PolarFire® FPGA and PolarFire SoC FPGA Memory Controller

Introduction

Microchip's PolarFire FPGAs are the fifth-generation family of non-volatile FPGA devices, built on state-of-the-art 28 nm non-volatile process technology. PolarFire FPGAs deliver the lowest power at mid-range densities. PolarFire FPGAs lower the cost of mid-range FPGAs by integrating the industry's lowest power FPGA fabric, lowest power 12.7 Gbps transceiver lane, built-in low power dual PCI Express Gen2 (EP/RP), and, on select data security (S) devices, an integrated low-power crypto co-processor.

Microchip's PolarFire SoC FPGAs are the fifth-generation family of non-volatile SoC FPGA devices, built on state-of-the-art 28 nm non-volatile process technology. The PolarFire SoC family offers industry's first RISC-V based SoC FPGAs capable of running Linux. It combines a powerful 64-bit 5x core RISC-V Microprocessor Subsystem (MSS), based on SiFive's U54-MC family, with the PolarFire FPGA fabric in a single device.

The following table lists the memory controller IP solutions in PolarFire and PolarFire SoC families.

Table 1. Memory Controller IP Solutions in PolarFire and PolarFire SoC Families

Memory Controller	PolarFire FPGA (MPF)	PolarFire SoC FPGA (MPFS)	Description
Hardened MSS DDR Controller (see MSS DDR Memory Controller)	—	✓	Supports DDR3, DDR4, LPDDR3, and LPDDR4 memory devices.
PolarFire DDR IP[1] (see 3. Fabric DDR Subsystem)	✓	✓	Supports DDR3, DDR4, and LPDDR3 memory devices.
PolarFire QDR IP[1] (see QDR Memory Controller)	✓	✓	Supports QDR II+ and QDR II+ Xtreme memory devices.
PolarFire Octal DDR PHY[1] (see 3.8 Octal DDR PHY-Only Solution)	✓	✓	Supports xSPI (JESD251), HyperBUS, and ONFI memory devices.

Note:

1. These IPs belong to the PolarFire family and can be seamlessly used in PolarFire SoC designs.

These memory controller solutions address the high-speed data transfer requirements for a wide range of applications and code execution.

© 2021 Microchip Technology Inc.
and its subsidiaries

References

- For more information about the PolarFire SoC MSS, see PolarFire SoC FPGA MSS Technical Reference Manual.
- For information about configuring the MSS DDR Controller, see PolarFire SoC Standalone MSS Configurator User Guide.
- For more information about the clocking resources, see PolarFire FPGA and PolarFire SoC FPGA Clocking Resources User Guide.
- For more information about SmartDebug, see PolarFire SmartDebug User Guide.
- For more information about PolarFire board design, see UG0726: PolarFire FPGA Board Design User Guide.
- For more information about PolarFire SoC board design, see PolarFire SoC Board Design Guidelines User Guide.

Table of Contents

© 2021 Microchip Technology Inc.
and its subsidiaries

1. Acronyms

Table 1-1. Acronyms

Acronym	Description
QDR	Quad Data Rate
SRAM	Synchronous Random Access Memory
1K	1024-bits
Word	8-bit, 9-bit data, or 18-bit data unit, depending on memory size
DDR	Double Data Rate

2. MSS DDR Memory Controller

The PolarFire SoC MSS includes a hardened DDR controller to address the memory solution requirements for a wide range of applications with varying power consumption and efficiency levels. The DDR controller along with other blocks external to MSS form the MSS DDR subsystem that can be configured to support DDR3, DDR4, LPDDR3, and LPDDR4 memory devices.

For more information about the features and functional blocks of the MSS DDR Controller, see PolarFire SoC FPGA MSS Technical Reference Manual. For information about configuring the MSS DDR Controller, see PolarFire SoC Standalone MSS Configurator User Guide.

3. Fabric DDR Subsystem

The Fabric DDR subsystem is made up of the following soft and hard blocks:

- PolarFire DDR controller (soft)
- Training logic (soft)
- I/O lane (hard)
- Phase-locked loop (PLL) (hard)

The following memory interface solutions can be created as shown in the following figures:

- Fabric DDR subsystem—consists of PolarFire DDR controller (PF_DDR IP), PLL, I/O lane, and training logic.
- DDR PHY-Only solution—consists of PLL, I/O lane, and training logic. This solution is intended for third-party DDR controller implementations.
- Octal DDR PHY-Only Solution—I/O lane, IOD logic, PLL, and soft fabric logic. This solution is intended for memory devices which use an 8-bit serial interface in DDR mode. For example, xSPI (JESD251), HyperBUS, and ONFI.

Figure 3-1. Fabric DDR Subsystem

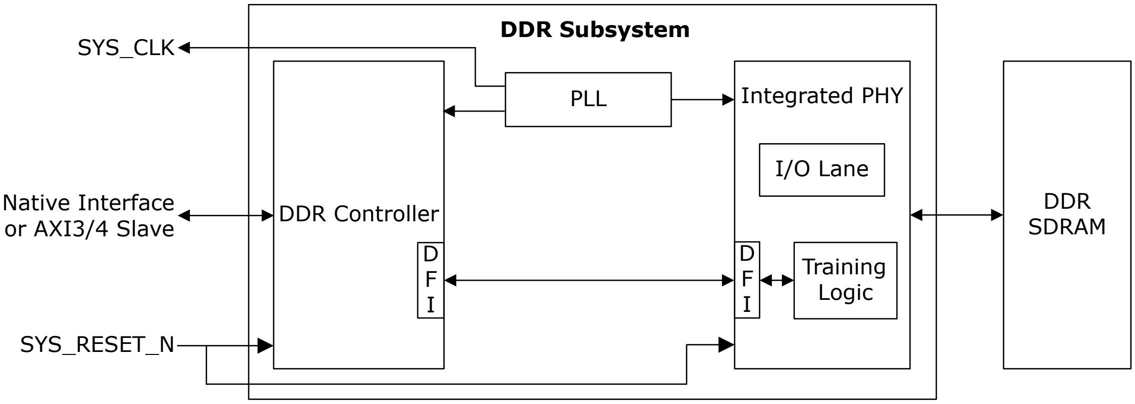


Figure 3-2. DDR PHY-Only Solution

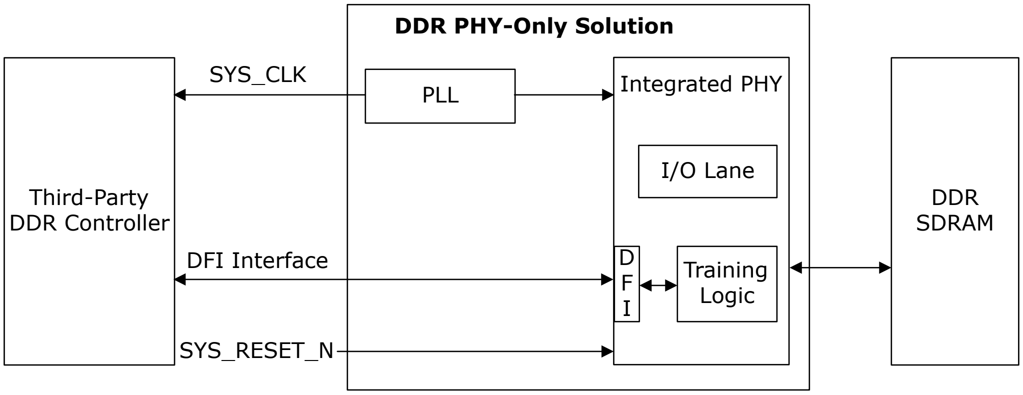
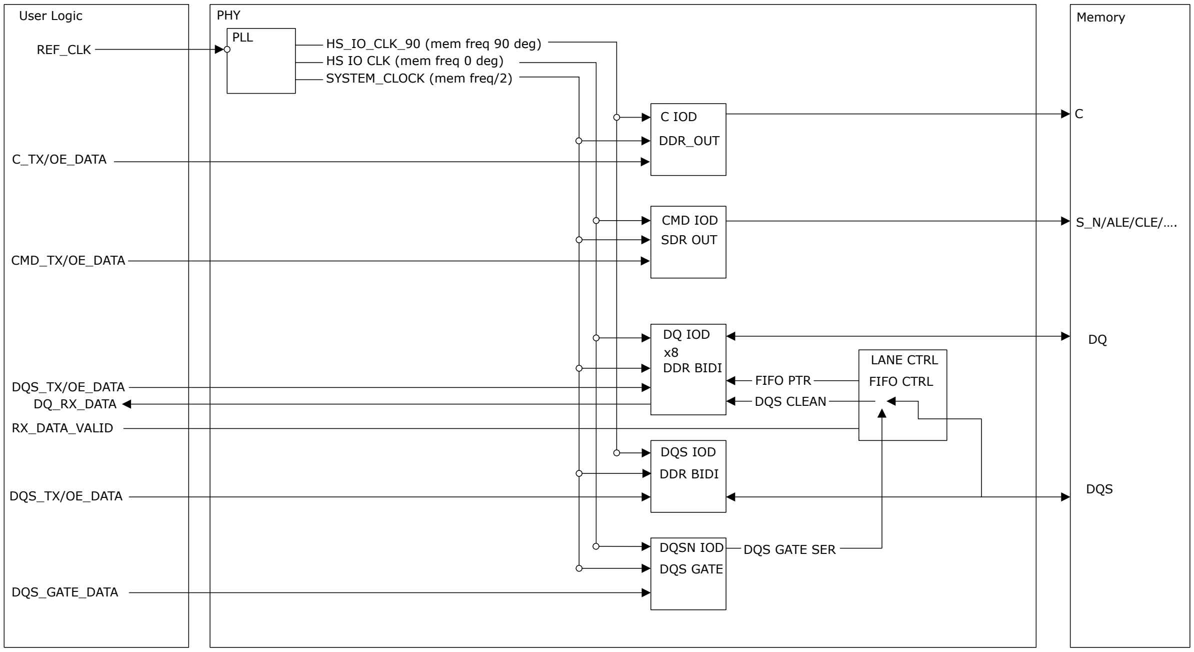


Figure 3-3. Octal DDR PHY-Only Solution



The Fabric DDR subsystem accepts read and write commands using the AMBA Advanced Extensible Interface 4 (AXI3/4) or using a native interface. The subsystem translates the AXI interface commands for the off-chip DDR SDRAM. It can also automatically perform DDR initialization, refresh, and ZQ calibration functions.

The Fabric DDR subsystem and the PHY-Only solution can be configured using the DDR Configurator from Libero® System-on-Chip (SoC) > IP catalog. One of the two configuration modes can be selected:

- Preset configuration—allows speed-based selection of memory devices for presetting memory initialization and timing parameters.
- User configuration—allows manual configuration of memory initialization and timing parameters.

3.1 Features

The following are the features of the Fabric DDR subsystem:

- Integrated hard PHY.
- Configurable to support DDR4, DDR3, DDR3L, and LPDDR3 memory devices, and dual in-line memory modules (DIMM). The subsystem supports:
  - DDR3 and DDR4 additive latency modes.
  - DDR4 temperature-controlled refresh modes.
- Up to 1600 Mbps (800 MHz DDR) data rates.
- Supports 16-bit, 32-bit (40-bit with ECC), and 64-bit (72-bit with ECC) DDR SDRAM data buses.
- Supports a maximum of 16 memory banks.
- Includes bank management logic that monitors the status of each SDRAM bank.
- Queue-based user interface that enables the DDR subsystem to optimize performance and throughput.
- Supports look-ahead precharge and auto-precharge modes.
- Supports automatic generation of initialization, ZQ calibration, and refresh sequences.
- Supports issue of commands with or without SDRAM auto-precharge.
- Supports 2T timing on SDRAM control signals (2T timing is always enabled).
- Includes multi-burst capability, allowing requests with bursts longer than that of the native memory burst.
- Single-error correction and double-error detection (SECDED) error-correcting code (ECC) protection.

3.2 Performance

The following table lists DDR3, DDR4, and LPDDR3 speeds.

Table 3-1. DDR Performance

Memory Type	I/O Type	Speed (-1)	Speed (STD)
DDR3	HSIO	1333	1067
	GPIO	1067	800
DDR4	HSIO	1600	1333
LPDDR3	HSIO	1333	800

Note: For DDR timing closure, use the SmartTime reports from Libero SoC.

The following table lists DDR3, DDR4, and LPDDR3 throughput.

Table 3-2. Throughput

Memory	Fabric Interface Frequency (MHz)	Fabric Interface Data Width	DDR Memory Data Width	Theoretical Throughput	Actual Throughput		Actual Throughput (%)	
				Max Throughput (MB/s)	Write (MB/s)	Read (MB/s)	Write (%)	Read (%)
DDR3	166.66	256	32	5344	4945.19	4987.06	92.5%	93.3%
DDR3 (AXI4)	166.66	256	32	5344	5338.89	4866.08	99.9%	91.2%
DDR4 (NI[1])	200	256	32	6400	5916.5	5866.9	92.4%	91.6%
DDR4 (AXI4)	200	256	32	6400	6395.9	5558.12	99.9%	86.84 %
LPDDR3 (NI[1])	166.66	256	32	5344	5034.4	4917.6	94.2%	92%
LPDDR3 (AXI4)	166.66	256	32	5344	5340.59	4681.22	99.9%	87.5%

Notes:

1. NI refers to native interface.
2. To get high DDR4 throughput, set tCCD_L to 4 on the Memory Timing tab of the PF_DDR4 Configurator.

3.3 Resource Utilization

The following table provides resource utilization data for the DDR3 subsystem with AXI3 and AXI4 interfaces.

Table 3-3. DDR3 Resource Utilization

Interface Type	DDR Width	AXI Width	Maximum Number of LUT4	Maximum Number of DFF	Maximum Number of µSRAM	Maximum Number of LSRAM
AXI3	16	64	10155	8373	57	6
		128	10375	9095	69	6
	32	64	14409	12107	107	2
		128	14818	12884	93	9
		256	15322	14346	115	9
	40	64	19282	13596	119	2
		128	—	—	—	—
		256	20289	15828	127	9
	64	512	25354	24849	67	58
	72	512	32280	26575	87	56
AXI4	16	64	10496	8626	33	18
		128	10715	9348	33	22
	32	64	14736	12354	45	21
		128	15083	13152	45	32
		256	15682	14614	45	39
	40	64	19500	13843	57	21
		128	20109	14664	57	32
		256	20615	16112	57	39
	64	512	25655	25102	69	72
	72	512	32495	26827	83	72

...........continued

Interface Type	DDR Width	AXI Width	Maximum Number of LUT4	Maximum Number of DFF	Maximum Number of μSRAM	Maximum Number of LSRAM
Native	16	64	7316	4456	19	2
		128	7316	4456	19	2
		256	7316	4456	19	2
		512	7316	4456	19	2
	32	64	11015	6982	29	2
		128	11015	6982	29	2
		256	11015	6982	29	2
		512	11015	6982	29	2
	40	64	15228	8133	38	2
		128	15228	8133	38	2
		256	15228	8133	38	2
		512	15228	8133	38	2
	64	512	18340	12053	53	2
	72	512	24563	13204	62	2

The following table provides resource utilization data for the DDR4 subsystem with AXI3 and AXI4 interfaces.

