

Table 4-65: Fault Injection Data (Word 3) Register

Bits	Name	Core Access	Reset Value	Description
31:0	FI_D3	W	0	Bit positions set to 1 toggle the corresponding Bits[127:96] of the next data word written to the memory. The register is automatically cleared after the fault has been injected.

FI_ECC

This register is used to inject errors in the generated ECC written to the memory and can be used to test the error correction and error signaling. The bits set in the register toggle the corresponding ECC bits of the next data written to memory. After the fault has been injected, the Fault Injection ECC register is cleared automatically.

The register is only implemented if C_ECC_TEST = ON or ECC_TEST_FL_XOR = ON and ECC = ON in a DDR3/DDR4 SDRAM design in the Vivado IP catalog.

Injecting faults should be performed in a critical region in software; that is, writing this register and the subsequent write to memory must not be interrupted.

[Table 4-66](#) describes the register bit usage when DQ_WIDTH = 72.

Table 4-66: Fault Injection ECC Register for 72-Bit External Memory Width

Bits	Name	Core Access	Reset Value	Description
7:0	FI_ECC	W	0	Bit positions set to 1 toggle the corresponding bit of the next ECC written to the memory. The register is automatically cleared after the fault has been injected.

[Table 4-67](#) describes the register bit usage when DQ_WIDTH = 144.

Table 4-67: Fault Injection ECC Register for 144-Bit External Memory Width

Bits	Name	Core Access	Reset Value	Description
15:0	FI_ECC	R	0	Bit positions set to 1 toggle the corresponding bit of the next ECC written to the memory. The register is automatically cleared after the fault has been injected.

PHY Only Interface

This section describes the FPGA logic interface signals and key parameters of the DDR3 and DDR4 PHY. The goal is to implement a “PHY Only” solution that connects your own custom Memory Controller directly to the DDR3/DDR4 SDRAM generated PHY, instead of interfacing to the user interface or AXI Interface of a DDR3/DDR4 SDRAM generated Memory Controller. The PHY interface takes DRAM commands, like Activate, Precharge, Refresh, etc. at its input ports and issues them directly to the DRAM bus.

The PHY does not take in “memory transactions” like the user and AXI interfaces, which translate transactions into one or more DRAM commands that meet DRAM protocol and timing requirements. The PHY interface does no DRAM protocol or timing checking. When using a PHY Only option, you are responsible for meeting all DRAM protocol requirements and timing specifications of all DRAM components in the system.

The PHY runs at the system clock frequency, or 1/4 of the DRAM clock frequency. The PHY therefore accepts four DRAM commands per system clock and issues them serially on consecutive DRAM clock cycles on the DRAM bus. In other words, the PHY interface has four command slots: slots 0, 1, 2, and 3, which it accepts each system clock. The command in slot position 0 is issued on the DRAM bus first, and the command in slot 3 is issued last. The PHY does have limitations as to which slots can accept read and write CAS commands. For more information, see [CAS Command Timing Limitations, page 181](#). Except for CAS commands, each slot can accept arbitrary DRAM commands.

The PHY FPGA logic interface has an input port for each pin on a DDR3 or DDR4 bus. Each PHY command/address input port has a width that is eight times wider than its corresponding DRAM bus pin. For example, a DDR4 bus has one `act_n` pin, and the PHY has an 8-bit `mc_ACT_n` input port. Each pair of bits in the `mc_ACT_n` port corresponds to a “command slot.” The two LSBs are slot0 and the two MSBs are slot3. The PHY address input port for a DDR4 design with 18 address pins is 144 bits wide, with each byte corresponding to the four command slots for one DDR4 address pin. There are two bits for each command slot in each input port of the PHY.

This is due to the underlying design of the PHY and its support for double data rate data buses. But as the DRAM command/address bus is single data rate, you must always drive the two bits that correspond to a command slot to the same value. See the following interface tables for additional descriptions and examples in the timing diagrams that show how bytes and bits correspond to DRAM pins and command slots.

The PHY interface has read and write data ports with eight bits for each DRAM DQ pin. Each port bit represents one data bit on the DDR DRAM bus for a BL8 burst. Therefore one BL8 data burst for the entire DQ bus is transferred across the PHY interface on each system clock. The PHY only supports BL8 data transfers. The data format is the same as the user interface data format. For more information, see [PHY, page 33](#).

The PHY interface also has several control signals that you must drive and/or respond to when a read or write CAS command is issued. The control signals are used by the PHY to manage the transfer of read and write data between the PHY interface and the DRAM bus. See the following signal tables and timing diagrams.

Your custom Memory Controller must wait until the PHY output `calDone` is asserted before sending any DRAM commands to the PHY. The PHY initializes and trains the DRAM before asserting `calDone`. For more information on the PHY internal structures and training algorithms, see the [PHY, page 33](#). After `calDone` is asserted, the PHY is ready to accept any DRAM commands.

The only required DRAM or PHY commands are related to VT tracking and DRAM refresh/ZQ. These requirements are detailed in [VT Tracking, page 183](#) and [Refresh and ZQ, page 184](#).

PHY Interface Signals

The PHY interface signals to the FPGA logic can be categorized into six groups:

- [Clocking and Reset](#)
- [Command and Address](#)
- [Write Data](#)
- [Read Data](#)
- [PHY Control](#)
- [Debug](#)

Clocking and Reset and Debug signals are described in other sections or documents. See the corresponding references. In this section, a description is given for each signal in the remaining four groups and timing diagrams show examples of the signals in use.

Clocking and Reset

For more information on the clocking and reset, see the [Clocking, page 85](#) section.

Command and Address

[Table 4-68](#) shows the command and address signals for a PHY only option.

