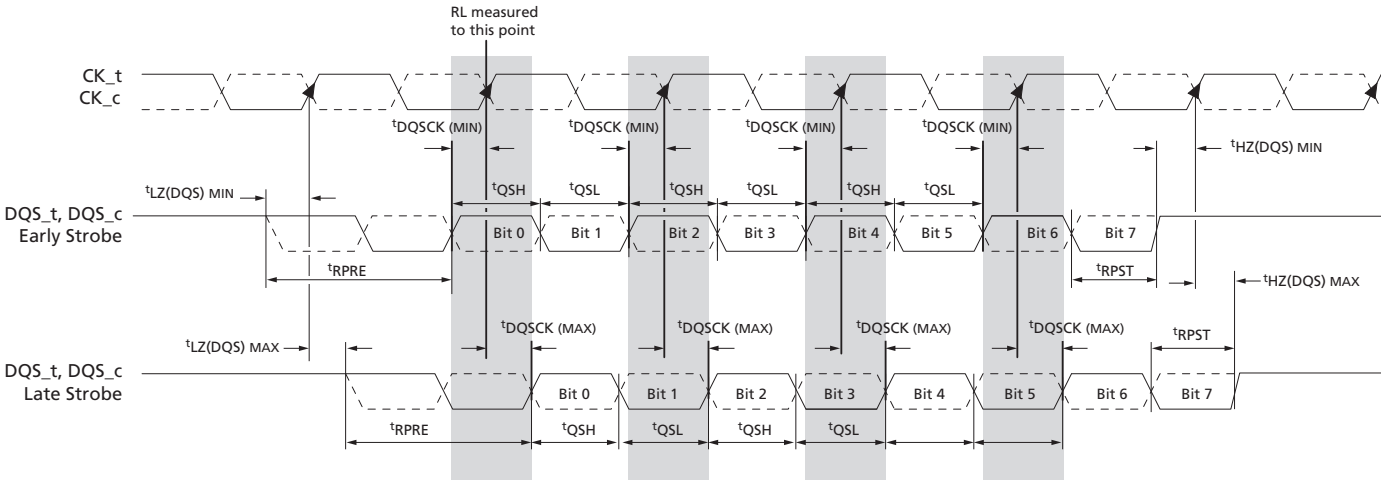
Rising data strobe edge parameters:

- ^{t}DQSCK (MIN)/(MAX) describes the allowed range for a rising data strobe edge relative to CK.
- ^{t}DQSCK is the actual position of a rising strobe edge relative to CK.
- ^{t}QSH describes the data strobe high pulse width.
- ^{t}HZ(DQS) DQS strobe going to high, nondrive level (shown in the postamble section of the figure below).

Falling data strobe edge parameters:

- ^{t}QSL describes the data strobe low pulse width.
- ^{t}LZ(DQS) DQS strobe going to low, initial drive level (shown in the preamble section of the figure below).

Figure 126: Clock-to-Data Strobe Relationship



Notes: 1. Within a burst, the rising strobe edge will vary within ^{t}DQSCKi while at the same voltage and temperature. However, when the device, voltage, and temperature variations are incorporated, the rising strobe edge variance window can shift between ^{t}DQSCK (MIN) and ^{t}DQSCK (MAX).
A timing of this window's right edge (latest) from rising CK_t, CK_c is limited by a device's actual ^{t}DQSCK (MAX).
A timing of this window's left inside edge (earliest) from rising CK_t, CK_c is limited by ^{t}DQSCK (MIN).