Table 3-4. DDR4 Resource Utilization

Interface Type	DDR Width	AXI Width	Maximum Number of LUT4	Maximum Number of DFF	Maximum Number of μSRAM	Maximum Number of LSRAM
AXI3	16	64	11215	8867	57	6
		128	11429	9589	69	6
	32	64	16265	12883	107	2
		128	16602	13660	93	9
		256	17209	15122	115	9
	40	64	21878	14498	119	2
		128	22439	15283	105	9
		256	22911	16730	127	9
	64	512	28688	26209	67	58
	72	512	36868	28092	87	56

...........continued

Interface Type	DDR Width	AXI Width	Maximum Number of LUT4	Maximum Number of DFF	Maximum Number of µSRAM	Maximum Number of LSRAM
AXI4	16	64	11578	9120	33	18
		128	11828	9842	33	22
	32	64	16556	13130	45	21
		128	16966	13928	45	32
		256	17547	15390	45	39
	40	64	22197	14775	57	21
		128	22764	15581	57	32
		256	23262	17028	57	39
	64	512	28920	26454	69	72
	72	512	37117	28341	83	72
Native	16	64	8424	4950	19	2
		128	8424	4950	19	2
		256	8424	4950	19	2
		512	8424	4950	19	2
	32	64	12884	7758	29	2
		128	12884	7758	29	2
		256	12884	7758	29	2
		512	12884	7758	29	2
	40	64	17878	9048	38	2
		128	17878	9048	38	2
		256	17878	9048	38	2
		512	17878	9048	38	2
	64	512	—	—	—	—
	72	512	28619	14716	62	2

3.4 Functional Description

The Fabric DDR subsystem translates the AXI/native interface requests into command sequences required by SDRAM devices. The DDR controller module then issues these commands to the PHY module, which sends and receives data to and from the DDR SDRAM.

The training logic manages DDR PHY interface (DFI) 3.1 requests between the I/O lane and the DDR controller. The PLL generates the clocks required by the DDR subsystem. For more information about the PLL, see PolarFire FPGA and PolarFire SoC FPGA Clocking Resources User Guide.

3.4.1 DDR Controller

The Fabric DDR controller is a soft IP core that consists of the following blocks:

- Control and timing block—contains the main controller logic.

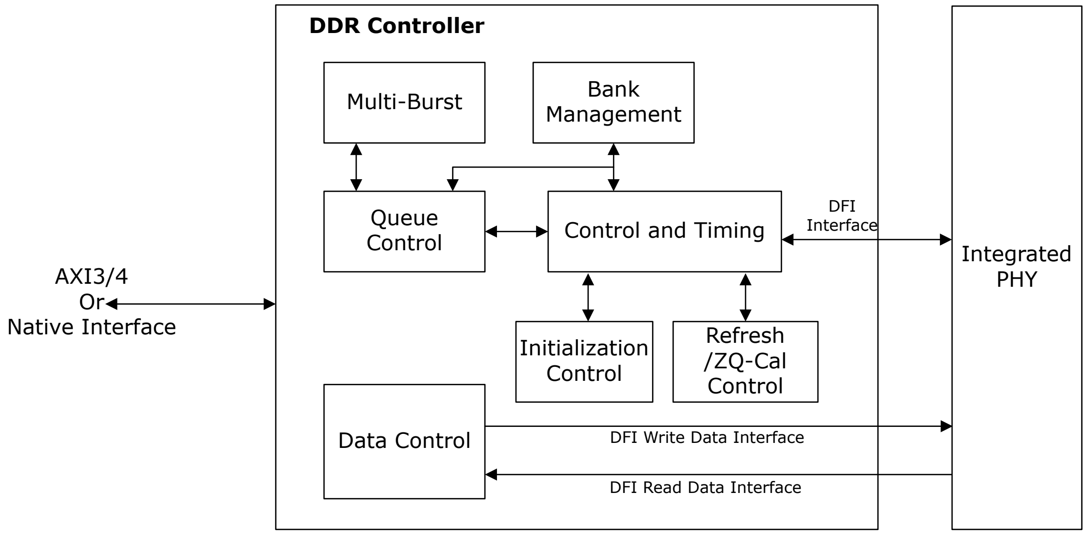
- Initialization control block—performs the initialization sequence after system reset (RESET_N) is deactivated or when dynamic reinitialization control (CTRLR_INIT) is pulsed.
- Bank management block—keeps track of the last opened row and bank to minimize command overhead.
- Refresh/ZQ calibration control block—performs automatic refresh/ZQ calibration commands to maintain data integrity.
- Queue control block—allows new requests to be accepted on every clock cycle as long as the queue is not full.
- Data control block—handles multiplexing and de-multiplexing of data flowing to and from the DDR SDRAM devices.
- Multi-burst block—allows requests longer than the programmed memory burst length. Also handles requests with starting addresses not aligned on a burst boundary, breaking the requests up as necessary to prevent data wrapping.

The queue-based implementation enables the Fabric DDR controller to optimize throughput and efficiency by looking ahead into the queue to perform precharges before the read and write commands are issued. Configure the queue depth to 3 or 4 using the Command queue depth option on DDR Configurator > Controller tab. For DDR4/LPDDR3, the default queue depth is 3.

The core also supports SECDED ECC for 40-bit and 72-bit data buses. SECDED ECC detects and corrects single-bit errors, and detects but does not correct double-bit errors. It cannot detect more than two bit errors in the data bus.

The following figure shows the functional blocks of the Fabric DDR controller.

Figure 3-4. DDR Controller Block Diagram



The following sections describe the key functions of the DDR controller.

- Periodic Functions
- Look-Ahead Operations (Precharge, and Auto-Precharge)
- Bank Management
- Multi-Burst Capability (Native-Interface Mode Only)
- Data Ordering with Non-Aligned Starting Addresses
- Auto-Precharge
- ECC

3.4.1.1 Periodic Functions

SDRAM devices require periodic functions, such as refresh and ZQ calibration, to maintain data integrity.

3.4.1.1.1 Refresh

The DDR controller automatically issues refresh commands to the attached SDRAM devices, with no user intervention required.

A refresh never interrupts an SDRAM read or write burst. However, if the subsystem determines that the refresh period has elapsed at a point concurrent with or prior to a queued read or write request, the request is held off until the refresh is performed.

3.4.1.1.2 ZQ Calibration

SDRAM devices require periodic calibration of output driver impedance and ODT values to minimize variations in voltage and temperature. This calibration is known as ZQ calibration. The DDR controller includes full support for ZQ calibration, including automatic calibration during the memory initialization sequence and periodic automatic calibrations after initialization.

ZQ calibration can be initiated either automatically or manually. The DDR Configurator provides an option to enable automatic ZQ calibration and to set automatic ZQ calibration period.

Depending on the type of SDRAM memory used in the device, four kinds of ZQ calibration are supported: ZQ calibration long (ZQCL), ZQ calibration short (ZQCS), ZQ calibration reset (ZQRESET), and ZQ calibration initialize (ZQINIT).

The following table lists the types of ZQ calibration commands issued in various SDRAM memory devices.

Table 3-5. ZQ Calibration Support Per Memory Type

Memory	Calibration Type	Time the ZQ Command is Issued
DDR3, DDR4	ZQCL	During initialization sequence.
	ZQCS	Periodically, after initialization (ZQCS is enabled by default but ZQCL can be selected for periodic ZQ calibration) periodic calibration is optional for DDR3.
LPDDR3	ZQCL	During initialization sequence.
	ZQCS	Periodically, after initialization (ZQCS is enabled by default but ZQCL can be selected for periodic ZQ calibration).
	ZQRESET	When ZQ must be reset to default output impedance.

PolarFire and PolarFire SoC devices support I/O calibration requests from the Fabric DDR controller, and calibrate only the I/Os that are used with the Fabric DDR controller.

3.4.1.2 Look-Ahead Operations (Precharge and Auto-Precharge)

The DDR controller contains logic that precharges and auto-precharges based on upcoming read or write requests pending in the queue. The DDR subsystem looks into the queue and determines whether a request requires a precharge command before a read or write command can be issued. When such a requirement is detected, the subsystem issues the necessary precharge command on the SDRAM bus to maximize the data bandwidth efficiency. The look-ahead auto-precharge logic issues an auto-precharge command (concurrent with the current read or write command) when it detects that a precharge to that bank is necessary for a read or write request currently pending in the command queue.

Notes:

1. By default, Look-Ahead operations are not enabled, they can be enabled using DDR configurator. Look-Ahead operations effect the timing closure of the DDR subsystem.
2. SDRAM commands are issued only for every 4 memory clocks, The timing diagrams shown in this document are only for general understanding and do not illustrate the same.

The following figure shows a sample timing diagram of the look-ahead precharge feature. In this sequence, two requests are issued to two different banks. The first request is issued to a bank that only requires an activate command. The second request is issued to a bank that requires a precharge and an activate. The first activate corresponding to the first request is issued at clock cycle 2. The write command associated with the first request does not occur until clock cycle 5 due to the tRCD requirement of SDRAM devices.

© 2021 Microchip Technology Inc.
and its subsidiaries

Figure 3-5. Sample Read and Write Sequence With Look Ahead Precharge (Full Wave)

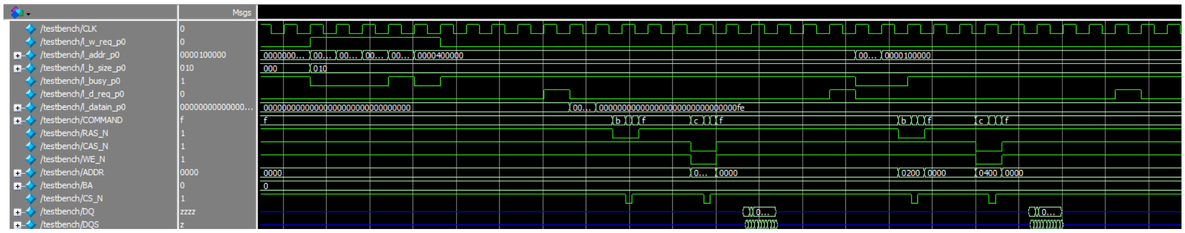
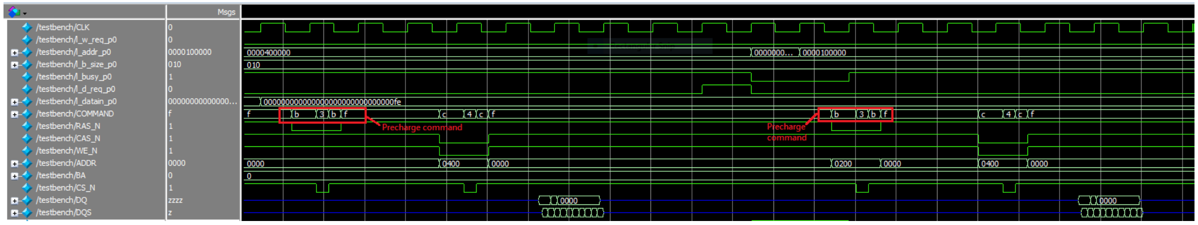


Figure 3-6. Sample Read and Write Sequence With Look Ahead Precharge (Zoomed)



The following points summarize a read-write sequence with Look-Ahead Precharge:

1. The L_W_REQ signal is asserted along with the L_ADDR signal, and the L_B_SIZE signal is set to 32.
2. The L_W_REQ signal is deasserted, indicating no other write requests are required.
3. As a result of the write request, the subsystem asserts the row address (A), bank address (BA), and chip select (CS_N) using the Look-Ahead Precharge command to open the bank at the requested row.
4. L_D_REQ transitions a clock cycle after which, the values that enable the Look-Ahead Precharge appear on the COMMAND signal as highlighted in Figure 3-6.
5. The subsystem issues the write command with a column address corresponding to the request.
6. The subsystem issues the next write command with the corresponding column address.
7. The written data begins to appear on the SDRAM bus on the DQ lines.
8. L_W_VALID is asserted, and written data appears on the native interface, and then L_W_VALID is deasserted.
9. The read operation follows the same sequence.

The following figure shows a sample timing diagram without the Look-Ahead Precharge feature.

Figure 3-7. Sample Read and Write Sequence Without Look Ahead Precharge (Full Wave)

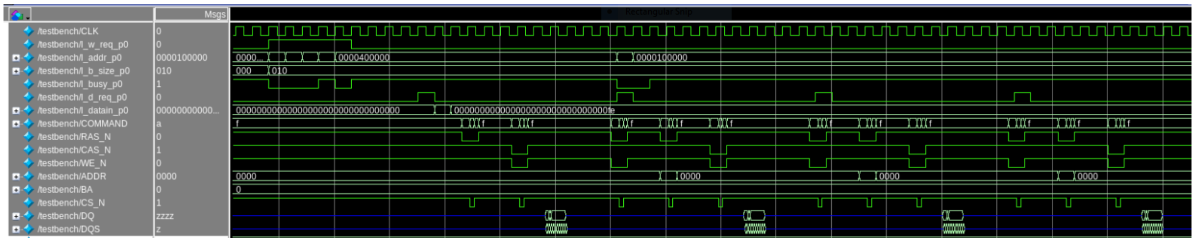
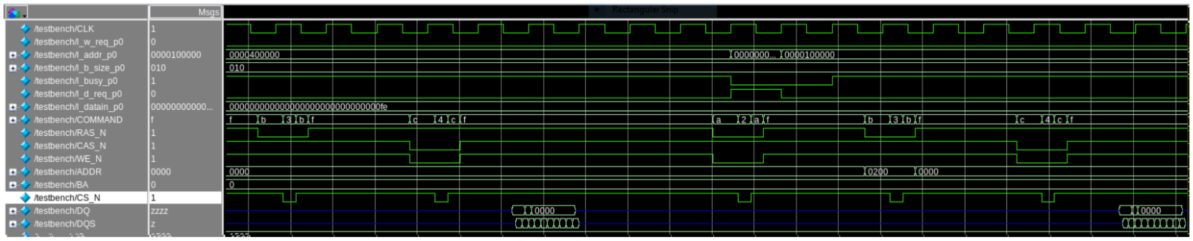


Figure 3-8. Sample Read and Write Sequence Without Look Ahead Precharge (Zoomed)



The following points summarize a read-write sequence without Look-Ahead Precharge:

1. The L_W_REQ signal is asserted along with the L_ADDR signal, and the L_B_SIZE signal is set to 32.
2. The L_W_REQ signal is deasserted, indicating no other write requests are required.
3. As a result of the write request, the subsystem asserts the row address (A), bank address (BA), and chip select (CS_N) using the look-Ahead Precharge command to open the bank at the requested row.
4. L_D_REQ transitions a clock cycle after which, the values that disable the Look-Ahead Precharge appear on the COMMAND signal as highlighted in Figure 3-8.
5. The subsystem issues the write command with a column address corresponding to the request.
6. The subsystem issues the next write command with the corresponding column address.
7. The written data begins to appear on the SDRAM bus on the DQ lines.
8. L_W_VALID is asserted, and written data appears on the native interface, and then L_W_VALID is deasserted.
9. The read operation follows the same sequence.

3.4.1.3 Bank Management

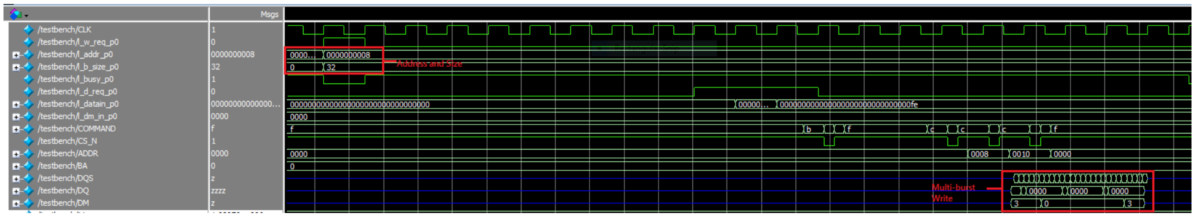
The DDR controller uses various bank management techniques to monitor the status of each SDRAM bank. Banks are only opened or closed when necessary, minimizing access delays. Up to sixteen DDR4 memory banks and eight DDR3/LPDDR3 memory banks can be managed at one time. Read and write requests are issued with minimal idle time between commands, limited only by DDR timing specifications. This timing mechanism results in minimal delay between requests, enabling up to 100% memory throughput for sequential accesses (not including refresh and ZQ calibration commands).