Table 4-68: Command and Address

Signal	Direction	Description
mc_ACT_n[7:0]	Input	DRAM ACT_n command signal for four DRAM clock cycles. Bits[1:0] correspond to the first DRAM clock cycle, Bits[3:2] to the second, Bits[5:4] to the third, and Bits[8:7] to the fourth. For center alignment to the DRAM clock with 1N timing, both bits of a given bit pair should be asserted to the same value. See timing diagrams for examples. All of the command/address ports in this table follow the same eight bits per DRAM pin format. Active-Low. This signal is not used in DDR3 systems.
mc_ADR[ADDR_WIDTH × 8 – 1:0]	Input	DRAM address. Eight bits in the PHY interface for each address bit on the DRAM bus. Bits[7:0] corresponds to DRAM address Bit[0] on four DRAM clock cycles. Bits[15:8] corresponds to DRAM address Bit[1] on four DRAM clock cycles, etc. See the timing diagrams for examples. All of the multi-bit DRAM signals in this table follow the same format of 1-byte of the PHY interface port corresponding to four commands for one DRAM pin. Mixed active-Low and High depending on which type of DRAM command is being issued, but follows the DRAM pin active-High/Low behavior. The function of each byte of the mc_ADR port depends on whether the memory type is DDR4 or DDR3 and the particular DRAM command that is being issued. These functions match the DRAM address pin functions. For example, with DDR4 memory and the mc_ACT_n port bits asserted High, mc_ADR[135:112] have the function of RAS_n, CAS_n, and WE_n pins.
mc_RAS_n[7:0]	Input	DDR3 DRAM RAS_n pin. Not used in DDR4 systems.
mc_CAS_n[7:0]	Input	DDR3 DRAM CAS_n pin. Not used in DDR4 systems.
mc_WE_n[7:0]	Input	DDR3 DRAM WE_n pin. Not used in DDR4 systems.
mc_BA[BANK_WIDTH × 8 – 1:0]	Input	DRAM bank address. Eight bits for each DRAM bank address.
mc_BG[BANK_GROUP_WIDTH × 8 – 1:0]	Input	DRAM bank group address. Eight bits for each DRAM pin.
mc_C[LR_WIDTH × 8 – 1:0]	Input	DDR4 DRAM Chip ID pin. Valid for 3DS RDIMMs only. LR_WIDTH is $\log_2(\text{StackHeight})$ where StackHeight (S_HEIGHT) is 2 or 4.
mc_CKE[CKE_WIDTH × 8 – 1:0]	Input	DRAM CKE. Eight bits for each DRAM pin.
mc_CS_n[CS_WIDTH × 8 – 1:0]	Input	DRAM CS_n. Eight bits for each DRAM pin. Active-Low.
mc_ODT[ODT_WIDTH × 8 – 1:0]	Input	DRAM ODT. Eight bits for each DRAM pin. Active-High.
mc_PAR[7:0]	Input	DRAM address parity. Eight bits for one DRAM parity pin. Note: This signal is valid for RDIMMs/LRDIMMs only.

Figure 4-14 shows the functional relationship between the PHY command/address input signals and a DDR4 command/address bus. The diagram shows an Activate command on system clock cycle N in the slot1 position. The **mc_ACT_n[3:2]** and **mc_CS_n[3:2]** are both asserted Low in cycle N, and all the other bits in cycle N are asserted High, generating an Activate in the slot1 position roughly two system clocks later and NOP/DESELECT commands on the other command slots.

On cycle N + 3, **mc_CS_n** and the **mc_ADR** bits corresponding to CAS/A15 are set to 0xFC. This asserts **mc_ADR[121:120]** and **mc_CS_n[1:0]** Low, and all other bits in cycle N + 3 High, generating a read command on slot0 and NOP/DESELECT commands on the other command slots two system clocks later. With the Activate and read command separated by three system clock cycles and taking into account the command slot position of both commands within their system clock cycle, expect the separation on the DDR4 bus to be 11 DRAM clocks, as shown in the DDR bus portion of [Figure 4-14](#).

Note: [Figure 4-14](#) shows the relative position of commands on the DDR bus based on the PHY input signals. Although the diagram shows some latency in going through the PHY to be somewhat realistic, this diagram does not represent the absolute command latency through the PHY to the DDR bus, or the system clock to DRAM clock phase alignment. The intention of this diagram is to show the concept of command slots at the PHY interface.

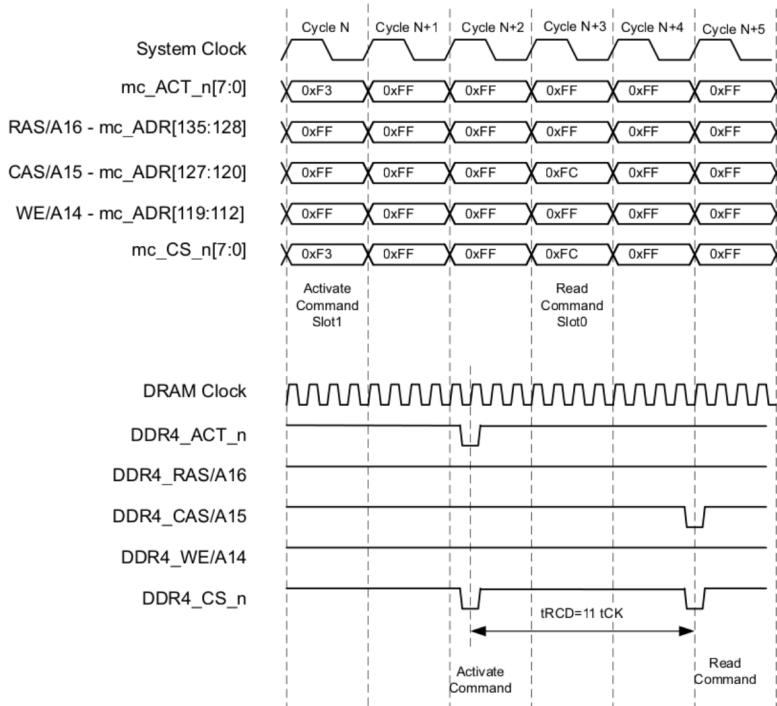


Figure 4-14: PHY Command/Address Input Signal with DDR4 Command/Address Bus

Figure 4-15 shows an example of using all four command slots in a single system clock. This example shows three commands to rank0, and one to rank1, in cycle N. BG and BA address pins are included in the diagram to spread the commands over different banks to not violate DRAM protocol. Table 4-69 lists the command in each command slot.

Table 4-69: Command Slots

Command Slot	0	1	2	3
DRAM Command	Read	Activate	Precharge	Refresh
Bank Group	0	1	2	0
Bank	0	3	1	0
Rank	0	0	0	1

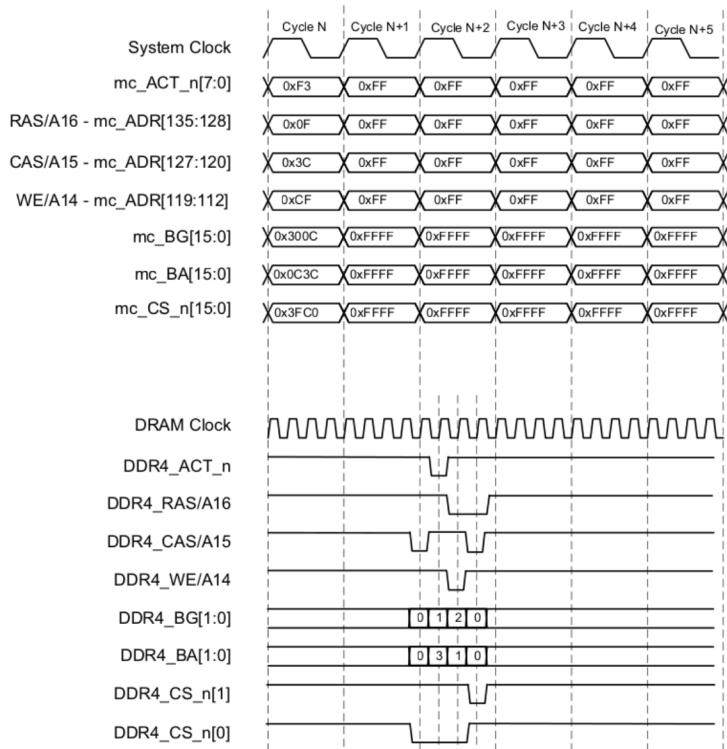


Figure 4-15: PHY Command/Address with All Four Command Slots

To understand how DRAM commands to different command slots are packed together, the following detailed example shows how to convert DRAM commands at the PHY interface to commands on the DRAM command/address bus. To convert PHY interface commands to DRAM commands, write out the PHY signal for one system clock in binary and reverse the bit order of each byte. You can also drop every other bit after the reversal because the bit pairs are required to have the same value. In the subsequent example, the `mc_BA[15:0]` signal has a cycle N value of `0x0C3C`:

Hex	0x0C3C
Binary	16'b0000_1100_0011_1100
Reverse bits in each byte	16'b0011_0000_0011_1100

Take the upper eight bits for DRAM `BA[1]` and the lower eight bits for DRAM `BA[0]` and the expected pattern on the DRAM bus is:

BA[1]	00	11	00	00
	0	1	0	0
	Low	High	Low	Low
BA[0]	00	11	11	00
	0	1	1	0
	Low	High	High	Low

This matches the DRAM `BA[1:0]` signal values of 0, 3, 1, and 0 shown in the [Figure 4-15](#).

Write Data

[Table 4-70](#) shows the write data signals for a PHY only option.

Table 4-70: Write Data

Signal	Direction	Description
<code>wrData[DQ_WIDTH × 8 – 1:0]</code>	Input	DRAM write data. Eight bits for each DQ lane on the DRAM bus. This port transfers data for an entire BL8 write on each system clock cycle. Write data must be provided to the PHY one cycle after the <code>wrDataEn</code> output signal asserts, or two cycles after if the ECC parameter is set to ON. This protocol must be followed. There is no data buffering in the PHY.
<code>wrDataMask[DM_WIDTH × 8 – 1:0]</code>	Input	DRAM write DM/DBI port. One bit for each byte of the <code>wrData</code> port, corresponding to one bit for each byte of each burst of a BL8 transfer. <code>wrDataMask</code> is transferred on the same system clock cycle as <code>wrData</code> . Active-High. For DDR3 interface, <code>wrDataMask</code> port appears in the Data Mask enabled option in Vivado IDE. For DDR4 interface, <code>wrDataMask</code> port appears in the “Data Mask and DBI” Vivado IDE option values of <code>DM_NO_DBI</code> and <code>DM_DBI_RD</code> .

Table 4-70: Write Data (Cont'd)

Signal	Direction	Description
wrDataEn	Output	Write data required. PHY asserts this port for one cycle for each write CAS command. Your design must provide wrData and wrDataMask at the PHY input ports on the cycle after wrDataEn asserts, or two cycles after if the ECC parameter is set to ON.
wrDataAddr[DATA_BUF_ADDR_WIDTH - 1:0]	Output	Optional control signal. PHY stores and returns a data buffer address for each in-flight write CAS command. The wrDataAddr signal returns the stored addresses. It is only valid when the PHY asserts wrDataEn. You can use this signal to manage the process of sending write data into the PHY for a write CAS command, but this is completely optional.
tCWL[5:0]	Output	Optional control signal. This output indicates the CAS write latency used in the PHY.
dBufAddr[DATA_BUF_ADDR_WIDTH - 1:0]	Input	Reserved. Should be tied Low.

Read Data

Table 4-71 shows the read data signals for a PHY only option.

Table 4-71: Read Data

Signal	Direction	Description
rdData[DQ_WIDTH × 8 – 1:0]	Output	DRAM read data. Eight bits for each DQ lane on the DRAM bus. This port transfers data for an entire BL8 read on each system clock cycle. rdData is only valid when the rdDataEn, per_rd_done, or rmw_rd_done is asserted. Your design must consume the read data when rdDataEn one of these "data valid" signals asserts. There is no data buffering in the PHY.
rdDataEn	Output	Read data valid. This signal asserts High to indicate that the rdData and rdDataAddr signals are valid. rdDataEn asserts High for one system clock cycle for each BL8 read, unless the read was tagged as a special type of read. See the optional per_rd_done and rmw_rd_done signals for details on special reads. rdData must be consumed when rdDataEn asserts or data is lost. Active-High.
rdDataAddr[DATA_BUF_ADDR_WIDTH - 1:0]	Output	Optional control signal. PHY stores and returns a data buffer address for each in-flight read CAS command. The rdDataAddr signal returns the stored addresses. It is only valid when the PHY asserts rdDataEn, per_rd_done, or rmw_rd_done. Your design can use this signal to manage the process of capturing and storing read data provided by the PHY, but this is completely optional.

Table 4-71: Read Data (Cont'd)

Signal	Direction	Description
per_rd_done	Output	Optional read data valid signal. This signal indicates that a special type of read has completed and its associated rdData and rdDataAddr signals are valid. When PHY input winInjTxn is asserted High at the same time as mcRdCAS, the read is tagged as a special type of read, and per_rd_done asserts instead of rdDataEn when data is returned.
rmw_rd_done	Output	Optional read data valid signal. This signal indicates that a special type of read has completed and its associated rdData and rdDataAddr signals are valid. When PHY input winRmw is asserted High at the same time as mcRdCAS, the read is tagged as a special type of read, and rmw_rd_done asserts instead of rdDataEn when data is returned.
rdDataEnd	Output	Unused. Tied High.

PHY Control

Table 4-72 shows the PHY control signals for a PHY only option.

Table 4-72: PHY Control

Signal	Direction	Description
calDone	Output	Indication that the DRAM is powered up, initialized, and calibration is complete. This indicates that the PHY interface is available to send commands to the DRAM. Active-High.
mcRdCAS	Input	Read CAS command issued. This signal must be asserted for one system clock if and only if a read CAS command is asserted on one of the command slots at the PHY command/address input ports. Hold at 0x0 until calDone asserts. Active-High.
mcWrCAS	Input	Write CAS command issued. This signal must be asserted for one system clock if and only if a write CAS command is asserted on one of the command slots at the PHY command/address input ports. Hold at 0x0 until calDone asserts. Active-High.
winRank[1:0]	Input	Target rank for CAS commands. This value indicates which rank a CAS command is issued to. It must be valid when either mcRdCAS or mcWrCAS is asserted. The PHY passes the value from this input to the XIPHY to select the calibration results for the target rank of a CAS command in multi-rank systems. In a single-rank system, this input port can be tied to 0x0.