2. Notwithstanding Note 1, a rising strobe edge with ^{t}DQSCK (MAX) at T(n) can not be immediately followed by a rising strobe edge with ^{t}DQSCK (MIN) at T(n + 1) because other timing relationships (^{t}QSH, ^{t}QSL) exist: if ^{t}DQSCK(n + 1) < 0: ^{t}DQSCK(n) < 1.0 ^{t}CK - (^{t}QSH (MIN) + ^{t}QSL (MIN)) - |^{t}DQSCK(n + 1) |.
3. The DQS_t, DQS_c differential output HIGH time is defined by ^{t}QSH, and the DQS_t, DQS_c differential output LOW time is defined by ^{t}QSL.
4. ^{t}LZ(DQS) MIN and ^{t}HZ(DQS) MIN are not tied to ^{t}DQSCK (MIN) (early strobe case), and ^{t}LZ(DQS) MAX and ^{t}HZ(DQS) MAX are not tied to ^{t}DQSCK (MAX) (late strobe case).
5. The minimum pulse width of READ preamble is defined by ^{t}RPRE (MIN).
6. The maximum READ postamble is bound by ^{t}DQSCK (MIN) plus ^{t}QSH (MIN) on the left side and ^{t}HZDSQ (MAX) on the right side.
7. The minimum pulse width of READ postamble is defined by ^{t}RPST (MIN).
8. The maximum READ preamble is bound by ^{t}LZDQS (MIN) on the left side and ^{t}DQSCK (MAX) on the right side.

Read Timing – Data Strobe-to-Data Relationship

The data strobe-to-data relationship is shown below and is applied when the DLL is enabled and locked.

Note: ^{t}DQSQ: both rising/falling edges of DQS; no ^{t}AC defined.

Rising data strobe edge parameters:

- ^{t}DQSQ describes the latest valid transition of the associated DQ pins.
- ^{t}QH describes the earliest invalid transition of the associated DQ pins.

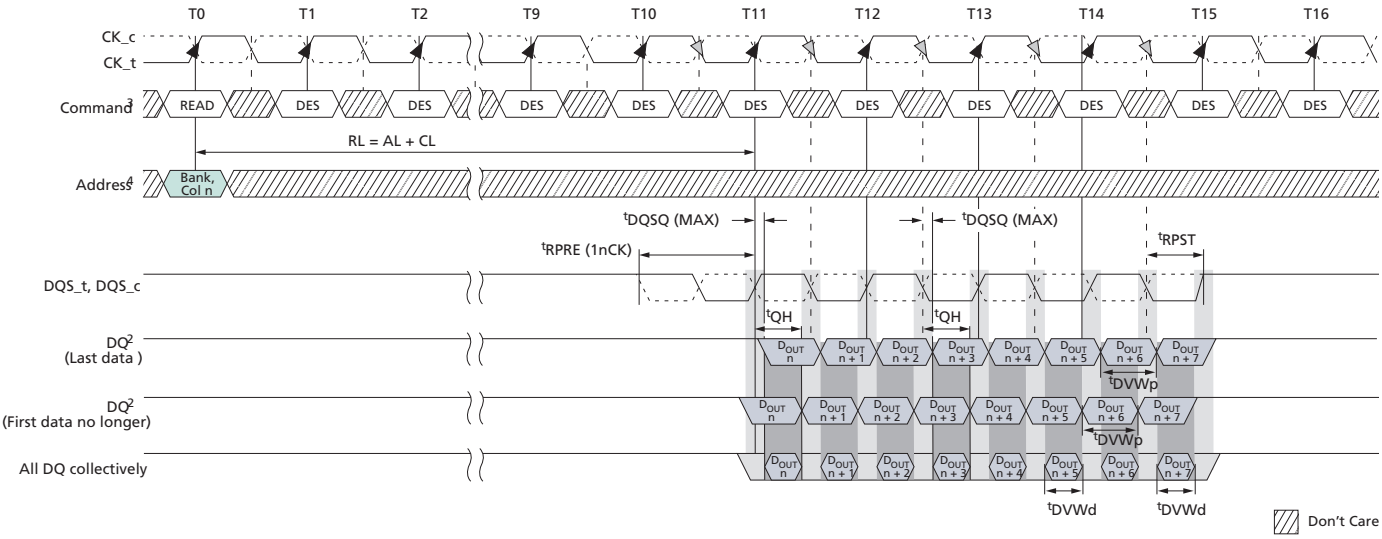
Falling data strobe edge parameters:

- ^{t}DQSQ describes the latest valid transition of the associated DQ pins.
- ^{t}QH describes the earliest invalid transition of the associated DQ pins.