3.4.1.4 Multi-Burst Capability (Native-Interface Mode Only)

The DDR controller can be configured with multi-burst capability if the native-interface mode is selected in the DDR Configurator. This feature extends the allowable range of requests at the burst size (L_B_SIZE) port to accommodate requests larger than the programmed burst length. It also handles requests with starting addresses not aligned to a burst boundary, breaking them up as necessary to prevent wrapped or non-sequential data access that may otherwise occur. For more information, see 3.4.1.5 Data Ordering with Non-Aligned Starting Addresses.

The following figure shows a sample write sequence with the multi-burst capability enabled. In this sequence, a write request is issued with a starting address (L_ADDR) 0x08 and a request size (L_B_SIZE) of 32 bytes.

Figure 3-9. Sample Write Sequence Demonstrating Multi-Burst Operation



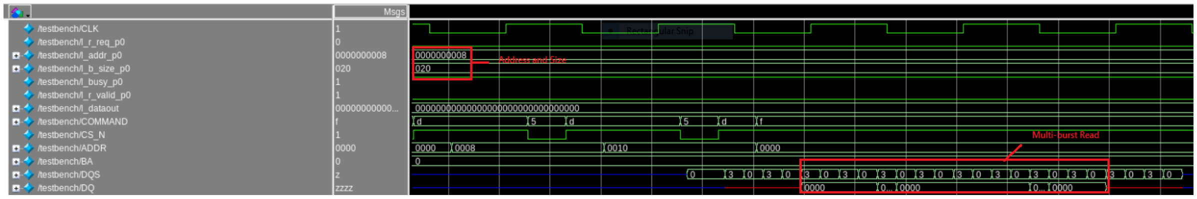
The following points summarize a write sequence with multi-burst capability:

1. The L_W_REQ signal is first asserted along with the L_ADDR signal, and the L_B_SIZE signal is set to 32.
2. The L_W_REQ signal is deasserted, indicating no other write requests are required.

3. As a result of the write request, the subsystem asserts the row address (A), bank address (BA), and chip select (CS_N) using the activate command to open the bank at the requested row.
4. The subsystem issues the write command with a column address corresponding to the request.
5. The subsystem issues the next write command with the corresponding column address.
6. The written data begins to appear on the SDRAM bus on the DQ lines.
7. L_W_VALID is asserted, and written data appears on the native interface, and then L_W_VALID is deasserted.

The following figure shows a sample read sequence with the multi-burst capability enabled. In this example, a read request is issued with a starting address (L_ADDR) 0x08 and a request size (L_B_SIZE) of 32 bytes.

Figure 3-10. Sample Read Sequence Demonstrating Multi-Burst Operation



The following points summarize a read sequence with multi-burst capability:

1. The L_R_REQ signal is first asserted along with the L_ADDR signal, and the L_B_SIZE signal is set to 32.
2. The L_R_REQ signal is deasserted, indicating no other read requests are required.
3. As a result of the read request, the subsystem asserts the row address (A), bank address (BA), and chip select (CS_N) using the activate command to open the bank at the requested row.
4. The subsystem issues the read command with a column address corresponding to the request.
5. The subsystem issues the next read command with the corresponding column address.
6. The read data begins to appear on the SDRAM bus on the DQ lines.
7. L_R_VALID is asserted, and read data appears on the native interface, and then L_R_VALID is deasserted.

3.4.1.5 Data Ordering with Non-Aligned Starting Addresses

For write and read requests with a starting address aligned to the programmed burst length, data flow to and from SDRAM devices is always sequential. Each successive data phase within a burst corresponds to a higher address value. If the starting address is not aligned to the programmed burst length, the data will not be sequential. At a minimum, a wrap occurs, because only the locations within a burst range can be accessed with the exact ordering depending on the lower three bits of the address. The multi-burst capability automatically deals with these issues, off-loading the burden from the user logic.

3.4.1.6 Auto-Precharge

Read and write commands can be issued to SDRAM devices with or without auto-precharge. If the auto-precharge option is selected, the SDRAM device automatically closes (precharges) the bank being read from or written to at the end of the transaction. Any subsequent access to this bank does not require a precharge command from the DDR subsystem for the transaction to be closed.

The L_AUTO_PCH signal is used to issue read and write requests with auto-precharge to the SDRAM. For write requests, the L_AUTO_PCH signal is asserted along with the L_W_REQ signal, and for read requests, the L_AUTO_PCH signal is asserted along with the L_R_REQ signal. It is acceptable to permanently tie the L_AUTO_PCH signal input high or low, as shown in Figure 3-9 and Figure 3-10.

The auto-precharge option is useful in situations where the requested read or write addresses are random. With random address sequences, banks of the exact row required by a subsequent request may not be open. If auto-precharge is not used for the previous access to a bank, subsequent transactions to that bank require the bank to be closed (precharged), causing a delay in the transaction execution. If auto-precharge is used with the previous access to the bank, the bank is closed and ready to be opened to the desired row.

3.4.1.7 ECC

When ECC is enabled, the DDR controller computes an 8-bit ECC for every 64-bit data to support SECDED. A write operation computes and stores ECC along with the data, and a read operation reads and checks the data against the stored ECC. Therefore, when ECC is enabled, single or double-bit errors might be received when reading uninitialized memory locations. To prevent this, all memory locations must be written to before being read.

For a non 64-bit write operation, the DDR controller performs a read-modify-write (RMW) operation as follows:

- Reads the data from DDR memory (4 or 8 DDR memory bursts)
- Modifies the read data with the user data
- Computes the ECC and writes to DDR memory

The DDR subsystem uses status signals to indicate a single-bit error (ECC_ERROR_1BIT) or a double-bit error (ECC_ERROR_2BIT) along with the error position (ECC_ERROR_POS) to the fabric.

3.4.2 Clocking Structure

Both the DDR subsystem and the DDR PHY-only solution require a dedicated PLL to generate the clocks, which are then distributed throughout the subsystem using HS_IO_CLK routes, dedicated pads, and fabric clock routing. This PLL generates aligned clocks for all sub-blocks for smooth operation and synchronous communication with the user logic.

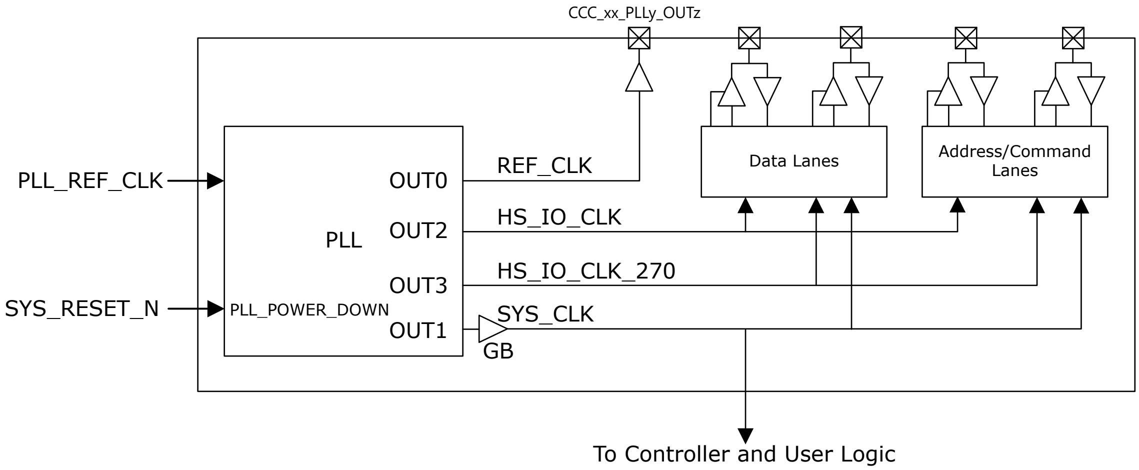
The PLL generates the following required clocks:

- REF_CLK— This clock is routed to the PHY for clocking the DDR memory device.
- HS_IO_CLK— This clock routed to I/O lanes and the training logic.
- HS_IO_CLK_270— HS_IO_CLK phase shifted by 270. This clock is also routed to I/O lanes and the training logic.
- SYS_CLK— This clock is routed to the DDR controller, training logic, and user logic in the fabric.

The HS_IO_CLK and REF_CLK clocks are generated with the same frequency and phase. The REF_CLK to SYS_CLK ratio is 4:1.

The following illustration shows the clocking structure of the DDR subsystem.

Figure 3-11. DDR Clocking Structure



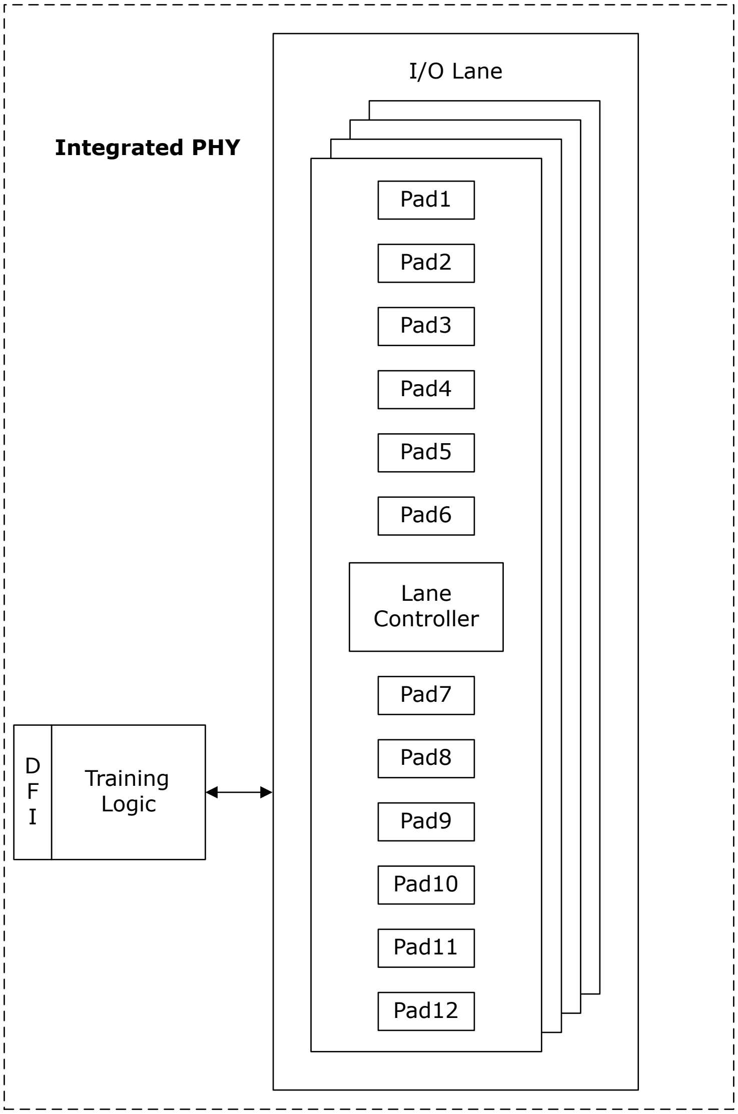
Note: The dedicated clock output pad, CCC_xx_PLLy_OUTz, is selected depending on the DDR subsystem placement constraint, where xx = SE/NE/NW/SW, y = 0/1, and z = 0/1.

3.4.3 Integrated PHY

The integrated PHY, which consists of the I/O lane and the training logic, provides a physical interface to DDR3, DDR4, and LPDDR3 SDRAM devices. It receives commands from the DDR controller and generates the DDR memory signals required to access the external DDR memory. The training logic in the PHY manages DFI 3.1

training requests between the I/O lane and the DDR controller. For more information about the DDR PHY-only integration, see 3.7 DDR PHY-Only Solution Integration. For more information about Octal DDR PHY-Only Solution, see 3.8 Octal DDR PHY-Only Solution.

Figure 3-12. DDR PHY



3.4.3.1 I/O Lane

An I/O lane contains 12 I/Os, a lane controller, I/O gearing logic, and a set of high-speed, low-skew clock resources. The I/O gearing logic in a lane enables easy data transfer between the high-speed I/O pad and the lower-speed FPGA core. The logic is used to either gear up the data rate from the FPGA fabric to the memory device or gear down the data rate from the memory device to the FPGA fabric. For information about non-memory interface usage, see PolarFire FPGA and PolarFire SoC FPGA User I/O User Guide.

The lane controller contains the logic for managing the read and write DQS signals and provides the lane clocks. The following table lists the number of lanes required for the DDR memory interface.

Table 3-6. DDR Memory Interface Data Lanes

Memory Type	DQ Width	Number of I/O Lanes
Address/Command Lanes	Data Lanes	

...........continued

Memory Type	DQ Width	Number of I/O Lanes	
DDR3/4	x16	3	2
DDR3/4	x32	3	4
DDR3/4	x40	3	5
DDR3/4	x64	3	8
DDR3/4	x72	3	9
LPDDR3	x16	2	2
LPDDR3	x32	2	4

Each data lane uses one I/O lane with 12 I/O pads each—two of the I/O pads are used for the DQS, eight for the DQ bits, and one for the data mask (DM). One I/O pad is left as spare.

Note: I/O usage depends on the configuration of the DDR memory, not all I/Os in the address/command lanes are used. Some I/Os in these lanes can be reused as normal I/Os. For data lanes, all 12 I/Os in the lane are used.

3.4.3.2 Training Logic

The training logic manages the DFI 3.1 training requests between the integrated PHY and DDR controller modules, and performs the following operations.

- Clock training
- Write leveling
- Read leveling
- I/O calibration

The CTRLR_READY signal is asserted to indicate the completion of initialization and training. This signal can be monitored from the fabric.

Clock Training

- HS_IO_CLK to SYS_CLK training—aligns HS_IO_CLK to the DDR controller clock. When it is aligned, the data can be transferred to or from the user logic.
- CMD/ADDR to REF_CLK training—aligns the rising edge of REF_CLK to the center of the address and command buses of the DDR memory. When the rising edge is aligned, DDR commands can be written to the SDRAM.

Write Leveling

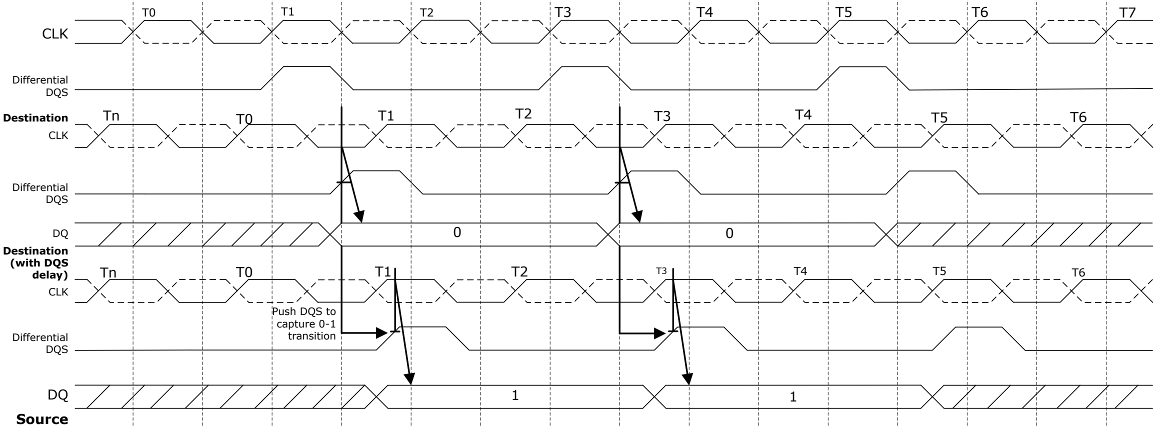
Write leveling is a training mode used during DRAM initialization. The write leveling process identifies the delay when the write DQS rising edge aligns with the rising edge of REF_CLK. By identifying this delay, the system can accurately align the write DQS within REF_CLK. When it is aligned, data can be written to the SDRAM. This alignment is at the memory chip, not at the I/O Lane.

The fly-by topology introduced in DDR3 improves signal integrity, but it creates variable skew between the CLK and DQS in every DRAM device. Long traces on DIMM can create skew greater than one MEM_CLK cycle.