Table 4-72: PHY Control (Cont'd)

Signal	Direction	Description
mcCasSlot[1:0]	Input	CAS command slot select. The PHY only supports CAS commands on even command slots. mcCasSlot indicates which of these two possible command slots a read CAS or write CAS was issued on. mcCasSlot is used by the PHY to generate XIPHY control signals, like DQ output enables, that need DRAM clock cycle resolution relative to the command slot used for a CAS command. Valid values after calDone asserts are 0x0 and 0x2. Hold at 0x0 until calDone asserts. This signal must be valid if mcRdCAS or mcWrCAS is asserted. For more information, see the CAS Command Timing Limitations, page 181 .
mcCasSlot2	Input	CAS slot 2 select. mcCasSlot2 serves a similar purpose as the mcCasSlot[1:0] signal, but mcCasSlot2 is used in timing critical logic in the PHY. Ideally mcCasSlot2 should be driven from separate flops from mcCasSlot[1:0] to allow synthesis/implementation to better optimize timing. mcCasSlot2 and mcCasSlot[1:0] must always be consistent if mcRdCAS or mcWrCAS is asserted. To be consistent, the following must be TRUE: mcCasSlot2==mcCasSlot[1]. Hold at 0x0 until calDone asserts. Active-High.
winInjTxn	Input	Optional read command type indication. When winInjTxn is asserted High on the same cycle as mcRdCAS, the read does not generate an assertion on rdDataEn when it completes. Instead, the per_rd_done signal asserts, indicating that a special type of read has completed and that its data is valid on the rdData output. In DDR3/DDR4 SDRAM controller designs, the winInjTxn/per_rd_done signals are used to track non-system read traffic by asserting winInjTxn only on read commands issued for the purpose of VT tracking.
winRmw	Input	Optional read command type indication. When winRmw is asserted High on the same cycle as mcRdCAS, the read does not generate an assertion on rdDataEn when it completes. Instead, the rmw_rd_done signal asserts, indicating that a special type of read has completed and that its data is valid on the rdData output. In DDR3/DDR4 SDRAM controller designs, the winRmw/rmw_rd_done signals are used to track reads issued as part of a read-modify-write flow. The DDR3/DDR4 SDRAM controller asserts winRmw only on read commands that are issued for the read phase of a RMW sequence.

Table 4-72: PHY Control (Cont'd)

Signal	Direction	Description
winBuf[DATA_BUF_ADDR_WIDTH - 1:0]	Input	Optional control signal. When either mcRdCAS or mcWrCAS is asserted, PHY stores the value on the winBuf signal. The value is returned on rdDataAddr or wrDataAddr, depending on whether mcRdCAS or mcWrCAS was used to capture winBuf. In DDR3/DDR4 SDRAM controller designs, these signals are used to track the data buffer address used to source write data or sink read return data.
gt_data_ready	Input	Update VT Tracking. This signal triggers the PHY to read RIU registers in the XIPHY that measure how well the DQS Gate signal is aligned to the center of the read DQS preamble, and then adjust the alignment if needed. This signal must be asserted periodically to keep the DQS Gate aligned as voltage and temperature drift. For more information, see VT Tracking, page 183 . Hold at 0x0 until calDone asserts. Active-High.

[Figure 4-16](#) shows a write command example. On cycle N, write command "A" is asserted on the PHY command/address inputs in the slot0 position. The `mcWrCAS` input is also asserted on cycle N, and a valid rank value is asserted on the `winRank` signal. In [Figure 4-16](#), there is only one `cs_n` pin, so the only valid `winRank` value is 0x0. The `mcCasSlot[1:0]` and `mcCasSlot2` signals are valid on cycle N, and specify slot0.

Write command "B" is then asserted on cycle N + 1 in the slot2 position, with `mcWrCAS`, `winRank`, `mcCasSlot[1:0]`, and `mcCasSlot2` asserted to valid values as well. On cycle M, PHY asserts `wrDataEn` to indicate that `wrData` and `wrDataMask` values corresponding to command A need to be driven on cycle M + 1.

[Figure 4-16](#) shows the data and mask widths assuming an 8-bit DDR4 DQ bus width. The delay between cycle N and cycle M is controlled by the PHY, based on the CWL and AL settings of the DRAM. `wrDataEn` also asserts on cycle M + 1 to indicate that `wrData` and `wrDataMask` values for command B are required on cycle M + 2. Although this example shows that `wrDataEn` is asserted on two consecutive system clock cycles, you should not assume this will always be the case, even if `mcWrCAS` is asserted on consecutive clock cycles as is shown here. There is no data buffering in the PHY and data is pulled into the PHY just in time. Depending on the CWL/AL settings and the command slot used, consecutive `mcWrCAS` assertions might not result in consecutive `wrDataEn` assertions.

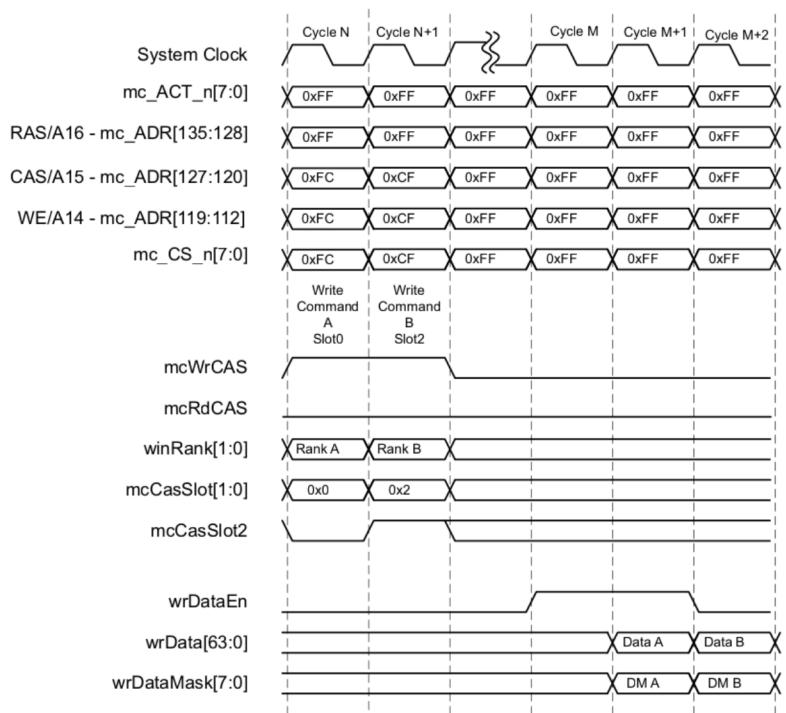


Figure 4-16: Write Command Example

Figure 4-17 shows a read command example. Read commands are issued on cycles N and N + 1 in slot positions 0 and 2, respectively. The **mcRdCAS**, **winRank**, **mcCasSlot**, and **mcCasSlot2** are asserted on these cycles as well. On cycles M + 1 and M + 2, PHY asserts **rdDataEn** and **rdData**.

Note: The separation between N and M + 1 is much larger than in the write example (Figure 4-16). In the read case, the separation is determined by the full round trip latency of command output, DRAM CL/AL, and data input through PHY.