Data valid window parameters:

- ^{t}DVWd is the Data Valid Window per device per UI and is derived from [^{t}QH - ^{t}DQSQ] of each UI on a given DRAM
- ^{t}DVWp is the Data Valid Window per pin per UI and is derived [^{t}QH - ^{t}DQSQ] of each UI on a pin of a given DRAM

Figure 127: Data Strobe-to-Data Relationship

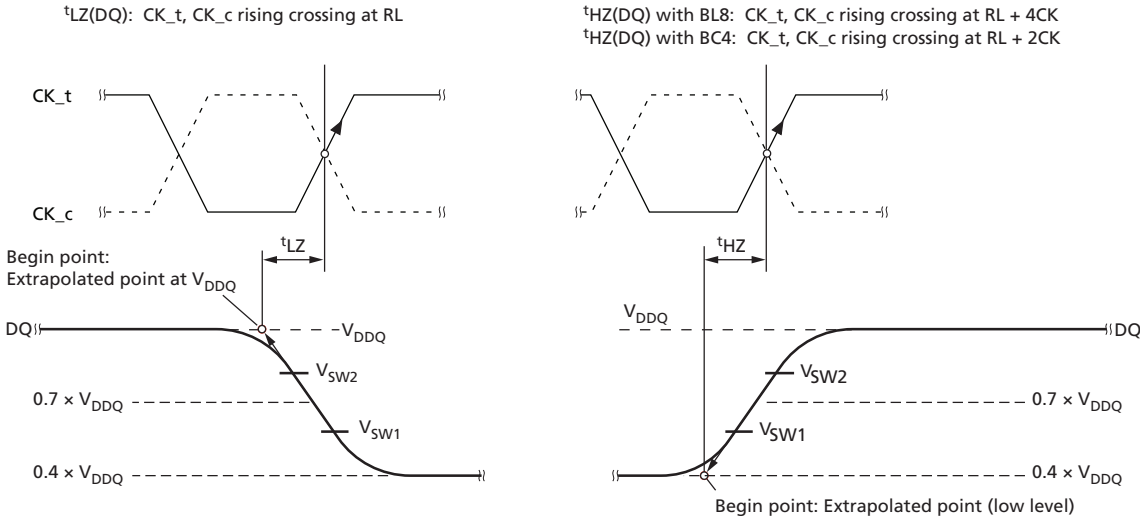


Notes: 1. BL = 8, RL = 11 (AL = 0, CL = 1), Premable = 1^{t}CK.
2. D_{OUT} *n* = data-out from column *n*.
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[A1:0 = 00] or MR0[A1:0 = 01] and A12 = 1 during READ commands at T0.
5. Output timings are referenced to V_{DDQ}, and DLL on for locking.
6. ^{t}DQSQ defines the skew between DQS to data and does not define DQS to clock.
7. Early data transitions may not always happen at the same DQ. Data transitions of a DQ can vary (either early or late) within a burst.

tLZ(DQS), tLZ(DQ), tHZ(DQS), and tHZ(DQ) Calculations

tHZ and tLZ transitions occur in the same time window as valid data transitions. These parameters are referenced to a specific voltage level that specifies when the device output is no longer driving tHZ(DQS) and tHZ(DQ), or begins driving tLZ(DQS) and tLZ(DQ). The figure below shows a method to calculate the point when the device is no longer driving tHZ(DQS) and tHZ(DQ), or begins driving tLZ(DQS) and tLZ(DQ), by measuring the signal at two different voltages. The actual voltage measurement points are not critical as long as the calculation is consistent. tLZ(DQS), tLZ(DQ), tHZ(DQS), and tHZ(DQ) are defined as singled-ended parameters.