However, the JEDEC® DDR TDQSS specification requires DQS and CLK to align ± 0.25 MEM_CLK. Write leveling compensates for this skew. It is the first in a sequence of training steps. The memory controller has a programmable DQS delay. The DRAM provides feedback to indicate if the DQS leads or lags the CLK.

As illustrated in the following figure, the training logic sends out widely spaced DQS pulses. The DRAM uses CLK as D input and DQS as CLK to the flip-flop. The Q output of the flip-flop is fed back to the prime DQ (the DQ bit on which feedback is provided). The prime DQ could be different for different vendors. The objective is to detect a 0-to-1 transition on the CLK with a DQS rising edge. This is done by moving the DQS in small steps until the sampled CLK changes from 0-to-1. When the transition is detected, DQ is aligned with DQS.

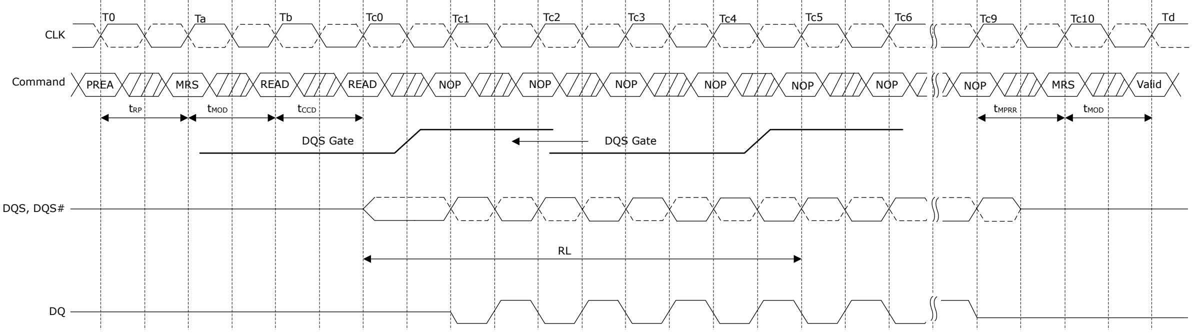
Figure 3-13. DQS Delay



Read Leveling

- DQS gate training—The training logic has a DQS gate signal for capturing the correct DQS strobe during read operations and for removing the DQS glitches. The purpose of gate training is to determine the optimum delay that can be applied to the DQS gate for it to function properly.
- As illustrated in the following figure, when the bus is idle, DQ/DQS are at VDD/2. The output of the DQS receiver is undefined. Read DQS is internally used to clock FIFO read data. The write pointer for read data must correspond to every incoming piece of data. This means incoming DQS must be qualified with DQS GATE cover pulse, which goes high during pre-amble and goes low during post-amble. For board design guidelines for DQ and DQS pins, see 5. PolarFire Board Design Recommendations and 6. PolarFire SoC Board Design Recommendations.
- The memory controller generates the read commands required for DQS gate training. The training logic places the DQS gate in the middle of the burst. It detects a 0-to-1 transition of DQS at the rising edge of the DQS gate. After the transition is detected, the DQS gate is advanced by 90° (of MEM_CLK) to place the rising edge in the middle of the DQS high pulse. Then, the rising edge is walked back one MEM_CLK at a time. The rising edge detects a value of 1 until it hits pre-amble.
- When the rising edge of the DQS gate detects a value of 0 for the first time during this walk back process, it is in the pre-amble area of the DQS burst. Only the rising edge of the DQS gate needs to be trained. The training logic runs internal timers to place the falling edge of the gate in the post-amble area.

Figure 3-14. DQS Gate Training



- Align read DQ bits—aligns the read DQ edges for a lane to increase the size of the data window.
- Align read DQS to DQ—aligns the DQS strobe to the center of the data window to increase its size. DQ/DQS centering is a two-step process. First, all DQ bits are aligned to maximize the data valid window. Then, the DQS strobe is placed at the center. Because of variables such as package routing, parasitics (controller and device), and board layout, the performance of all DQ bits is not identical. Each DQ bit in a byte lane are impacted differently. Consequently, there is always a best case and worst case DQ bit within a byte lane. Placement of DQS at the center of data valid window is done using known patterns. When the read data matches with the

data written to a location, the DQS is moved both to the left and right until it fails. After both the left and right margins are identified, the mean of the left and right margins [(left + right)/2] is computed, and the DQS is placed at the newly computed mean value. The following figures illustrate DQ/DQS before and after centering.

Figure 3-15. DQ/DQS Before Centering

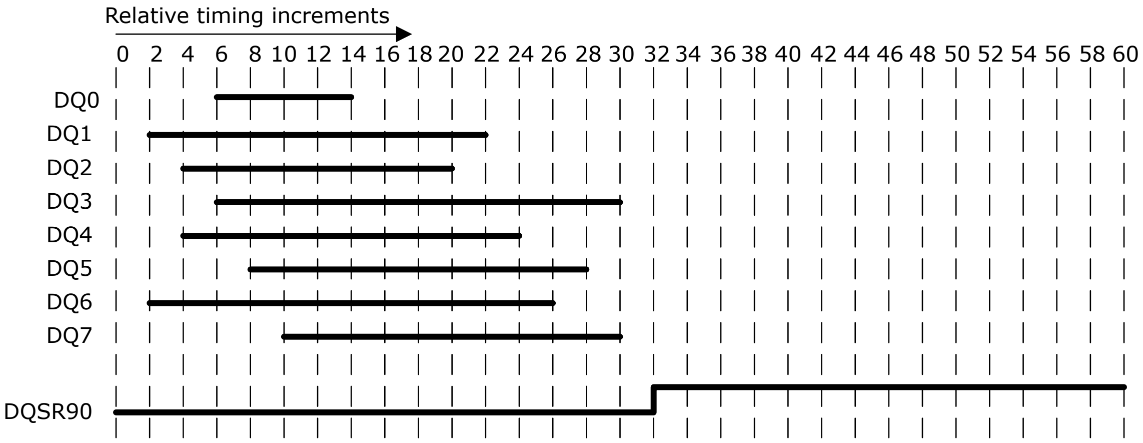
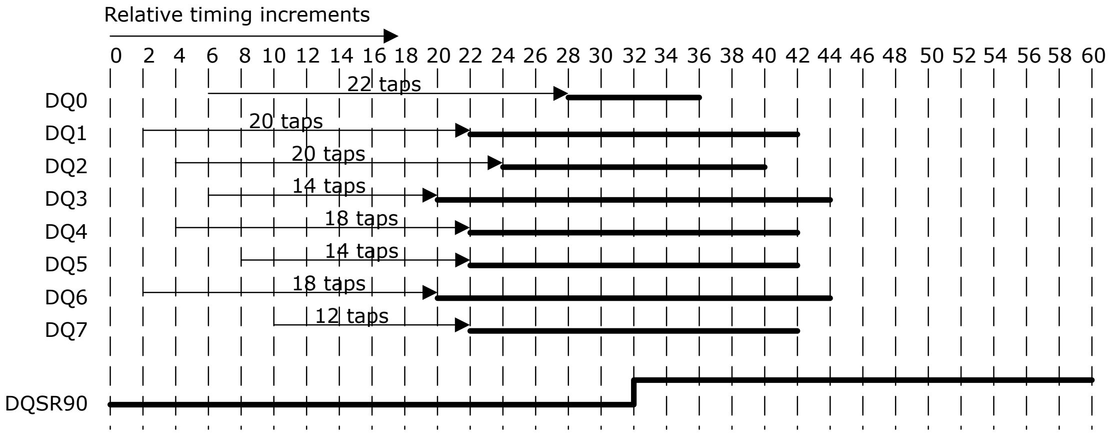


Figure 3-16. DQ/DQS After Centering



I/O Calibration

The Fabric DDR subsystem performs DDR memory I/O calibration after device power-up. For more information about I/O calibration, see PolarFire FPGA and PolarFire SoC FPGA Device Power-Up and Resets User Guide.

3.4.4 Initialization Sequence

After the asynchronous system reset (SYS_RESET_N) and PLL_LOCK are deasserted, the DDR controller performs a JEDEC compliant initialization sequence for memory devices. Each DDR SDRAM device has a series of mode registers (MR) that are accessed using the mode-register-set (MRS) commands. These mode registers set various SDRAM behaviors, such as burst length, CAS latency, ODT value, additive latency, and so on. These registers can be configured using the DDR Configurator.

The CTRLR_READY signal is asserted to indicate that initialization is completed. This signal can be monitored from the fabric.

The following sections explain the automatic initialization sequence for DDR3, DDR4, and LPDDR3 memory devices.

3.4.4.1 DDR3

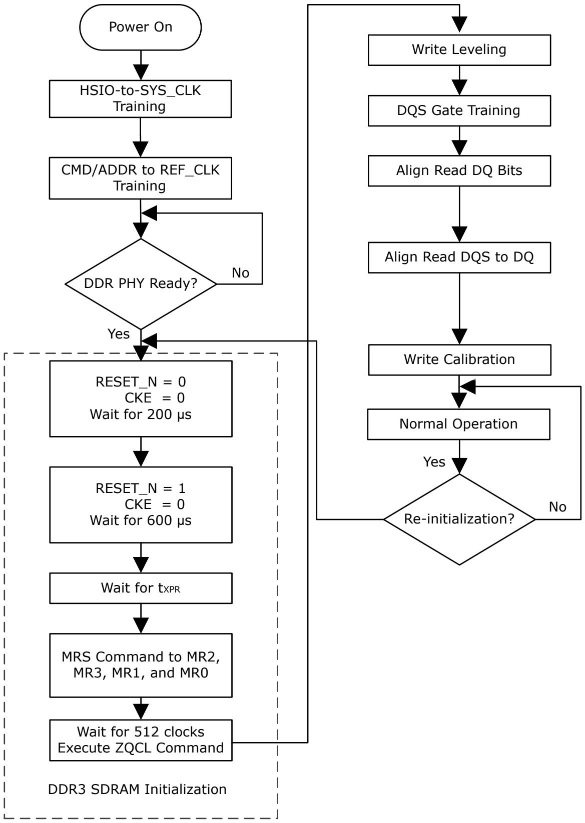
Automatic initialization of DDR3 memory involves the following steps:

1. Training logic performs HS_IO_CLK-to-SYS_CLK training.
2. Training logic performs CMD/ADDR to REF_CLK training.
3. DDR controller waits for the PHY to be ready. At this point, it is assumed that the PHY outputs stable clocks and signal levels.
4. Controller asserts RESET_N and CKE low.
5. Controller deasserts RESET_N after 200 µs, and then deasserts CKE after an additional 600 µs.
6. Controller waits for tXPR.
7. MRS command is sent to the following mode registers (in order): MR2, MR3, MR1, and MR0.
8. Controller waits for 512 clock cycles. During this wait time, the ZQCL (long) command is sent to each rank independently to calibrate the RTT and RON values.
9. Training logic performs write leveling.
10. Training logic performs DQS gate training.
11. Training logic aligns read DQ bits.
12. Training logic aligns read DQS to DQ.
13. Write calibration is performed.
14. Normal operation begins.

The controller reinitializes the SDRAM memory when a low-to-high transition is detected on the reinitialization control signal (CTRLR_INIT).

The following figure shows the DDR3 automatic initialization flow.

Figure 3-17. Automatic Initialization Flow for DDR3 Memory



3.4.4.2 DDR4

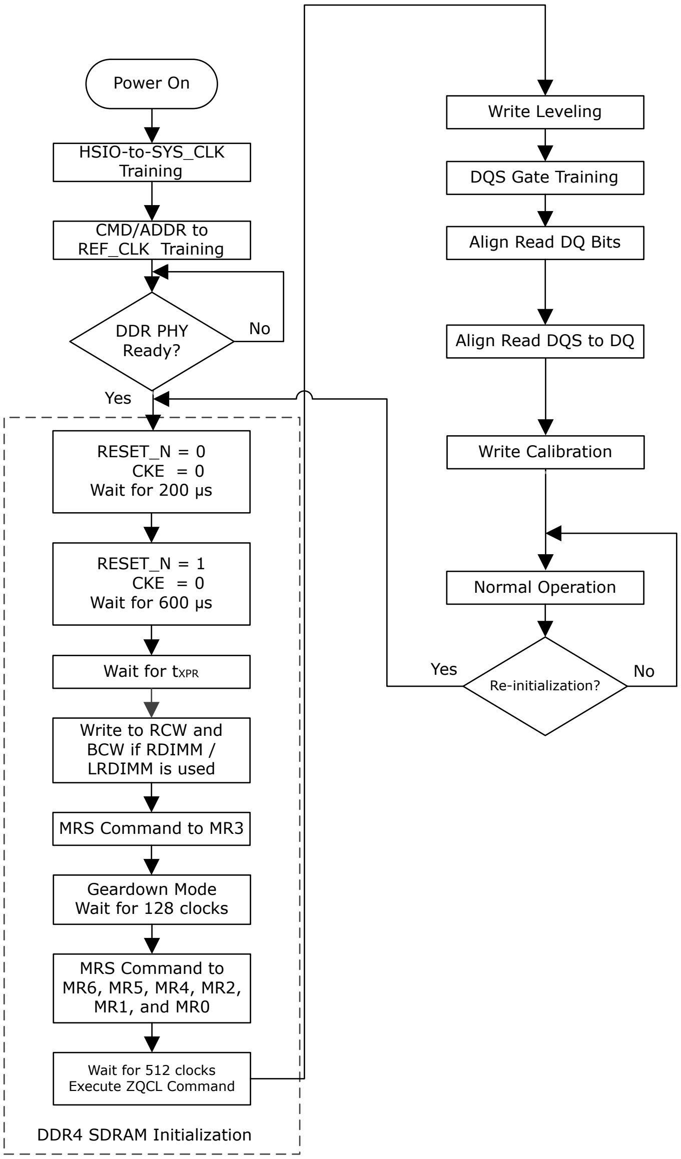
Automatic initialization of DDR4 memory involves the following steps:

1. Training logic performs HS_IO_CLK-to-SYS_CLK training.
2. Training logic performs CMD/ADDR to REF_CLK training.
3. DDR controller waits for the PHY to be ready. At this point, it is assumed that the PHY outputs stable clocks and signal levels at this point.
4. Controller asserts RESET_N and CKE low.
5. Controller deasserts RESET_N after 200 µs, and then deasserts CKE after an additional 600 µs.
6. Controller waits for tXPR.
7. Controller writes to RCW and BCW if RDIMM/LRDIMM is used.
8. MRS command is sent to the following mode registers (in order): MR3, MR6, MR5, MR4, MR2, MR1, and MR0.
9. Controller waits for 128 clock cycles (geardown mode). Controller waits for an additional 512 clock cycles for the ZQCL (long) command to be sent to each rank independently to calibrate the RTT and RON values.
10. Training logic performs write leveling.
11. Training logic performs DQS gate training.
12. Training logic aligns read DQ bits.
13. Training logic aligns read DQS to DQ.
14. Write calibration is performed.
15. Normal operation begins.

The controller reinitializes the SDRAM memory when a low-to-high transition is detected on the reinitialization control signal (CTRLR_INIT).

The following figure shows the DDR4 automatic initialization flow.