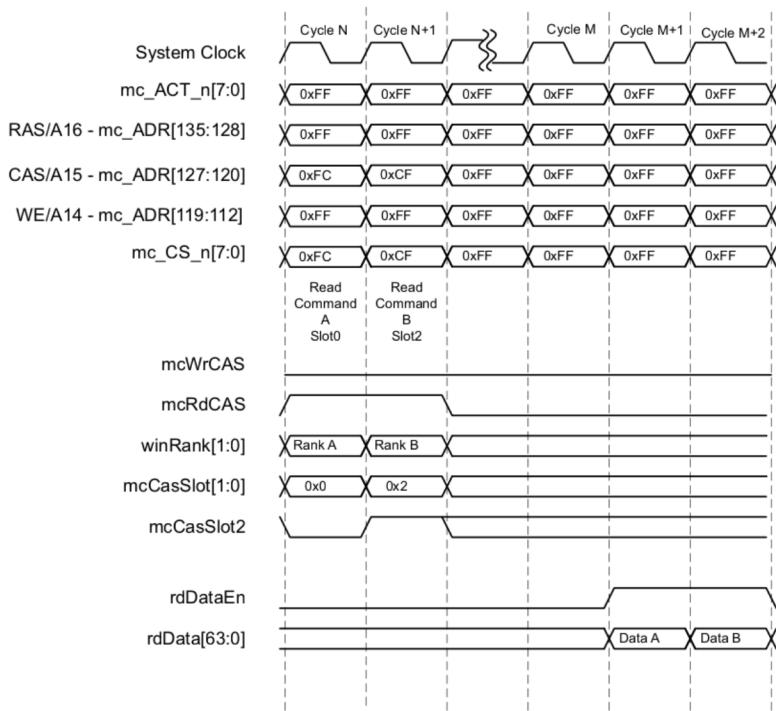


Figure 4-17: Read Command Example

Debug

The debug signals are explained in [Debug Tools, page 583](#).

PHY Only Parameters

All PHY parameters are configured by the DDR3/DDR4 SDRAM software. [Table 4-73](#) describes the PHY parameters. These parameter values must not be modified in the DDR3/DDR4 SDRAM generated designs. The parameters are set during core generation. The core must be regenerated to change any parameter settings.

Table 4-73: PHY Only Parameters

Parameter Name	Default Value	Allowable Values	Description
ADDR_WIDTH	18	DDR4 18.. 17 DDR3 16.. 13	Number of DRAM Address pins
BANK_WIDTH	2	DDR4 2 DDR3 3	Number of DRAM Bank Address pins

Table 4-73: PHY Only Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
BANK_GROUP_WIDTH	2	DDR4 2.. 1 DDR3 N/A	Number of DRAM Bank Group pins
CK_WIDTH	1	2.. 1	Number of DRAM Clock pins
CKE_WIDTH	1	2.. 1	Number of DRAM CKE pins
CS_WIDTH	1	2.. 1	Number of DRAM CS pins
ODT_WIDTH	1	4.. 1	Number of DRAM ODT pins
DRAM_TYPE	"DDR4"	"DDR4," "DDR3,"	DRAM Technology
DQ_WIDTH	16	Minimum = 8 Must be multiple of 8	Number of DRAM DQ pins in the channel
DQS_WIDTH	2	Minimum = 1 x8 DRAM – 1 per DQ byte x4 DRAM – 1 per DQ nibble	Number of DRAM DQS pins in the channel
DM_WIDTH	2	Minimum = 0 x8 DRAM – 1 per DQ byte x4 DRAM – 0	Number of DRAM DM pins in the channel
DATA_BUF_ADDR_WIDTH	5	5	Number of data buffer address bits stored for a read or write transaction
ODTWR	0x8421	0xFFFF .. 0x0000	Reserved for future use
ODTWRDEL	8	Set to CWL	Reserved for future use
ODTWRDUR	6	7.. 6	Reserved for future use
ODTRD	0x0000	0xFFFF.. 0x0000	Reserved for future use
ODTRDDEL	11	Set to CL	Reserved for future use
ODTRDDUR	6	7.. 6	Reserved for future use
ODTWR0DEL ODTWR0DUR ODTRD0DEL ODTRD0DUR ODTNOP	N/A	N/A	Reserved for future use
MR0	0x630	Legal SDRAM configuration	DRAM MR0 setting
MR1	0x101	Legal SDRAM configuration	DRAM MR1 setting
MR2	0x10	Legal SDRAM configuration	DRAM MR2 setting
MR3	0x0	Legal SDRAM configuration	DRAM MR3 setting
MR4	0x0	Legal SDRAM configuration	DRAM MR4 setting. DDR4 only.

Table 4-73: PHY Only Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
MR5	0x400	Legal SDRAM configuration	DRAM MR5 setting. DDR4 only.
MR6	0x800	Legal SDRAM configuration	DRAM MR6 setting. DDR4 only.
SLOT0_CONFIG	0x1	0x1 0x3 0x5 0xF	For more information, see SLOT0_CONFIG .
SLOT1_CONFIG	0x0	0x0 0x2 0xC 0xA	For more information, see SLOT0_CONFIG .
SLOT0_FUNC_CS	0x1	0x1 0x3 0x5 0xF	Memory bus CS_n pins used to send all DRAM commands including MRS to memory. Each bit of the parameter represents 1-bit of the CS_n bus, for example, the LSB indicates CS_n[0], and the MSB indicates CS_n[3]. For DIMMs this parameter specifies the CS_n pins connected to DIMM slot 0. Note: slot 0 used here should not be confused with the "command slot0" term used in the description of the PHY command/address interface. For more information, see SLOT0_FUNC_CS .
SLOT1_FUNC_CS	0x0	0x0 0x2 0xC 0xA	See the SLOT0_FUNC_CS description. The only difference is that SLOT1_FUNC_CS specifies CS_n pins connected to DIMM slot 1.
REG_CTRL	OFF	ON OFF	Enable RDIMM RCD initialization and calibration
CA_MIRROR	OFF	ON OFF	Enable Address mirroring. This parameter is set to ON for the DIMMs that support address mirroring.
DDR4_REG_RC03	0x30	Legal RDIMM RCD configuration	RDIMM RCD control word 03
DDR4_REG_RC04	0x40	Legal RDIMM RCD configuration	RDIMM RCD control word 04
DDR4_REG_RC05	0x50	Legal RDIMM RCD configuration	RDIMM RCD control word 05
tCK	938	Minimum 833	DRAM clock period in ps
tXPR	72	Minimum 1. DRAM tXPR specification in system clocks	See JEDEC DDR SDRAM specification [Ref 1].