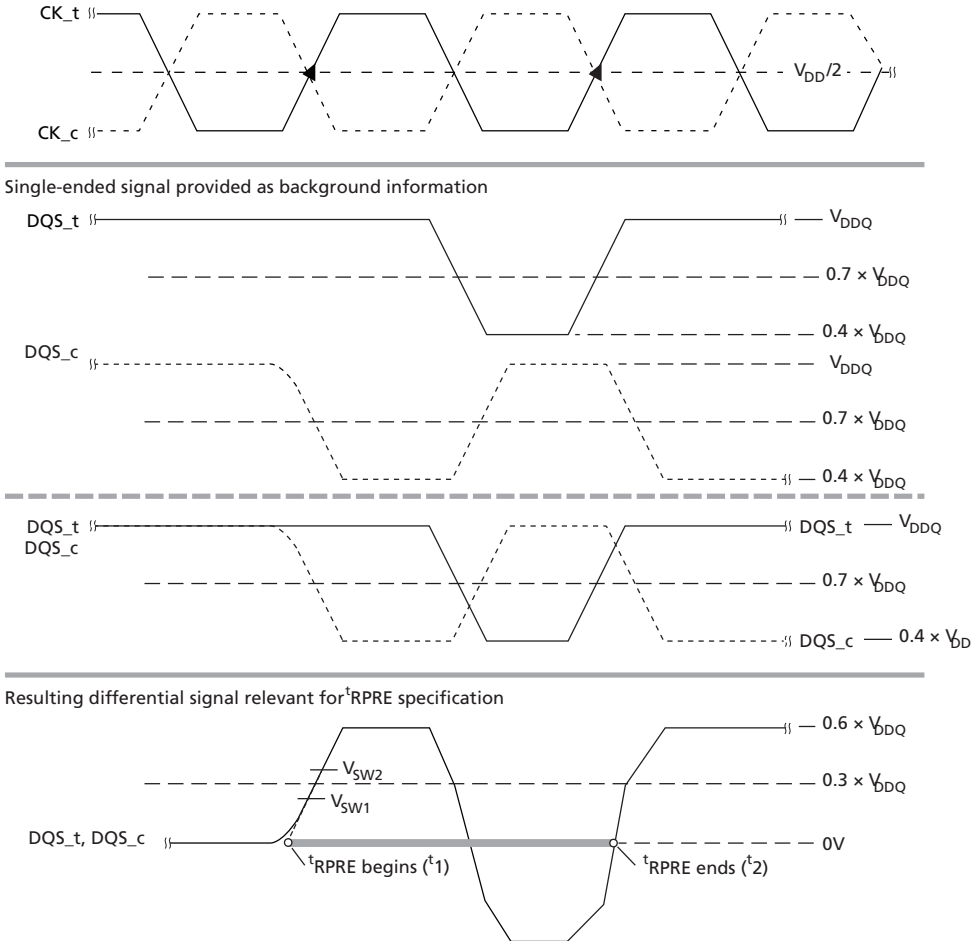
Figure 128: tLZ and tHZ Method for Calculating Transitions and Endpoints



Notes: 1. V_{SW1} = (0.70 - 0.04) × V_{DDQ} for both tLZ and tHZ.
2. V_{SW2} = (0.70 + 0.04) × V_{DDQ} for both tLZ and tHZ.
3. Extrapolated point (low level) = V_{DDQ}/(50 + 34) × 34 = 0.4 × V_{DDQ}
Driver impedance = RZQ/7 = 34Ω
V_{TT} test load = 50Ω to V_{DDQ}.