Figure 3-18. Automatic Initialization Flow for DDR4 Memory



3.4.4.3 LPDDR3

Automatic initialization of LPDDR3 memory involves the following steps:

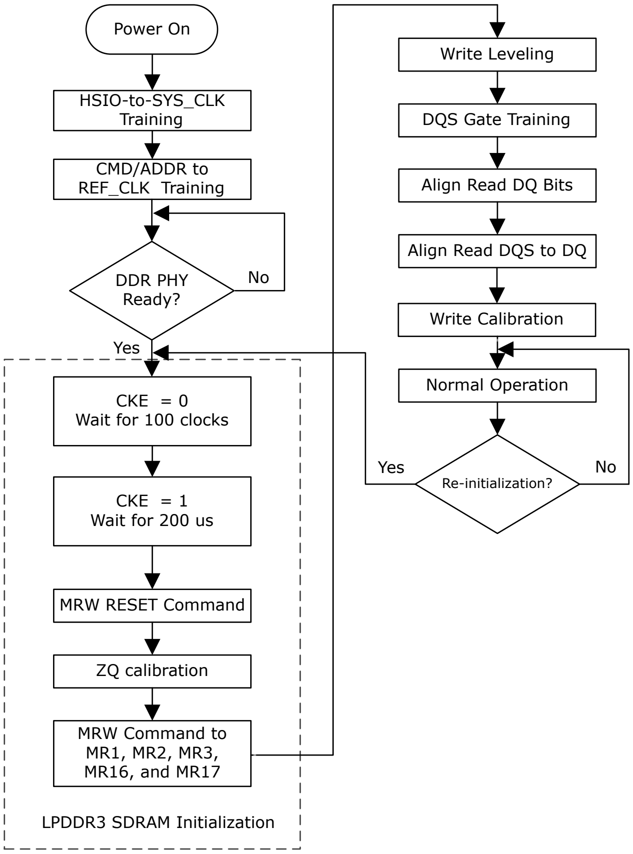
1. Training logic performs HS_IO_CLK-to-SYS_CLK training.
2. Training logic performs CMD/ADDR to REF_CLK training.
3. DDR controller waits for the PHY to be ready. At this point, it is assumed that the PHY is outputs stable clocks and signal levels.

4. CKE is held low for tINIT1. The LPDDR3 specification is 100 ns (minimum), but the controller waits for 100 clock cycles.
5. CKE is held high for tINIT3. The LPDDR3 specification is 200 µs.
6. MRW RESET command is issued.
7. Controller waits for tINIT5.
8. Controller sequentially issues the initialization calibration command to each active rank. This enables a configuration where all ranks share a ZQ resistor.
9. Controller waits for tZQINIT after each calibration command is performed.
10. MRS command sent to the following mode registers (in order): MR1, MR2, MR3, MR16, and MR17.
11. Training logic performs write leveling.
12. Training logic performs DQS gate training.
13. Training logic aligns read DQ bits.
14. Training logic aligns read DQS to DQ.
15. Write calibration is performed.
16. Normal operation begins.

The controller reinitializes the SDRAM memory when a low-to-high transition is detected on the reinitialization control signal (CTRLR_INIT).

The following figure shows the LPDDR3 automatic initialization flow.

Figure 3-19. Automatic Initialization Flow for LPDDR3 Memory



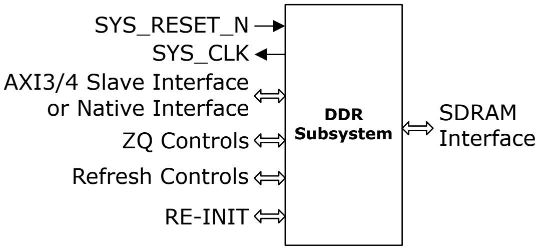
3.5 DDR Subsystem Ports

The Fabric DDR subsystem ports are categorized into the following groups:

- AXI3/4 slave interface signals—required when the DDR subsystem is configured in AXI3/4 mode. An AXI master is implemented in the FPGA fabric to perform DDR memory transactions by interfacing with the DDR subsystem.
- Native interface signals—required when the DDR subsystem is configured in native interface mode. To perform DDR memory transactions, custom logic must be implemented in the FPGA fabric to interface with the DDR subsystem. Logic connected to the native interface must operate synchronously with the SYS_CLK.
- Refresh controls—required for user-initiated refresh control.
- ZQ controls—required for user-initiated ZQ calibration control.
- RE-INIT controls—required for re-initializing DDR memories.
- SDRAM interface signals—required for connecting to the DDR SDRAM.

The following figure shows the DDR subsystem ports.

Figure 3-20. Fabric DDR Subsystem Ports



In addition to ports shown in the preceding figure, the Fabric DDR subsystem also has some generic ports, see the following table.

Table 3-7. Generic Signals

Signal Name	Direction	Description
SYS_CLK	Output	Clock for user logic generated by the embedded PLL. All native interface signals are synchronous to this clock. Always on the global clock network.
SYS_RESET_N	Input	Active-low asynchronous system reset. SYS_RESET_N must be generated by ANDing the DEVICE_INIT_DONE and BANK_X_CALIB_STATUS signals of the PF_INIT_MONITOR IP[1]. BANK_X refers to the BANK where Fabric DDR subsystem is placed.
PLL_REF_CLK	Input	Reference clock to the PLL.
PLL_LOCK	Output	Lock signal generated by the PLL to indicate that the PLL is locked on to the PLL_REF_CLK signal.
CTRLR_READY	Output	Signal that is deasserted when in reset or after CTRLR_INIT is asserted, then asserted after initialization and training sequences (if applicable) are completed.

...........continued

Signal Name	Direction	Description
stat_ca_parity_error	Output	For DDR4: This signal is asserted when an assertion is detected on the ALERT_n pin. Each occurrence of a write CRC error or CA parity error causes the assertion of the DDR4 device ALERT_n pin. The value is cleared on read.

Note:

1. For more information about the PF_INIT_MONITOR IP, see PolarFire FPGA and PolarFire SoC FPGA Device Power-Up and Resets User Guide.

The DDR Controller complies with the AXI3/4 protocol. The AXI3/4 slave interface provides the following features:

- Supports all AXI allowed burst sizes, types, and lengths.
- Handles all AXI wrapping conditions.
- Supports 64-bit, 128-bit, 256-bit, and 512-bit AXI interfaces.

For more information about the AXI3/4 protocol, see the *AMBA AXI and ACE Protocol Specification*.

The following table lists the AXI3/4 slave interface signals. All AXI interface signals are active-high and are synchronous to SYS_CLK.

Table 3-8. AXI3/4 Slave Interface Signals

Signal Name	Direction	Description
Write Address Channel		
AXI0_AWID[X-1:0][2]	Input	Write address ID.
AXI0_AWADDR[31:0]	Input	Write address.
AXI0_AWLEN[M:0]	Input	Burst length. M = 7 for AXI4 IF and M = 3 for AXI3 IF.
AXI0_AWSIZE[2:0]	Input	Burst size.
AXI0_AWBURST[1:0]	Input	Burst type.
AXI0_AWLOCK[1:0]	Input	Lock type.Not supported.
AXI0_AWCACHE[3:0]	Input	Memory type. Not supported.
AXI0_AWPROT[2:0]	Input	Protection type. Not supported.
AXI0_AWVALID	Input	Write address valid.
AXI0_AWREADY	Output	Write address ready.
Write Data Channel		
AXI0_WID[X-1:0]	Input	Write ID tag. Supported only in AXI3.
AXI0_WDATA[N-1:0][1]	Input	Write data.
AXI0_WSTRB[N/8 - 1:0][1]	Input	Write strobes.
AXI0_WLAST	Input	Write last.
AXI0_WVALID	Input	Write valid.
AXI0_WREADY	Output	Write ready.
Write Response Channel		
AXI0_BID[X-1:0]	Output	Response ID tag.

...........continued		
Signal Name	**Direction**	**Description**
AXI0_BRESP[1:0]	Output	Write response.
AXI0_BVALID	Output	Write response valid.
AXI0_BREADY	Input	Response ready.
Read Address Channel		
AXI0_ARID[X-1:0][2]	Input	Read address ID.
AXI0_ARADDR[31:0]	Input	Read address.
AXI0_ARLEN[M:0]	Input	Burst length. M = 7 for AXI4 IF and M = 3 for AXI3 IF.
AXI0_ARSIZE[2:0]	Input	Burst size.
AXI0_ARBURST[1:0]	Input	Burst type.
AXI0_ARLOCK[1:0]	Input	Lock type. Not supported.
AXI0_ARCACHE[3:0]	Input	Memory type. Not supported.
AXI0_ARPROT[2:0]	Input	Protection type. Not supported.
AXI0_ARVALID	Input	Read address valid.
AXI0_ARREADY	Output	Read address ready.
Read Data Channel		
AXI0_RID[X-1:0][2]	Output	Read ID tag.
AXI0_RDATA[N-1:0]	Output	Read data.
AXI0_RRESP[1:0]	Output	Read response.
AXI0_RLAST	Output	Read last.
AXI0_RVALID	Output	Read valid.
AXI0_RREADY	Input	Read ready.

Notes:

1. N can be configured as 64,128, 256, or 512 using the Fabric DDR subsystem configurator.
2. ‘X’ can be configured between 1 to 8 using the Fabric DDR Configurator.

The following table lists the native interface signals. All control signals are active-high and are synchronous to SYS_CLK.

Table 3-9. Native Interface Signals

Signal Name	Direction	Description
L_ADDR[38:0]	Input	Native interface address sizes: DDR4 = 39 bits, DDR3 = 36 bits, and LPDDR3 = 36 bits
L_B_SIZE[10:0]	Input	Native interface burst length in terms of bytes. It must be in multiples of the native interface bus width.
L_R_REQ	Input	Native interface read request.
L_W_REQ	Input	Native interface write request.

...........continued

Signal Name	Direction	Description
L_AUTO_PCH	Input	When asserted along with L_R_REQ or L_W_REQ, causes the command to be issued as read with auto-precharge and write with auto-precharge, respectively.
L_BUSY	Output	Specifies that the subsystem is busy and is not accepting new requests. A command is accepted on any clock cycle where L_R_REQ or L_W_REQ is set, and L_BUSY is low. If L_BUSY is high when L_R_REQ or L_W_REQ is set, the request may be kept asserted (along with the desired L_ADDR, L_B_SIZE and L_AUTO_PCH values) until L_BUSY goes low.
L_D_REQ	Output	Requests data on the native interface write data bus (L_DATAIN) during a write transaction. Asserts one clock cycle prior to when data is required.
L_D_REQ_LAST_P0	Output	Requests the last data on the native interface write data bus This signal is used along with L_D_REQ
L_R_VALID	Output	Data-valid indication for data on the native interface read data bus (L_DATAOUT).
L_R_VALID_LAST_P0	Output	Data-valid indication for the last data on the native interface read data bus This signal is used along with L_R_VALID
L_DATAIN[N:0]	Input	Input data bus. This data bus is eight times the width of the SDRAM device data bus. Memory width (bits): 16, 32, 64. Input data bus (bits): 128, 256, 512.
L_DATAOUT[N:0]	Output	Output data bus. This data bus is twice the width of the SDRAM device data bus. Memory width (bits): 16, 32, 64. Output data bus (bits): 128, 256, 512.
L_DM_IN[N:0]	Input	Individual byte masks during data write. Memory width (bits): 16, 32, 64. Data mask bus (bits): 16, 32, 64.

The following table lists the refresh control signals. To expose these signals, select the **Enable User Refresh Controls** check box from DDR3/LPDDR3 configurator > Controller tab > Efficiency.

Table 3-10. Refresh Control Signals

Signal Name	Direction	Description
L_REF_REQ	Input	User-initiated refresh control. Causes a refresh command to be issued at the next opportunity. This signal is only used when manual control of refresh is desired. The DDR Configurator provides an option (Enable User Refresh Controls) to expose the user-initiated refresh control (L_REF_REQ) signal. tREFI parameter in the DDR Configurator specifies the period between refreshes.
L_REF_ACK	Output	Refresh acknowledge. Asserted for one clock cycle when a refresh command is being issued. Typically used to determine when a refresh was last issued in user-controlled refresh. Can be ignored when the subsystem is automatically generating refreshes.

The following table lists the ZQ calibration control signals. To expose these signals, enable User ZQ Calibration Controls in the DDR Configurator Memory Timing tab.

Table 3-11. ZQ Calibration Control Signals

Signal Name	Direction	Description
L_ZQ_CAL_REQ	Input	Causes user-initiated ZQCS or ZQCL requests to be issued at the next opportunity. The ZQCS or ZQCL are issued to the ranks corresponding to the asserted bits. ZQ calibration can be initiated either automatically or manually. The DDR Configurator provides an option to enable automatic ZQ calibration and to set the automatic ZQ calibration period.
L_ZQ_CAL_ACK	Output	ZQ calibration acknowledge. Asserted for a single clock cycle when a ZQ calibration command is issued to memory devices. Used for ZQ calibrations initiated by asserting the L_ZQ_CAL_REQ signal.

The following table describes the reinitialization control signal. To expose this signal, enable RE-INIT Controls in the DDR Configurator Controller tab.

Table 3-12. Reinitialization Control Signal

Signal Name	Direction	Description
CTRLR_INIT	Input	Causes the subsystem to reissue the initialization sequence to the SDRAM. The initialization begins when a low-to-high transition is detected on the input. The controller always issues the initialization sequence (including the startup delay) after a reset, regardless of this signal’s state. If run-time reinitialization is not required, the signal can be tied low.

The following table lists the ECC status signals.

Table 3-13. ECC Status Signals

Signal Name	Direction	Description
ECC_ERROR_1BIT	Output	Active when a 1-bit error is detected on the data being presented on the AXI RDATA/L_DATAOUT port.
ECC_ERROR_2BIT	Output	Active when a 2-bit error is detected on the data being presented on the AXI RDATA/L_DATAOUT port.
ECC_ERROR_POS[6:0]	Output	Indicates bit position of the error in a 64-bit data output for a 1-bit error.

The following table lists the SDRAM interface signals.

Table 3-14. SDRAM Interface Signals

Signal Name	Direction	Description
CK	Output	Differential clock pair forwarded to SDRAM.
CK_N	Output	Differential clock pair forwarded to SDRAM.
RESET_N	Output	SDRAM reset. Supported only for DDR3 and DDR4.
A[15:0]	Output	Address bus. Sampled during the active, precharge, read, and write commands. Also provides the mode register value during MRS commands. Bus width for LPDDR3 is 10 bits, DDR3 is 16 bits, and DDR4 is 14 bits.

...........continued

Signal Name	Direction	Description
BA[2:0]	Output	Bank address. Sampled during active, precharge, read, and write commands to determine which bank the command is to be applied to. Supported only for DDR3 and DDR4. For DDR4, bus width is 2 bits. For DDR3, bus width is 3 bits.
BG[1:0]	Output	DDR bank group address for DDR4 only.
CS_N	Output	SDRAM chip select.
CKE	Output	SDRAM clock enable. Held low during initialization to ensure SDRAM DQ and DQS outputs are in the hi-Z state.
RAS_N	Output	SDRAM row address strobe command. Supported only for DDR3 and DDR4.
CAS_N	Output	SDRAM column access strobe command. Supported only for DDR3 and DDR4.
WE_N	Output	SDRAM write-enable command. Supported only for DDR3 and DDR4.
ODT	Output	On-die termination control. ODT is asserted during reads and writes according to the ODT activation settings in the DDR Configurator.
PAR	Output	Command and address parity output. Supported only for DDR4.
ALERT_N	Input	Alert signaling command/address parity or write CRC error. Supported only for DDR4.
DQ	Bidirectional	SDRAM data bus. Supports 16-bit, 32-bit, 39-bit, 64-bit, and 72-bit DDR SDRAM data buses.
DM/DM_N	Output	Write data mask. DM for DDR3/LPDDR3 and DM_N for DDR4.
DQS	Bidirectional	Strobes data into the SDRAM devices during writes and into the DDR subsystem during reads.
DQS_N	Bidirectional	Complimentary DQS.
SHIELD	Output	Pads must be connected to ground. They are placed between the data lanes for improving signal integrity.

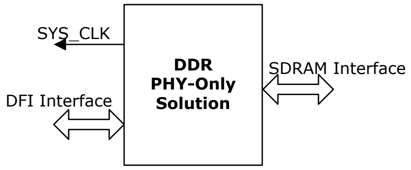
3.5.1 DDR PHY-Only Solution Ports

The DDR PHY-only solution ports are divided into the following groups:

- DFI interface—signals required when the DDR subsystem is configured for DDR-PHY mode. This interface allows a third-party DDR controller hosted in the fabric to connect directly to the PHY.
- SDRAM interface—signals connected to the DDR SDRAM.