Table 4-73: PHY Only Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
tMOD	6	Minimum 1. DRAM tMOD specification in system clocks	See JEDEC DDR SDRAM specification [Ref 1].
tMRD	2	Minimum 1. DRAM tMRD specification in system clocks	See JEDEC DDR SDRAM specification [Ref 1].
tZQINIT	256	Minimum 1. DRAM tZQINIT specification in system clocks	See JEDEC DDR SDRAM specification [Ref 1].
TCQ	100	100	Flop clock to Q in ps. For simulation purposes only.
EARLY_WR_DATA	OFF	OFF	Reserved for future use
EXTRA_CMD_DELAY	0	2.. 0	Added command latency in system clocks. Added command latency is required for some configurations. See details in CL/CWL section.
ECC	"OFF"	OFF	Enables early wrDataEn timing for DDR3/DDR4 SDRAM generated controllers when set to ON. PHY only designs must set this to OFF.
DM_DBI	"DM_NODBI"	"NONE" "DM_NODBI" "DM_DBIRD" "NODM_DBIRW" "NODM_DBIRD" "NODM_DBIWRD" "NODM_NODBI"	DDR4 DM/DBI configuration. For details, see Table 4-75.
USE_CS_PORT	1	0 = no CS_n pins 1 = CS_n pins used	Controls whether or not CS_n pins are connect to DRAM. If there are no CS_n pins the PHY initialization and training logic issues NOPs between DRAM commands. If there are no CS_n pins, The DRAM chip select pin (CS#) must be tied Low externally at the DRAM.
DRAM_WIDTH	8	16, 8, 4	DRAM component DQ width
RANKS	1	4, 2, 1	Number of ranks in the memory subsystem
nCK_PER_CLK	4	4	Number of DRAM clocks per system clock
C_FAMILY	"kintexu"	"kintexu" "virtexu"	Device information used by MicroBlaze controller in the PHY.
BYTES	4	Minimum 3	Number of XIPHY "bytes" used for data, command, and address

Table 4-73: PHY Only Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
DBYTES	2	Minimum 1	Number of bytes in the DRAM DQ bus
IOBTYPE	{39'b001_001_00 1_001_001_101_ 101_001_001_00 1_001_001_001, 39'b001_001_00 1_001_001_001_00 001_001_001_00 1_001_001_001_001, 39'b000_011_01 1_011_011_111_ 111_011_011_01 1_011_001_011, 39'b001_011_01 1_011_011_111_ 111_011_011_01 1_011_001_011_011} 1_011_001_011_011}	3'b000 = Unused pin 3'b 001 = Single-ended output 3'b 010 = Single-ended input 3'b011 = Single-ended I/O 3'b100 = Unused pin 3'b 101 = Differential Output 3'b 110 = Differential Input 3'b 111 = Differential INOUT	IOB setting
PLL_WIDTH	1	DDR3/DDR4 SDRAM generated values	Number of PLLs
CLKOUTPHY_MODE	"VCO_2X"	VCO_2X	Determines the clock output frequency based on the VCO frequency for the BITSLICE_CONTROL block
PLLCLK_SRC	0	0 = pll_clk0 1 = pll_clk1	XIPHY PLL clock source
DIV_MODE	0	0 = DIV4 1 = DIV2	XIPHY controller mode setting
DATA_WIDTH	8	8	XIPHY parallel input data width
CTRL_CLK	0x3	0 = Internal, local div_clk used 1 = External RIU clock used	Internal or external XIPHY clock for the RIU
INIT	((15 × BYTES){1'b1})	1'b0 1'b1	3-state bitslice OSERDES initial value
RX_DATA_TYPE	{15'b000000_00_0000_00, 15'b000000_00_0000_00, 15'b011110_10_11110_01, 15'b011110_10_11110_01}	2'b00 = None 2'b01 = DATA(DQ_EN) 2'b10 = CLOCK(DQS_EN) 2'b11 = DATA_AND_CLOCK	XIPHY bitslice setting

Table 4-73: PHY Only Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
TX_OUTPUT_PHASE_90	{13'b111111111111, 1111, 13'b111111111111 111, 13'b0000011000010, 13'b1000011000010}	1'b0 = No offset 1'b1 = 90° offset applied	XIPHY setting to apply 90° offset on a given bitslice
RXTX_BITSLICE_EN	{13'b11111011111111, 1111, 13'b11111111111111 111, 13'b01111011111111, 13'b11111011111111}	1'b0 = No bitslice 1'b1 = Bitslice enabled	XIPHY setting to enable a bitslice
NATIVE_ODLAY_BYPASS	{(13 × BYTES){1'b0}}	1'b0 = FALSE 1'b1 = TRUE (Bypass)	Bypass the ODELAY on output bitslices
EN_OTHER_PCLK	{BYTES(2'b01)}	1'b 0 = FALSE (not used) 1'b 1 = TRUE (used)	XIPHY setting to route capture clock from other bitslice
EN_OTHER_NCLK	{BYTES(2'b01)}	1'b 0 = FALSE (not used) 1'b 1 = TRUE (used)	XIPHY setting to route capture clock from other bitslice
RX_CLK_PHASE_P	{((BYTES – DBYTES){2'b00}), {DBYTES{2'b11}}}	2'b00 for Address/Control, 2'b11 for Data	XIPHY setting to shift the read clock DQS_P by 90° relative to the DQ
RX_CLK_PHASE_N	{((BYTES – DBYTES){2'b00}), {DBYTES{2'b11}}}	2'b00 for Address/Control, 2'b11 for Data	XIPHY setting to shift the read clock DQS_N by 90° relative to the DQ
TX_GATING	{((BYTES – DBYTES){2'b00}), {DBYTES{2'b11}}}	2'b00 for Address/Control, 2'b11 for Data	Write DQS gate setting for the XIPHY
RX_GATING	{((BYTES – DBYTES){2'b00}), {DBYTES{2'b11}}}	2'b00 for Address/Control, 2'b11 for Data	Read DQS gate setting for the XIPHY
EN_DYN_ODLY_MODE	{((BYTES – DBYTES){2'b00}), {DBYTES{2'b11}}}	2'b00 for Address/Control, 2'b11 for Data	Dynamic loading of the ODELAY by XIPHY
BANK_TYPE	"HP_IO"	"HP_IO" "HR_IO"	Indicates whether selected bank is HP or HR
SIM_MODE	"BFM"	"FULL", "BFM"	Flag to set if the XIPHY is used ("UNISIM") or the behavioral model for simulation speed up.
SELF_CALIBRATE	{(2 × BYTES){1'b0}}	{(2 × BYTES){1'b0}} for simulation, {(2 × BYTES){1'b1}} for hardware	BISC self calibration