^{t}RPRE Calculation

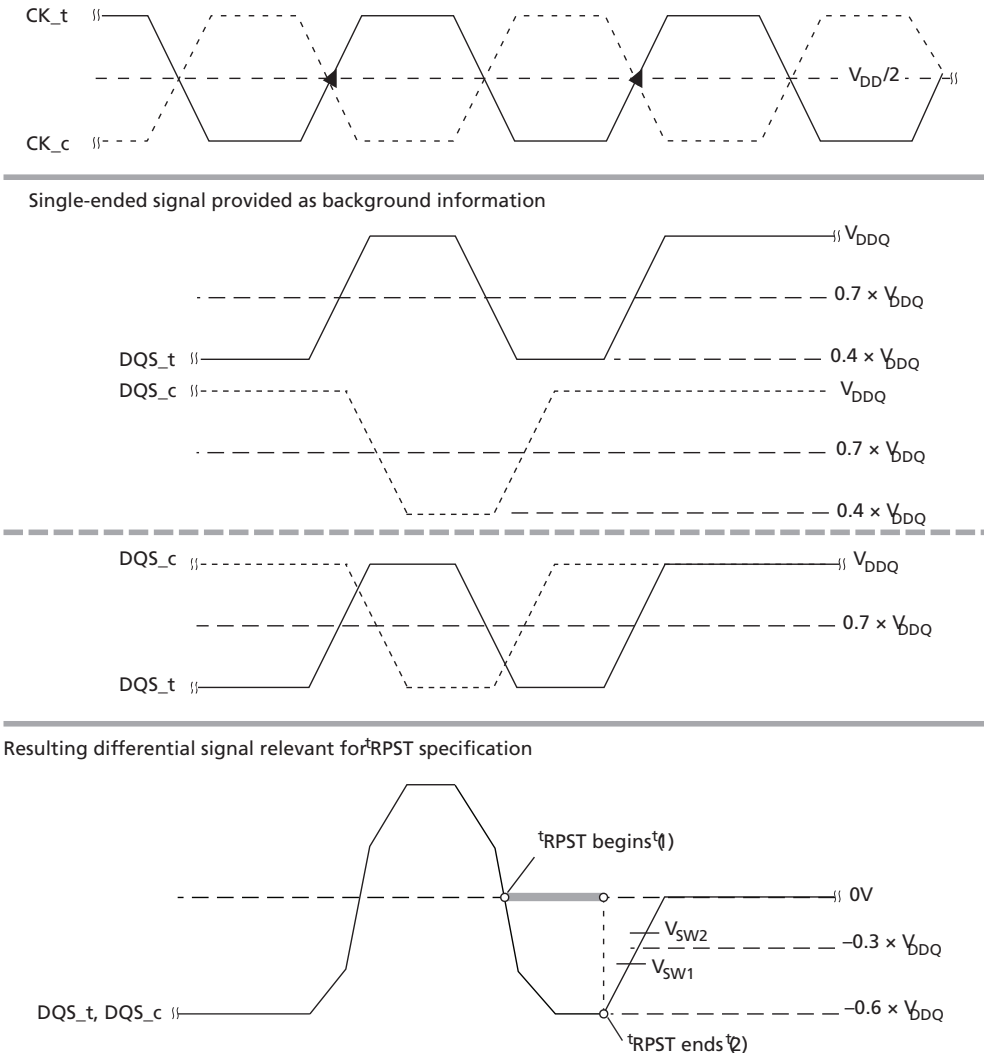
Figure 129: ^{t}RPRE Method for Calculating Transitions and Endpoints



Notes: 1. V_{sw1} = (0.3 - 0.04) × V_{DDQ}.
2. V_{sw2} = (0.30 + 0.04) × V_{DDQ}.
3. DQS_t and DQS_c low level = V_{DDQ}/(50 + 34) × 34 = 0.4 × V_{DDQ}
   Driver impedance = RZQ/7 = 34Ω
   V_{TT} test load = 50Ω to V_{DDQ}.