The following illustration shows the DDR PHY-only solution ports.

Figure 3-21. DDR PHY-Only Solution Ports



3.5.2 DFI Interface

The DDR memory controller interface solution leverages the DDR PHY interface (DFI 3.1) for connections between the controller and the PHY. The control signal, write data, read data update, status, and training interfaces are listed in the following tables. Use the PolarFire FPGA DDR3 PHY configurator to expose the DFI interface. For more information about the interface, see to the DFI 3.1 Specification available on the DFI Group DDR PHY website.

Table 3-15. DFI Control Signal Interface (synchronous to SYS_CLK)

Signal Name	Direction	Description
DFI_ADDRESS_P0[15:0]	Input	DFI Address bus
DFI_ADDRESS_P1[15:0]	Input	DFI Address bus
DFI_ADDRESS_P2[15:0]	Input	DFI Address bus
DFI_ADDRESS_P3[15:0]	Input	DFI Address bus
DFI_BANK_P0[2:0]	Input	DFI bank bus (only for DDR3 and DDR4)
DFI_BANK_P1[2:0]	Input	DFI bank bus (only for DDR3 and DDR4)
DFI_BANK_P2[2:0]	Input	DFI bank bus (only for DDR3 and DDR4)
DFI_BANK_P3[2:0]	Input	DFI bank bus (only for DDR3 and DDR4)
DFI_CAS_N_P0	Input	DFI column address strobe (only for DDR3 and DDR4)
DFI_CAS_N_P1	Input	DFI column address strobe (only for DDR3 and DDR4)
DFI_CAS_N_P2	Input	DFI column address strobe (only for DDR3 and DDR4)
DFI_CAS_N_P3	Input	DFI column address strobe (only for DDR3 and DDR4)
DFI_CKE_P0[0]	Input	DFI clock enable
DFI_CKE_P1[0]	Input	DFI clock enable
DFI_CKE_P2[0]	Input	DFI clock enable
DFI_CKE_P3[0]	Input	DFI clock enable
DFI_CS_N_P0[0]	Input	DFI chip select
DFI_CS_N_P1[0]	Input	DFI chip select
DFI_CS_N_P2[0]	Input	DFI chip select
DFI_CS_N_P3[0]	Input	DFI chip select
DFI_ODT_P0[0]	Input	DFI on-die termination control
DFI_ODT_P1[0]	Input	DFI on-die termination control
DFI_ODT_P2[0]	Input	DFI on-die termination control
DFI_ODT_P3[0]	Input	DFI on-die termination control
DFI_RAS_N_P0	Input	DFI row address strobe (only for DDR3 and DDR4)
DFI_RAS_N_P1	Input	DFI row address strobe (only for DDR3 and DDR4)
DFI_RAS_N_P2	Input	DFI row address strobe (only for DDR3 and DDR4)
DFI_RAS_N_P3	Input	DFI row address strobe (only for DDR3 and DDR4)
DFI_RESET_N_P0	Input	DFI reset
DFI_RESET_N_P1	Input	DFI reset
DFI_RESET_N_P2	Input	DFI reset

...........continued

Signal Name	Direction	Description
DFI_RESET_N_P3	Input	DFI reset
DFI_WE_N_P0	Input	DFI write enable (only for DDR3 and DDR4)
DFI_WE_N_P1	Input	DFI write enable (only for DDR3 and DDR4)
DFI_WE_N_P2	Input	DFI write enable (only for DDR3 and DDR4)
DFI_WE_N_P3	Input	DFI write enable (only for DDR3 and DDR4)

Table 3-16. DFI Write Data Interface

Signal Name	Direction	Description
DFI_WRDATA_CS_N_P0[0]	Input	DFI write data chip select
DFI_WRDATA_CS_N_P1[0]	Input	DFI write data chip select
DFI_WRDATA_CS_N_P2[0]	Input	DFI write data chip select
DFI_WRDATA_CS_N_P3[0]	Input	DFI write data chip select
DFI_WRDATA_EN_P0[63:0][1]	Input	DFI write data and data mask enable
DFI_WRDATA_EN_P1[63:0]	Input	DFI write data and data mask enable
DFI_WRDATA_EN_P2[63:0]	Input	DFI write data and data mask enable
DFI_WRDATA_EN_P3[63:0]	Input	DFI write data and data mask enable
DFI_WRDATA_MASK_P0[15:0][1]	Input	DFI write data byte mask
DFI_WRDATA_MASK_P1[15:0][1]	Input	DFI write data byte mask
DFI_WRDATA_MASK_P2[15:0][1]	Input	DFI write data byte mask
DFI_WRDATA_MASK_P3[15:0][1]	Input	DFI write data byte mask
DFI_WRDATA_P0[127:0][1]	Input	These signals transfer write Data from memory controller to PHY
DFI_WRDATA_P1[127:0][1]	Input	These signals transfer write Data from memory controller to PHY
DFI_WRDATA_P2[127:0][1]	Input	These signals transfer write Data from memory controller to PHY
DFI_WRDATA_P3[127:0][1]	Input	These signals transfer write Data from memory controller to PHY

Note: Depends on DQ width (see Table 3-22 and Table 3-23).

Table 3-17. DFI Read Data Interface

Signal Name	Direction	Description
DFI_RDDATA_CS_N_P0[0]	Input	DFI read data chip select
DFI_RDDATA_CS_N_P1[0]	Input	DFI read data chip select
DFI_RDDATA_CS_N_P2[0]	Input	DFI read data chip select
DFI_RDDATA_CS_N_P3[0]	Input	DFI read data chip select
DFI_RDDATA_EN_P0[63:0][1]	Input	DFI read data enable

...........continued		
Signal Name	**Direction**	**Description**
DFI_RDDATA_EN_P1[63:0][1]	Input	DFI read data enable
DFI_RDDATA_EN_P2[63:0][1]	Input	DFI read data enable
DFI_RDDATA_EN_P3[63:0][1]	Input	DFI read data enable
DFI_RDDATA_VALID_W0[7:0][1]	Output	DFI read data valid
DFI_RDDATA_VALID_W1[7:0][1]	Output	DFI read data valid
DFI_RDDATA_VALID_W2[7:0][1]	Output	DFI read data valid
DFI_RDDATA_VALID_W3[7:0][1]	Output	DFI read data valid
DFI_RDDATA_W0[127:0][1]	Output	DFI read data
DFI_RDDATA_W1[127:0][1]	Output	DFI read data
DFI_RDDATA_W2[127:0][1]	Output	DFI read data
DFI_RDDATA_W3[127:0][1]	Output	DFI read data

Note:

1. Depends on DQ width (see Table 3-22 and Table 3-23).

Table 3-18. DFI Write Calibration Interface

Signal Name	Direction	Description
CAL_L_BUSY	Input	Write Calibration Busy. Indicates this interface is not accepting new commands. A command is accepted when CAL_L_R_REQ or CAL_L_W_REQ is set and CAL_L_BUSY is low.
CAL_L_DATAOUT[511:0][1]	Input	Write calibration data output from the controller Depends on DQ width.
CAL_L_D_REQ	Input	Write calibration data request from the controller.
CAL_L_R_VALID	Input	Write calibration read valid from the controller.
CAL_L_DATAIN[511:0][1]	Output	Write calibration data input to the controller Depends on DQ width.
CAL_L_DM_IN[63:0][1]	Output	Write calibration data mask input to the controller Depends on DQ width.
CAL_L_R_REQ	Output	Write calibration read request to the controller.
CAL_L_W_REQ	Output	Write calibration write request to the controller.
CAL_SELECT	Output	Write Calibration Select to the controller.
CAL_INIT_ACK[2]	Input	Calibration initialization bus available handshake from DDR controller.
CAL_INIT_CS[1:0][2]	Output	Calibration initialization bus chip select.
CAL_INIT_MR_ADDR[7:0][2]	Output	Calibration initialization bus mode register write address.
CAL_INIT_MR_DATA[17:0][2]	Output	Calibration initialization bus mode register write data.
CAL_INIT_MR_MASK[17:0][2]	Output	Calibration initialization bus mode register write mask.
CAL_INIT_MR_W_REQ[2]	Output	Calibration initialization bus mode register write request.

Notes:

1. Depends on DQ width (see Table 3-22 and Table 3-23).
2. These ports are only for DDR4.
3. The DRAM interface is same as Table 3-14 for DDR PHY only solution.

Table 3-19. DFI Training Interface

Signal Name	Direction	Description
DFI_RDLVL_CS_N[0]	Input	Chip select for read data eye training
DFI_RDLVL_EN	Input	PHY read data eye training enable
DFI_RDLVL_GATE_EN	Input	PHY read gate training enable
DFI_WRLVL_CS_N[0]	Input	Chip select for write leveling
DFI_WRLVL_EN	Input	PHY write leveling logic enable
DFI_WRLVL_STROBE	Input	Initiates capture of write level response
DFI_RDLVL_RESP[7:0]	Output	Indicates data eye or gate training complete
DFI_WRLVL_RESP[7:0]	Output	Indicates PHY has completed write leveling

Table 3-20. DFI Status Interface

Signal Name	Direction	Description
DFI_INIT_START	Input	DFI setup stabilization
DFI_INIT_COMPLETE	Output	PHY initialization complete

Table 3-21. DFI Clock and Miscellaneous

Signal Name	Direction	Description
PLL_LOCK	Output	PLL lock indicator
PLL_REF_CLK	Input	PLL reference clock
SYS_CLK	Output	PLL system clock output to controller
SYS_RESET_N	Input	Global reset input
SYNC_SYS_RST_N	Output	Synchronized reset output
CTRLR_READY_IN	Input	Controller ready status from controller to PHY
CTRLR_READY_OUT	Output	Controller ready status from PHY to controller
DFI_TRAINING_COMPLETE	Output	PHY output to controller to indicate training complete

3.6 Functional Timing Diagrams

This section describes the write and read operations of the Fabric DDR subsystem in native interface and AXI4 interface modes.

3.6.1 Native Interface Transactions

This section describes the read/write rules and sequences in the native interface. For information about native interface signals, see Table 3-9.

Note: During the simulation users can find difference between Core configuration timings (JEDEC) and simulation results. This is due to the internal architecture of the controller (T4), which includes a 4-stage queue so that commands to memory are issued at every 4 clock cycles.

3.6.1.1 SDRAM Writes

SDRAM writes are requested in the native interface by asserting the L_W_REQ signal high, and by driving the starting address and the burst size on the L_ADDR and L_B_SIZE signals respectively. A write can be requested with auto-precharge by asserting the L_AUTO_PCH signal along with the L_W_REQ signal.

The following are the rules for write requests in the native interface:

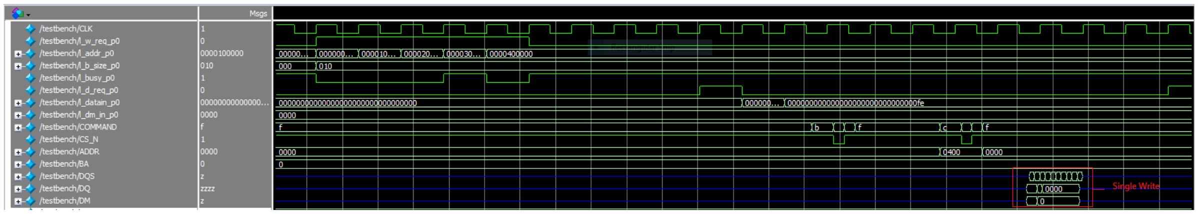
- A write is accepted by the subsystem on any clock cycle where the L_W_REQ signal is asserted while the L_BUSY signal is deasserted. After L_BUSY is deasserted by the controller, L_W_REQ may remain asserted to request a follow-on write transaction. L_W_REQ may remain asserted over any number of clock periods to generate any number of cascaded write requests.
- The values of the L_ADDRL_B_SIZE, and L_AUTO_PCH signals are captured when the L_BUSY signal is low. It is acceptable to permanently tie the L_B_SIZE and L_AUTO_PCH signals to fixed values.
- The L_W_REQ signal cannot be asserted while the L_R_REQ signal is asserted.
- The data request (L_D_REQ) signal is asserted one clock prior to the data signal at the L_DATAIN bus and the subsystem masks the information at the L_DM_IN bus.
- The timing relationship between an initial L_W_REQ, L_D_REQ, and L_BUSY or between L_BUSY assertions and de-assertions as a result of multiple cascaded writes varies depending on the status of the banks being accessed, the configuration port settings, and the refresh and initialization status.

Note: The user logic must not rely on any fixed timing relationship between the L_W_REQ, L_D_REQ, and L_BUSY signals.

3.6.1.1.1 Sample Write Sequence

The following figure shows a sample write sequence consisting of single write request.

Figure 3-22. Example Single Write Sequence



The following points summarize a write sequence:

1. The L_W_REQ signal is first asserted along with the L_ADDR signal, and the L_B_SIZE signal is set to 32.
2. If the L_BUSY signal is not asserted in clock cycle 1 (indicating that the request is accepted by the subsystem), the L_W_REQ signal remains asserted. The L_ADDR and L_B_SIZE signals are updated for the next write request.
3. The L_W_REQ signal is deasserted, indicating that no other write requests are required.
4. As a result of the write request, the subsystem asserts the row address (A), bank address (BA), and chip select (CS_N) using the activate command to open the bank at the requested row.
5. As a result of the write request, the subsystem issues a write command with the appropriate column address.In response to the next write request, the subsystem issues another write command with the corresponding column address.
6. The subsystem requests data in the native interface by asserting the L_D_REQ signal.

In this sequence, all the writes are to the same bank and row. If the next write request targets a different bank, the DDR subsystem may have to issue a precharge and/or activate commands prior to issuing write commands.

3.6.1.2 SDRAM Reads

SDRAM reads are requested in the native interface by asserting the L_R_REQ signal, and by driving the starting address and burst size on the L_ADDR and L_B_SIZE signals respectively. The L_AUTO_PCH signal may also be asserted along with the L_R_REQ signal to issue the read as a read with auto-precharge. A read can be requested with auto-precharge by asserting the L_AUTO_PCH signal along with the L_R_REQ signal.

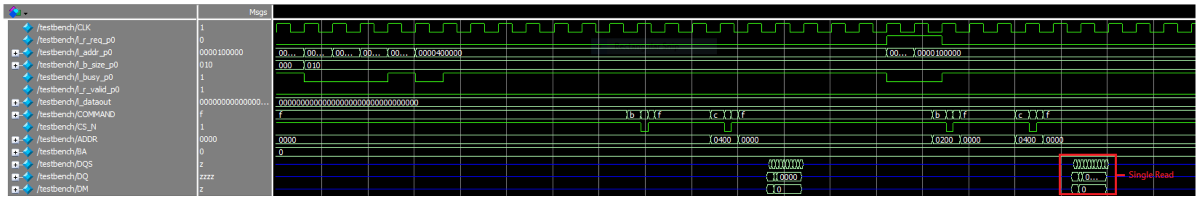
The following are the rules for read requests at in the native interface:

- A read is accepted by the subsystem on any clock cycle where the L_R_REQ signal is asserted while the L_BUSY signal is deasserted. After L_BUSY is deasserted by the subsystem, the L_R_REQ signal may remain asserted to request a follow-on read transaction. The L_R_REQ signal may remain asserted over any number of clock periods to generate any number of cascaded read requests.
- The values of the L_ADDRL_B_SIZE, and L_AUTO_PCH signals are captured when the L_BUSY signal is low. The L_B_SIZE and L_AUTO_PCH signals can be tied to fixed values, if desired.
- The L_R_REQ signal cannot be asserted while the L_W_REQ signal is asserted.
- The read data valid (L_R_VALID) signal is asserted when valid data is available at the L_DATAOUT bus.
- The timing relationship between an initial L_R_REQ, L_R_VALID, and L_BUSY or between L_BUSY assertions and de-assertions as a result of multiple cascaded reads varies depending on the status of the banks being accessed, configuration port settings, refresh status, and initialization status.
- The user logic must not rely on any fixed timing relationship between the L_R_REQ, L_R_VALID, and L_BUSY signals.