Table 4-73: PHY Only Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
BYPASS_CAL	"FALSE"	"TRUE" for simulation, "FALSE" for hardware	Flag to turn calibration ON/OFF
CAL_WRLVL	"FULL"	"FULL"	Flag for calibration, write-leveling setting
CAL_DQS_GATE	"FULL"	"FULL"	Flag for calibration, DQS gate setting
CAL_RDLVL	"FULL"	"FULL"	Flag for calibration, read training setting
CAL_WR_DQS_DQ	"FULL"	"FULL"	Flag for calibration, write DQS-to-DQ setting
CAL_COMPLEX	"FULL"	"SKIP", "FULL"	Flag for calibration, complex pattern setting
CAL_RD_VREF	"SKIP"	"SKIP", "FULL"	Flag for calibration, read V_{REF} setting
CAL_WR_VREF	"SKIP"	"SKIP", "FULL"	Flag for calibration, write V_{REF} setting
CAL_JITTER	"FULL"	"FULL", "NONE"	Reserved for verification. Speed up calibration simulation. Must be set to "FULL" for all hardware test cases.
t200us	53305 decimal	0x3FFFF.. 1	Wait period after BISC complete to DRAM reset_n deassertion in system clocks
t500us	133263 decimal	0x3FFFF.. 1	Wait period after DRAM reset_n deassertion to CKE assertion in system clocks

EXTRA_CMD_DELAY Parameter

Depending on the number of ranks, ECC mode, and DRAM latency configuration, PHY must be programmed to add latency on the DRAM command address bus. This provides enough pipeline stages in the PHY programmable logic to close timing and to process **mcWrCAS**. Added command latency is generally needed at very low CWL in single-rank configurations, or in multi-rank configurations. Enabling ECC might also require adding command latency, but this depends on whether your controller design (outside the PHY) depends on receiving the **wrDataEn** signal a system clock cycle early to allow for generating ECC check bits.

The EXTRA_CMD_DELAY parameter is used to add one or two system clock cycles of delay on the DRAM command/address path. The parameter does not delay the **mcWrCAS** or **mcRdCAS** signals. This gives the PHY more time from the assertion of **mcWrCAS** or **mcRdCAS** to generate XIPHY control signals. To the PHY, an EXTRA_CMD_DELAY setting of one or two is the same as having a higher CWL or AL setting.

[Table 4-74](#) shows the required EXTRA_CMD_DELAY setting for various configurations of CWL, CL, and AL.

Table 4-74: EXTRA_CMD_DELAY Configuration Settings

DRAM Configuration			Required EXTRA_CMD_DELAY	
DRAM CAS Write Latency CWL	DRAM CAS Latency CL	DRAM Additive Latency MR1[4:3]	Single-Rank without ECC	Single-Rank with ECC or Multi-Rank
5	5	0	1	2
5	5	1	0	1
5	5	2	1	2
5	5	3	1	2
5	6	0	1	2
5	6	1	0	1
5	6	2	0	1
5	6	3	0	1
6	6	0	1	2
6	6	1	0	1
6	6	2	0	1
6	6	3	0	1
6	7	0	1	2
6	7	1	0	1
6	7	2	0	1
6	7	3	0	1
6	8	0	1	2
6	8	1	0	0
6	8	2	0	1
6	8	3	0	1
7	7	0	1	2
7	7	1	0	0
7	7	2	0	1
7	7	3	0	1
7	8	0	1	2
7	8	1	0	0
7	8	2	0	0
7	8	3	0	0
7	9	0	1	2
7	9	1	0	0
7	9	2	0	0

Table 4-74: EXTRA_CMD_DELAY Configuration Settings (Cont'd)

DRAM Configuration			Required EXTRA_CMD_DELAY	
DRAM CAS Write Latency CWL	DRAM CAS Latency CL	DRAM Additive Latency MR1[4:3]	Single-Rank without ECC	Single-Rank with ECC or Multi-Rank
7	9	3	0	0
7	10	0	1	2
7	10	1	0	0
7	10	2	0	0
7	10	3	0	0
8	8	0	1	2
8	8	1	0	0
8	8	2	0	0
8	8	3	0	0
8	9	0	1	2
8	9	1	0	0
8	9	2	0	0
8	9	3	0	0
8	10	0	1	2
8	10	1	0	0
8	10	2	0	0
8	10	3	0	0
8	11	0	1	2
8	11	1	0	0
8	11	2	0	0
8	11	3	0	0
9 to 12	X	0	0	1
9 to 12	X	1, 2, or 3	0	0
≥13	X	0	0	0
≥13	X	1, 2, or 3	0	0

DM_DBI Parameter

The PHY supports the DDR4 DBI function on the read path and write path. Table 4-75 show how read and write DBI can be enabled separately or in combination.

When write DBI is enabled, Data Mask is disabled. The DM_DBI parameter only configures the PHY and the MRS parameters must also be set to configure the DRAM for DM/DBI.

Table 4-75: DM_DBI PHY Settings

DM_DBI Parameter Value	PHY Read DBI	PHY Write DBI	PHY Write Data Mask
None	Disabled	Disabled	Disabled
DM_NODBI	Disabled	Disabled	Enabled
DM_DBIRD	Enabled	Disabled	Enabled
NODM_DBIWR	Disabled	Enabled	Disabled
NODM_DBIRD	Enabled	Disabled	Disabled
NODM_DBIWRD	Enabled	Enabled	Disabled
NODM_NODBI	Disabled	Disabled	Disabled

The allowed values for the DM_DBI option in the GUI are as follows for x8 and x16 parts ("X" indicates supported and "--" indicates not supported):

Table 4-76: DM_DBI Options

Option Value	Native		AXI	
	ECC Disable	ECC Enable	ECC Disable	ECC Enable
DM_NO_DBI ⁽¹⁾	X	-	X	-
DM_DBI_RD	X	-	X	-
NO_DM_DBI_RD	X	X	-	X
NO_DM_DBI_WR	X	X	-	X
NO_DM_DBI_WR_RD	X	X	-	X
NO_DM_NO_DBI ⁽²⁾	-	X	-	X

Notes:

1. Default option for ECC disabled interfaces.
2. Default option for ECC enabled interfaces.

For x4 parts, the supported DM_DBI option value is "NONE."

DBI can be enabled to reduce power consumption in the interface by reducing the total number of DQ signals driven Low and thereby reduce noise in the V_{CCO} supply. For further information where this might be useful for improved signal integrity, see Answer Record AR 70006.

CAS Command Timing Limitations

The PHY only supports CAS commands on even command slots, that is, 0 and 2. This limitation is due to the complexity of the PHY logic driven by the PHY control inputs, like the mcWrCAS and mcRdCAS signals, not the actual DRAM command signals like mc_ACT_n[7:0], which just pass through the PHY after calDone asserts. The PHY logic is

complex because it generates XIPHY control signals based on the DRAM CWL and CL values with DRAM clock resolution, not just system clock resolution.

Supporting two different command slots for CAS commands adds a significant amount of logic on the XIPHY control paths. There are very few pipeline stages available to break up the logic due to protocol requirements of the XIPHY. CAS command support on all four slots would further increase the complexity and degrade timing.

Minimum Write CAS Command Spacing

The minimum Write CAS to Write CAS command spacing to different ranks is eight DRAM clocks. This is a PHY limitation. If you violate this timing, the PHY might not have enough time to switch its internal delay settings and drive Write DQ/DQS on the DDR bus with correct timing. The internal delay settings are determined during calibration, and it varies with system layout.