^{t}RPST Calculation

Figure 130: ^{t}RPST Method for Calculating Transitions and Endpoints



Notes: 1. V_{sw1} = (–0.3 - 0.04) × V_{DDQ}.
2. V_{sw2} = (–0.30 + 0.04) × V_{DDQ}.
3. DQS_t and DQS_c low level = V_{DDQ}/(50 + 34) × 34 = 0.4 × V_{DDQ}
Driver impedance = RZQ/7 = 34Ω
V_{TT} test load = 50Ω to V_{DDQ}.

READ Burst Operation

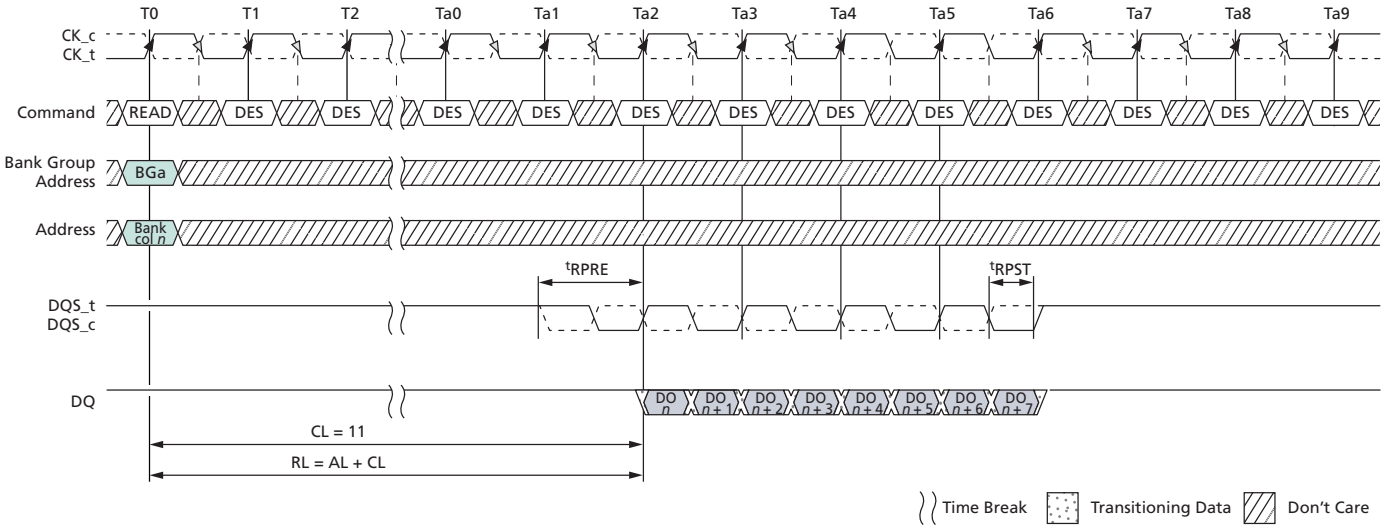
DDR4 READ commands support bursts of BL8 (fixed), BC4 (fixed), and BL8/BC4 on-the-fly (OTF); OTF uses address A12 to control OTF when OTF is enabled:

- A12 = 0, BC4 (BC4 = burst chop)
- A12 = 1, BL8

READ commands can issue precharge automatically with a READ with auto precharge command (RDA), and is enabled by A10 HIGH:

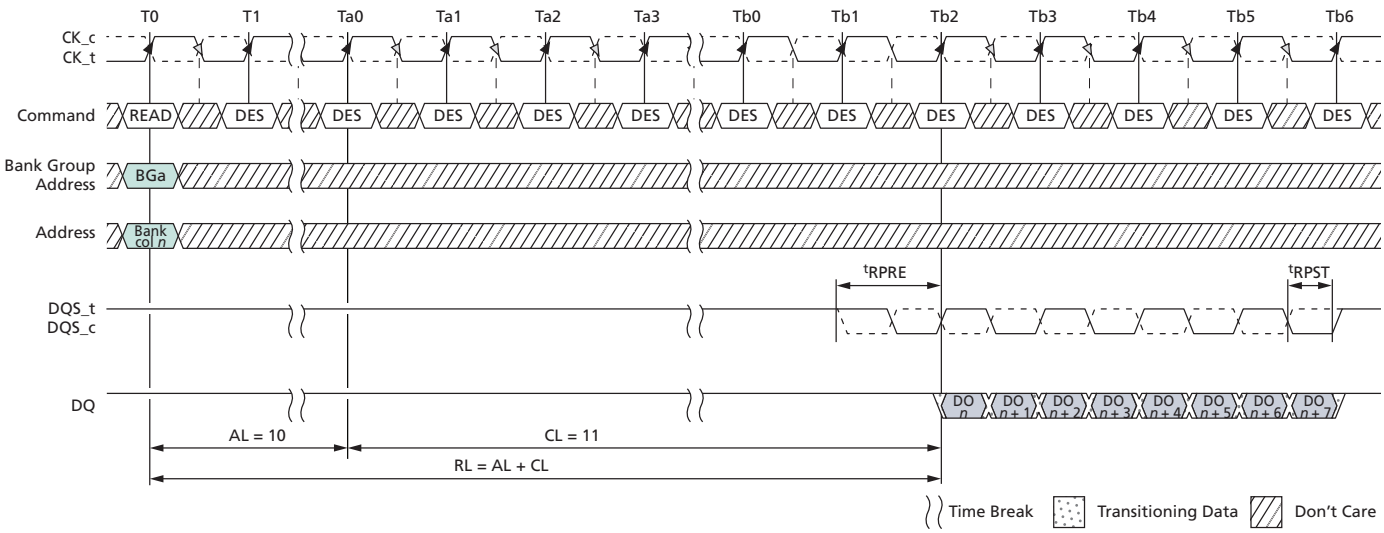
- READ command with A10 = 0 (RD) performs standard read, bank remains active after READ burst.
- READ command with A10 = 1 (RDA) performs read with auto precharge, bank goes in to precharge after READ burst.

Figure 131: READ Burst Operation, RL = 11 (AL = 0, CL = 11, BL8)



Notes: 1. BL8, RL = 0, AL = 0, CL = 11, Preamble = 1^{t}CK.
2. DO *n* = data-out from column *n*.
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[1:0] = 00 or MR0[1:0] = 01 and A12 = 1 during READ command at T0.
5. CA parity = Disable, CS to CA latency = Disable, Read DBI = Disable.

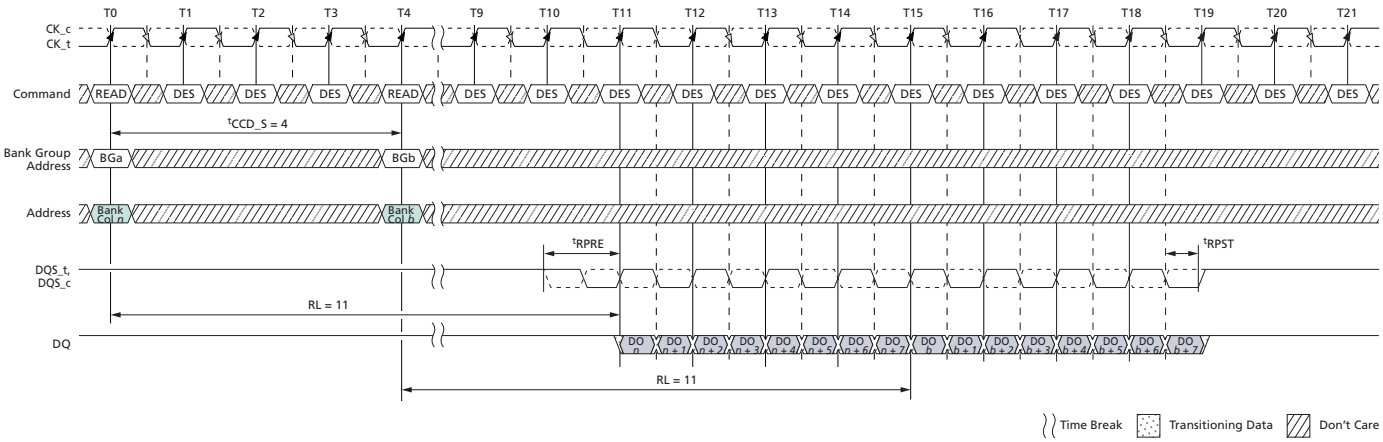
Figure 132: READ Burst Operation, RL = 21 (AL = 10, CL = 11, BL8)