3.6.1.2.1 Sample Read Sequence

The following figure shows a sample read sequence consisting of single read request.

Figure 3-23. Example Sample Read Sequence



The following points summarize a read sequence:

1. The L_R_REQ signal is first asserted along with the L_ADDR signal, and the L_B_SIZE signal is set to 32.
2. In the previous clock cycle, the l_busy signal was not asserted, indicating that the subsystem accepted the first request on that cycle. The L_R_REQ signal remains asserted. The L_ADDR and L_B_SIZE signals are updated for the next read request.
3. The L_R_REQ signal is deasserted, indicating no other read requests are required.
4. As a result of the read request, the subsystem asserts the row address (A), bank address (BA), and chip select (CS_N) using the activate command to open the bank at the requested row.
5. The subsystem issues a read command with a column address corresponding to the request at the respective clock cycle.
6. The subsystem issues the next read command with the corresponding column address.
7. Read data begins to appear on the SDRAM bus on the DQ lines.
8. L_R_VALID is asserted, and read data appears in the native interface.
9. L_R_VALID is deasserted.

In this sequence, all read requests are to the same bank and row. If the next read was to different banks or rows, the core may have issued precharge and/or activate commands prior to issuing read commands.

3.6.2 AXI Transactions

The following timing diagrams illustrate the operation of the DDR controller with the AXI interface. The Micron DDR3 16-bit memory model is used to perform the read and write transactions.

Figure 3-24. AXI Single Write Transaction

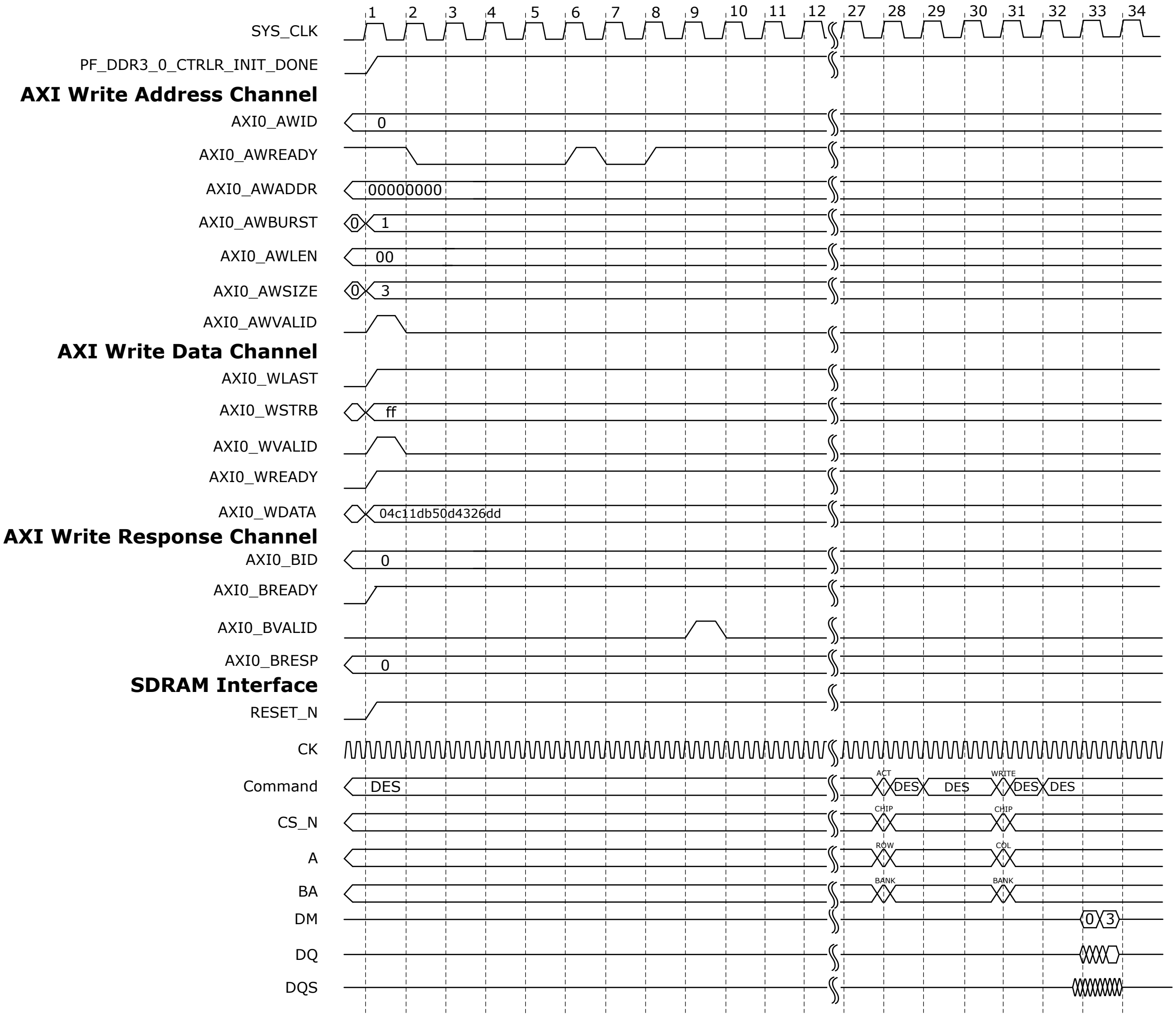


Figure 3-25. AXI Burst Write Transaction (INCR-2)

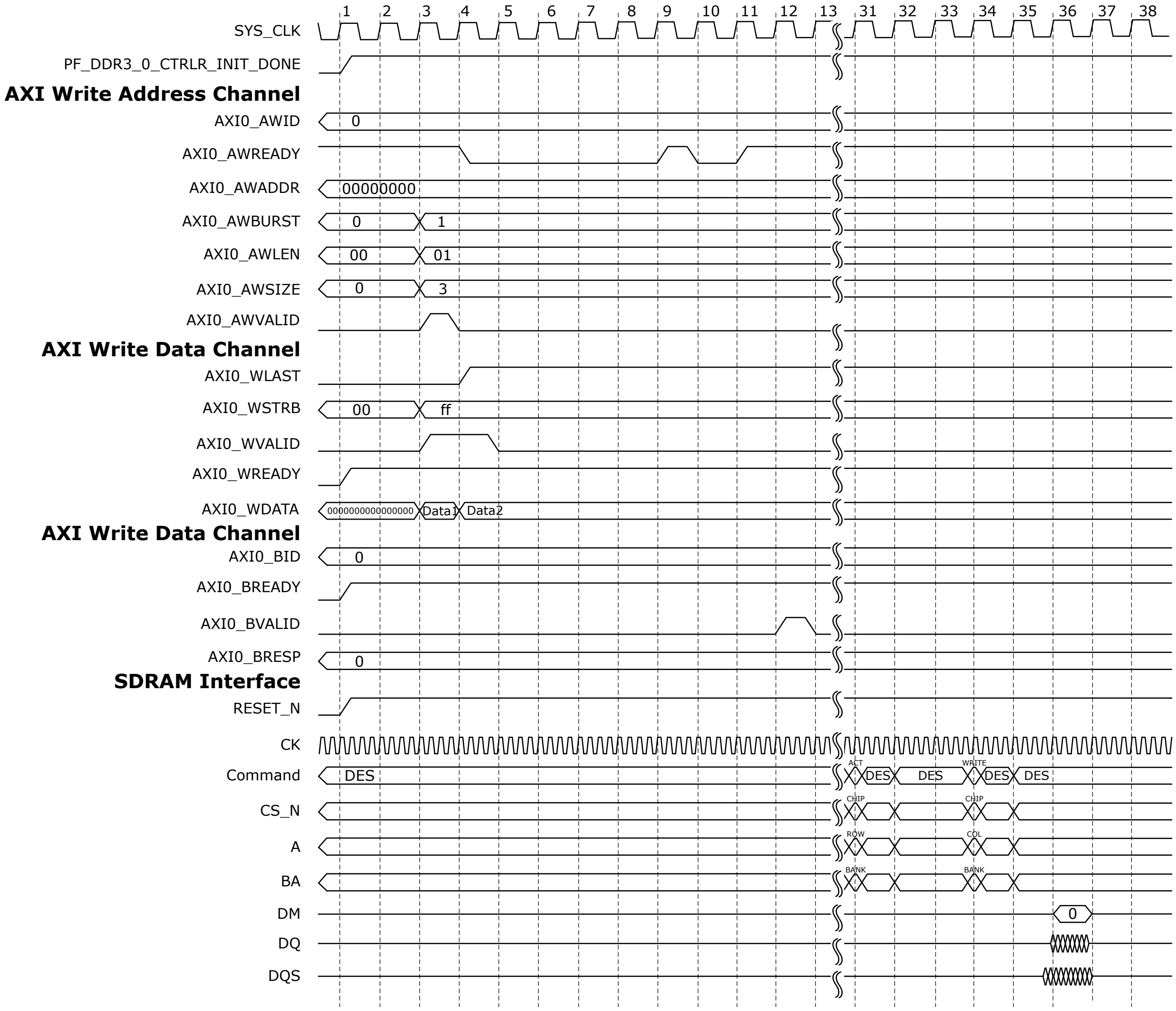


Figure 3-26. AXI Single Read Transaction

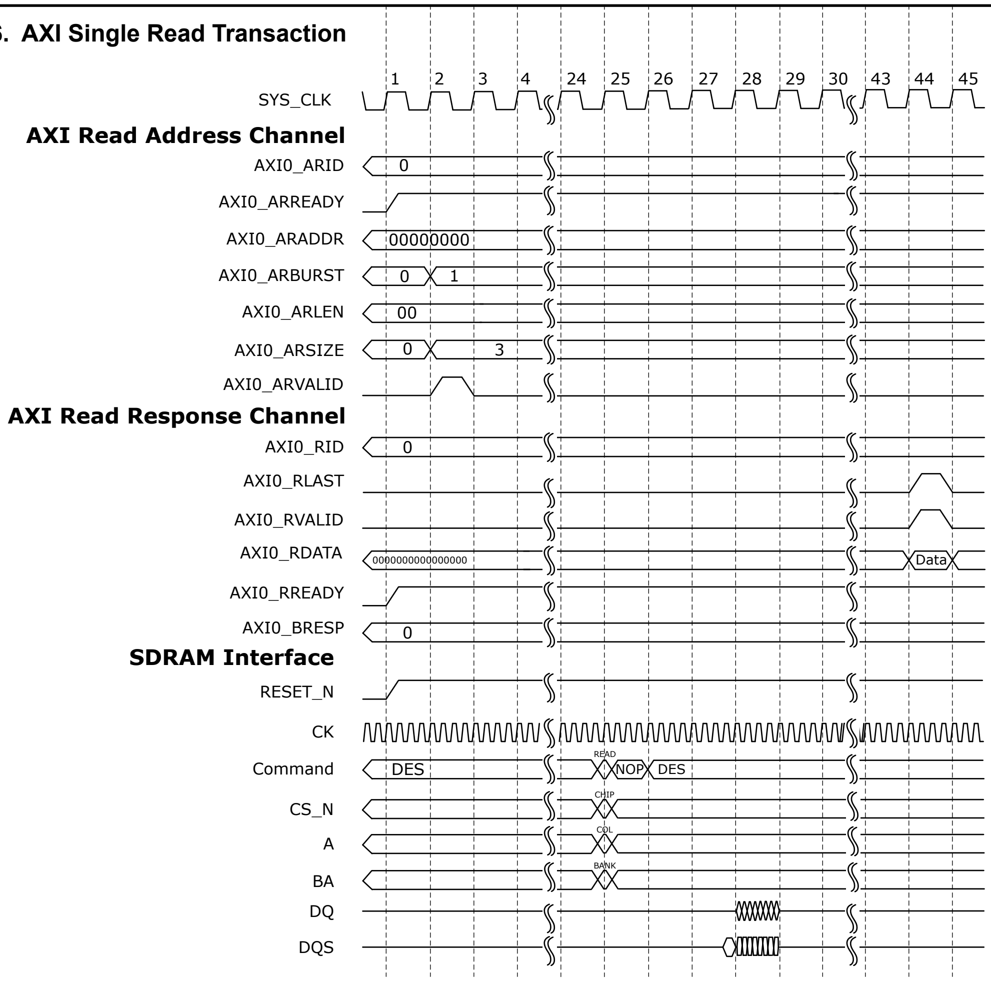
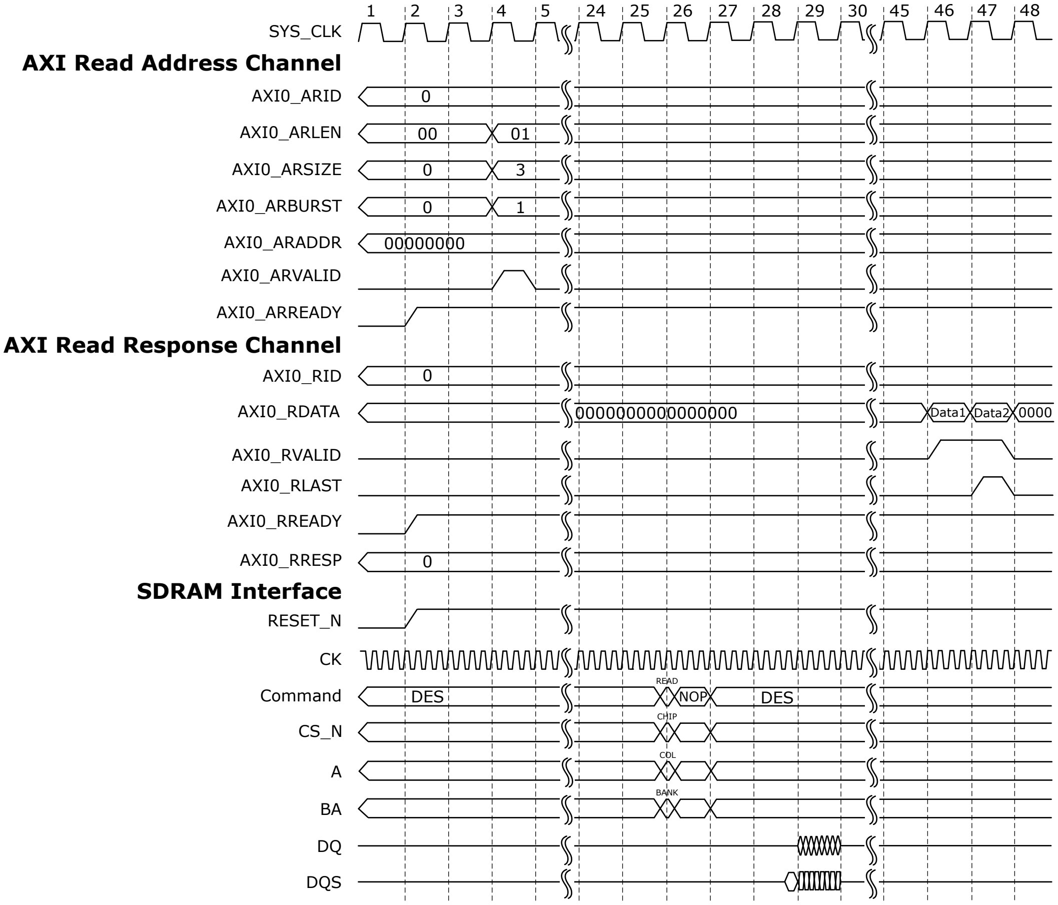


Figure 3-27. AXI Burst Read Transaction (INCR-2)