Following the memory system layout guidelines ensures that a spacing of eight DRAM clocks is sufficient for correct operation. Write to Write timing to the same rank is limited only by the DRAM specification and the command slot limitations for CAS commands discussed earlier.

System Considerations for CAS Command Spacing

System layout and timing uncertainties should be considered in how your custom controller sets minimum CAS command spacing. The controller must space the CAS commands so that there are no DRAM timing violations and no DQ/DQS bus drive fights. When a DDR3/DDR4 SDRAM generated memory controller is instantiated, the layout guidelines are considered and command spacing is adjusted accordingly for a worst case layout.

Consider Read to Write command spacing, the JEDEC® DRAM specification [Ref 1] shows the component requirement as: $RL + BL/2 + 2 - WL$. This formula only spaces the Read DQS post-amble and Write DQS preamble by one DRAM clock on an ideal bus with no timing skews. Any DQS flight time, write leveling uncertainty, jitter, etc. reduces this margin. When these timing errors add up to more than one DRAM clock, there is a drive fight at the FPGA DQS pins which likely corrupts the Read transaction. A DDR3/DDR4 SDRAM generated controller uses the following formula to delay Write CAS after a Read CAS to allow for a worst case timing budget for a system following the layout guidelines: $RL + BL/2 + 4 - WL$.

Read CAS to Read CAS commands to different ranks must also be spaced by your custom controller to avoid drive fights, particularly when reading first from a "far" rank and then from a "near" rank. A DDR3/DDR4 SDRAM generated controller spaces the Read CAS commands to different ranks by at least six DRAM clock cycles.

Write CAS to Read CAS to the same rank is defined by the JEDEC DRAM specification [Ref 1]. Your controller must follow this DRAM requirement, and it ensures that there is no possibility of drive fights for Write to Read to the same rank. Write CAS to Read CAS spacing

to different ranks, however, must also be limited by your controller. This spacing is not defined by the JEDEC DRAM specification [Ref 1] directly.

Write to Read to different ranks can be spaced much closer together than Write to Read to the same rank, but factors to consider include write leveling uncertainty, jitter, and tDQSCK. A DDR3/DDR4 SDRAM generated controller spaces Write CAS to Read CAS to different ranks by at least six DRAM clocks.

Additive Latency

The PHY supports DRAM additive latency. The only effect on the PHY interface due to enabling Additive Latency in the MRS parameters is in the timing of the `wrDataEn` signal after `mcWrCAS` assertion. The PHY takes the AL setting into account when scheduling `wrDataEn`. You can also find the `rdDataEn` asserts much later after `mcRdCAS` because the DRAM returns data much later. The AL setting also has an impact on whether or not the `EXTRA_CMD_DELAY` parameter needs to be set to a non-zero value.

VT Tracking

The PHY requires read commands to be issued at a minimum rate to keep the read DQS gate signal aligned to the read DQS preamble after `calDone` is asserted. In addition, the `gt_data_ready` signal needs to be pulsed at regular intervals to instruct the PHY to update its read DQS training values in the RIU. Finally, the PHY requires periodic gaps in read traffic to allow the XIPHY to update its gate alignment circuits with the values the PHY programs into the RIU. Specifically, the PHY requires the following after `calDone` asserts:

1. At least one read command every 1 μ s. For a multi-rank system any rank is acceptable.
2. The `gt_data_ready` signal is asserted for one system clock cycle after `rdDataEn` or `per_rd_done` signal asserts at least once within each 1 μ s interval.
3. There is a three contiguous system clock cycle period with no read CAS commands asserted at the PHY interface every 1 μ s.

The PHY cannot interrupt traffic to meet these requirements. It is therefore your custom Memory Controller's responsibility to issue DRAM commands and assert the `gt_data_ready` input signal in a way that meets the above requirements.

Figure 4-18 shows two examples where the custom controller must interrupt normal traffic to meet the VT tracking requirements. The first example is a High read bandwidth workload with `mcRdCAS` asserted continuously for almost 1 μ s. The controller must stop issuing read commands for three contiguous system clock cycles once each 1 μ s period, and assert `gt_data_ready` once per period.

The second example is a High write bandwidth workload with `mcWrCAS` asserted continuously for almost 1 μ s. The controller must stop issuing writes, issue at least one read command, and then assert `gt_data_ready` once per 1 μ s period.



IMPORTANT: The controller must not violate DRAM protocol or timing requirements during this process.

Note: The VT tracking diagrams are not drawn to scale.

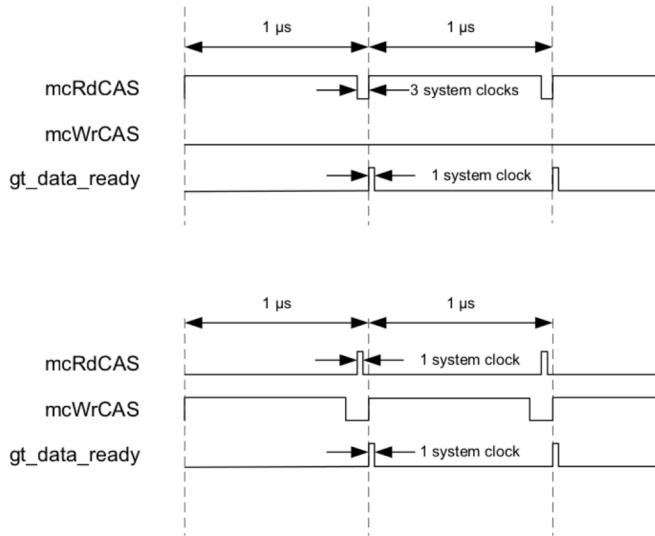


Figure 4-18: VT Tracking Diagrams

A workload that has a mix of read and write traffic in every 1 μ s interval might naturally meet the first and third VT tracking requirements listed above. In this case, the only extra step required is to assert the **gt_data_ready** signal every 1 μ s and regular traffic would not be interrupted at all. The custom controller, however, is responsible for ensuring all three requirements are met for all workloads. DDR3/DDR4 SDRAM generated controllers monitor the **mcRdCAS** and **mcWrCAS** signals and decide each 1 μ s period what actions, if any, need to be taken to meet the VT tracking requirements. Your custom controller can implement any scheme that meets the requirements described here.

Refresh and ZQ

After **calDone** is asserted by the PHY, periodic DRAM refresh and ZQ calibration are the responsibility of your custom Memory Controller. Your controller must issue refresh and ZQ commands, meet DRAM refresh and ZQ interval requirements, while meeting all other DRAM protocol and timing requirements. For example, if a refresh is due and you have open pages in the DRAM, you must precharge the pages, wait tRP, and then issue a refresh command, etc. The PHY does not perform the precharge or any other part of this process for you.