Notes: 1. BL8, RL = 21, AL = (CL - 1), CL = 11, Preamble = 1^{t}CK.
2. DO *n* = data-out from column *n*.
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[1:0] = 00 or MR0[1:0] = 01 and A12 = 1 during READ command at T0.
5. CA parity = Disable, CS to CA latency = Disable, Read DBI = Disable.

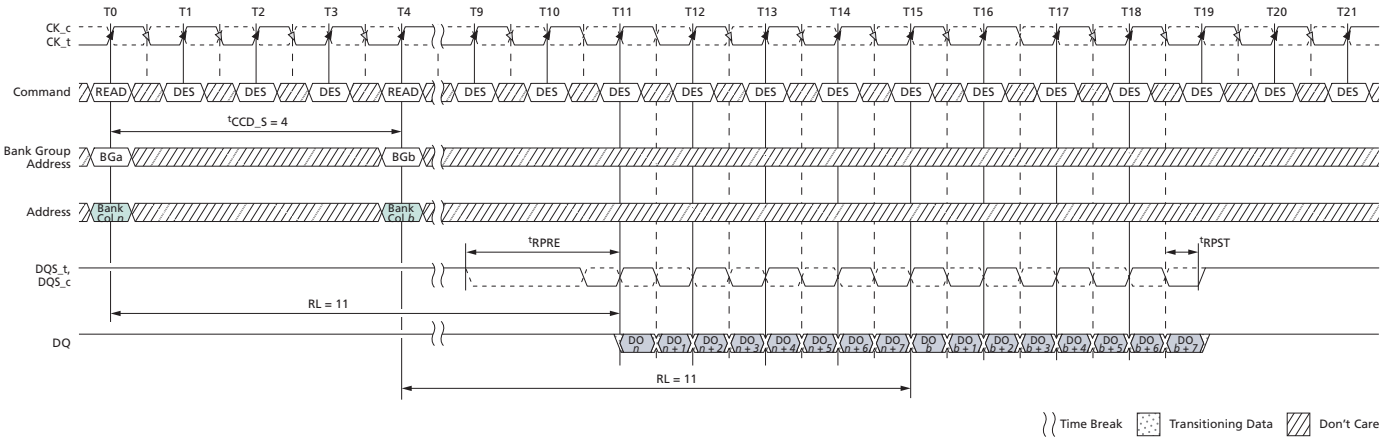
READ Operation Followed by Another READ Operation

Figure 133: Consecutive READ (BL8) with 1tCK Preamble in Different Bank Group



Notes: 1. BL8, AL = 0, CL = 11, Preamble = 1tCK.
2. DO *n* (or *b*) = data-out from column *n* (or column *b*).
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[1:0] = 00 or MR0[1:0] = 01 and A12 = 1 during READ commands at T0 and T4.
5. CA parity = Disable, CS to CA latency = Disable, Read DBI = Disable.

Figure 134: Consecutive READ (BL8) with 2tCK Preamble in Different Bank Group



Notes: 1. BL8, AL = 0, CL = 11, Preamble = 2tCK.
2. DO *n* (or *b*) = data-out from column *n* (or column *b*).
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[1:0] = 00 or MR0[1:0] = 01 and A12 = 1 during READ commands at T0 and T4.
5. CA parity = Disable, CS to CA latency = Disable, Read DBI = Disable.