3.7 DDR PHY-Only Solution Integration

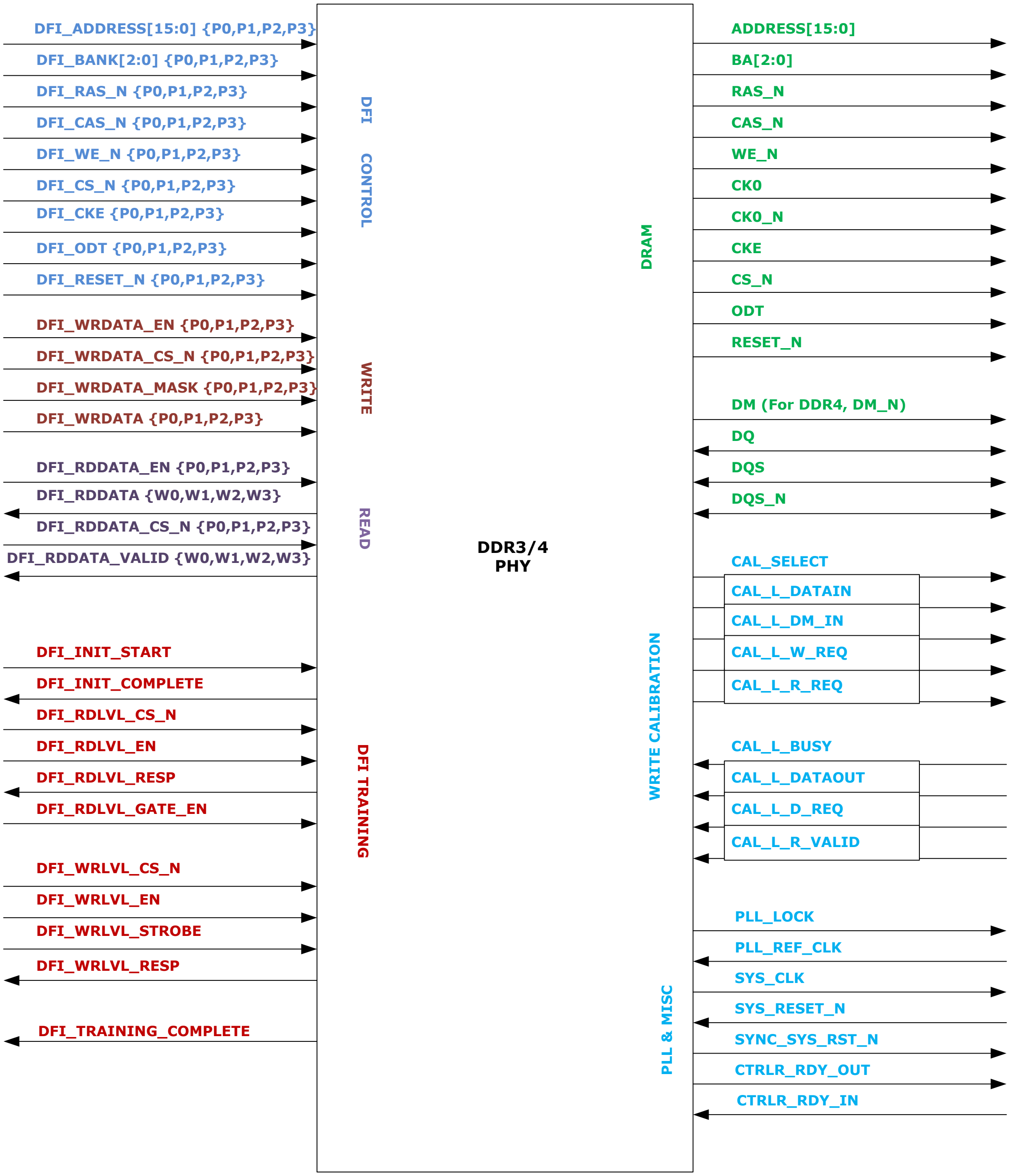
This section describes how to integrate third party DDR memory controllers with PolarFire DDR PHY. The PHY top-level behavior on both DFI and DRAM interfaces are detailed in this section.

The PolarFire DDR PHY-only solution is DFI 3.1 complaint and supports DDR3, DDR4, and LPDDR3.

The PolarFire DDR PHY-only solution includes a PLL (CCC – Clock Conditioning circuit) and an integrated PHY as shown in Figure 3-2. The integrated PHY includes I/O lanes and training logic, I/O lanes are used for command/data.

The following figure shows the top-level I/O of the DDR PHY. The signals are grouped and color coded for easy visualization.

Figure 3-28. DDR3 PHY-Only Top-Level I/O



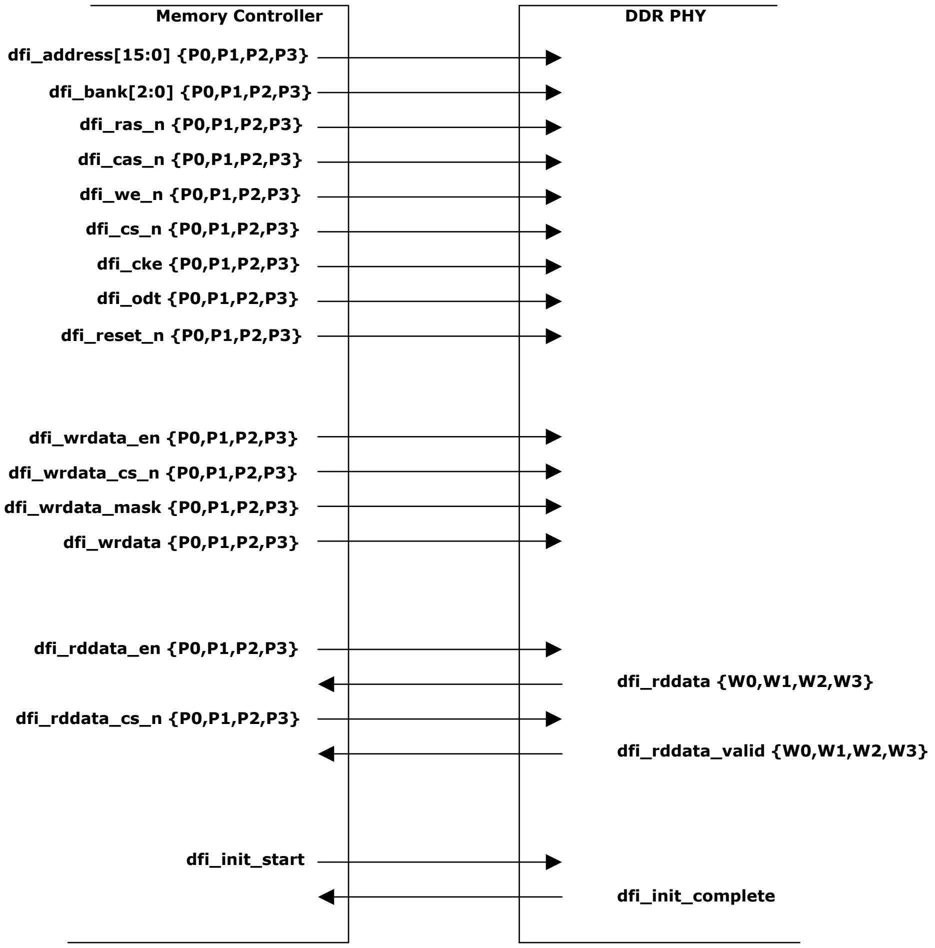
As shown in the previous figure, DFI signals **DFI_INIT_START** and **DFI_INIT_COMPLETE** are part of the status interface according to the DFI 3.1 specification. These signals are shown as part of the training interface because the PHY performs the clock training from SYS_CLK to HSIO_CLK and from HSIO_CLK to REF_CLK during the initialization.

3.7.1 Controller and PHY Integration

This section describes the PHY top-level pin list and how they interconnect to other modules in a DDR subsystem. The DDR3 PHY is compliant with DFI 3.1 and can be easily integrated with controllers that comply with DFI 3.1. All of the pin list, description, behavior, and timing are as per DFI 3.1. For more information, see the DFI 3.1 specification.

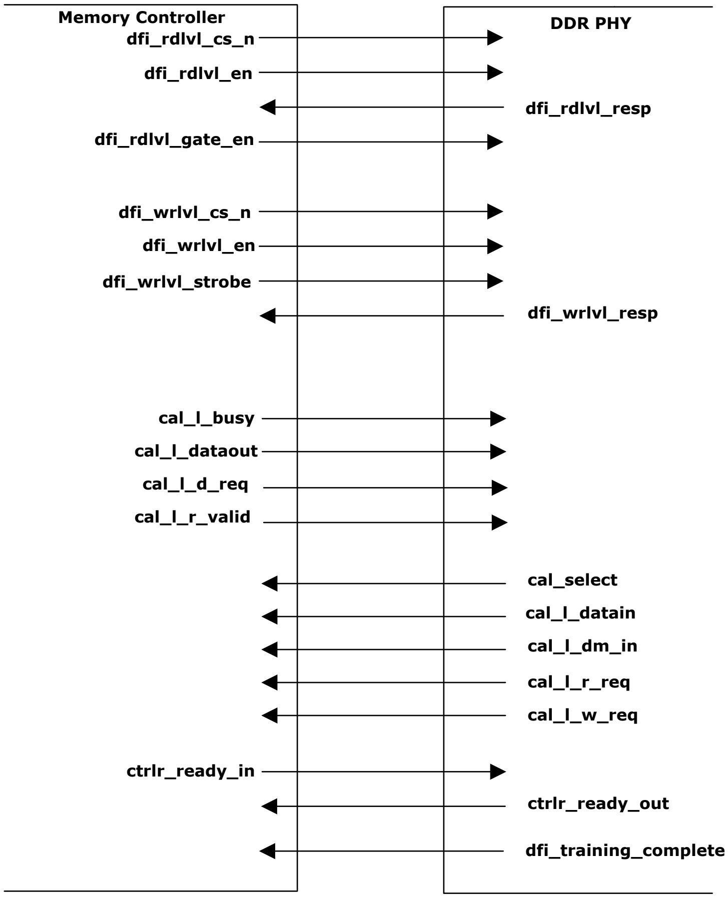
The following figure shows how PHY integrates with a memory controller for control, write and read interfaces.

Figure 3-29. DDR PHY DFI Interface


Memory Controller
DDR PHY
dfi_address[15:0] {P0,P1,P2,P3}
dfi_bank[2:0] {P0,P1,P2,P3}
dfi_ras_n {P0,P1,P2,P3}
dfi_cas_n {P0,P1,P2,P3}
dfi_we_n {P0,P1,P2,P3}
dfi_cs_n {P0,P1,P2,P3}
dfi_cke {P0,P1,P2,P3}
dfi_odt {P0,P1,P2,P3}
dfi_reset_n {P0,P1,P2,P3}
dfi_wrdata_en {P0,P1,P2,P3}
dfi_wrdata_cs_n {P0,P1,P2,P3}
dfi_wrdata_mask {P0,P1,P2,P3}
dfi_wrdata {P0,P1,P2,P3}
dfi_rddata_en {P0,P1,P2,P3}
dfi_rddata {W0,W1,W2,W3}
dfi_rddata_cs_n {P0,P1,P2,P3}
dfi_rddata_valid {W0,W1,W2,W3}
dfi_init_start
dfi_init_complete

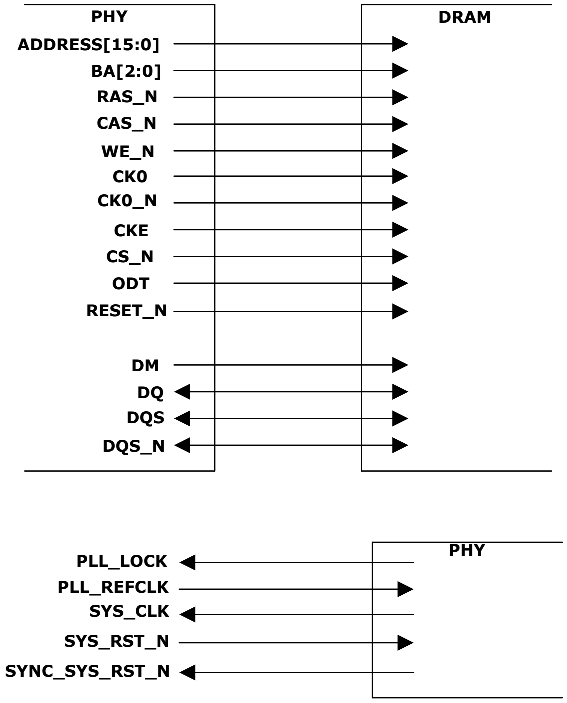
The following figure shows how the training IP inside the DDR PHY integrates with a memory controller. The training IP includes write calibration.

Figure 3-30. DDR PHY Training IP Interface with Write Calibration



The following figure shows how DDR PHY interfaces with DRAM, PLL, and Miscellaneous.

Figure 3-31. DDR PHY Interface with DRAM, PLL, and Miscellaneous



3.7.2 PHY Configurator

The PolarFire DDR PHY IP core is delivered with a configurator, which includes parameters like bus width, density, organization and frequency of operation. The port widths on DFI read and write interfaces depend on the DQ bus width selected for the design.

The following tables list the DFI read and write interfaces and their corresponding DQ bus width on the DRAM interface.

Table 3-22. DFI Port Widths Vs DRAM DQ Bus Width (16-bit Interface)

DFI Signal name	Interface	DRAM DQ Width	Remarks
dfi_rddata_w0[31:0]	READ	DQ[15:0] DQS[1:0] DQS_N[1:0]	dfi_rddata_w* width is [31:0] for 16 bit DRAM interface. (4 x 32 = 128 bit data transferred to controller in a single SYS_CLK)
dfi_rddata_w1[31:0]			
dfi_rddata_w2[31:0]			
dfi_rddata_w3[31:0]			
dfi_rddata_valid_w0[1:0]			dfi_rddata_valid width is 1:0 for two byte lanes.
dfi_rddata_valid_w1[1:0]			
dfi_rddata_valid_w2[1:0]			
dfi_rddata_valid_w3[1:0]			
dfi_rddata_cs_n_p0**			cs_n_p** width is 1. These signals are always asserted in the Libero SoC design
dfi_rddata_cs_n_p1**			
dfi_rddata_cs_n_p2**			
dfi_rddata_cs_n_p3**			
dfi_rddata_en_p0[1:0]*			*dfi_rddata_en_p* is 15:0. Out of this only 1:0 are used for two byte lanes.
dfi_rddata_en_p1[1:0]*			
dfi_rddata_en_p2[1:0]*			
dfi_rddata_en_p3[1:0]*			
dfi_wrdata_p0[31:0]	WRITE	DQ[15:0] DQS[1:0] DQS_N[1:0]	dfi_wrdata_p* width is [31:0] for 16 bit DRAM interface. (4 x 32 = 128 bit Data transferred from controller in a single SYS_CLK)
dfi_wrdata_p1[31:0]			
dfi_wrdata_p2[31:0]			
dfi_wrdata_p3[31:0]			
dfi_wrdata_mask_p0[3:0]			Data on dfi_wrdata is valid only when wrdata_mask is ‘0’ (4bytes of wrdata[31:0] have 4 bits of mask)
dfi_wrdata_mask_p1[3:0]			
dfi_wrdata_mask_p2[3:0]			
dfi_wrdata_mask_p3[3:0]			
dfi_wrdata_cs_n_p0**			cs_n_p** width is 1. These signals are always asserted in Libero SoC design
dfi_wrdata_cs_n_p1**			
dfi_wrdata_cs_n_p2**			
dfi_wrdata_cs_n_p3**			
dfi_wrdata_en_p0[1:0]*			*dfi_wrdata_en_p* is 15:0. Out of this only 1:0 are used for two byte lanes.
dfi_wrdata_en_p1[1:0]*			
dfi_wrdata_en_p2[1:0]*			
dfi_wrdata_en_p3[1:0]*			

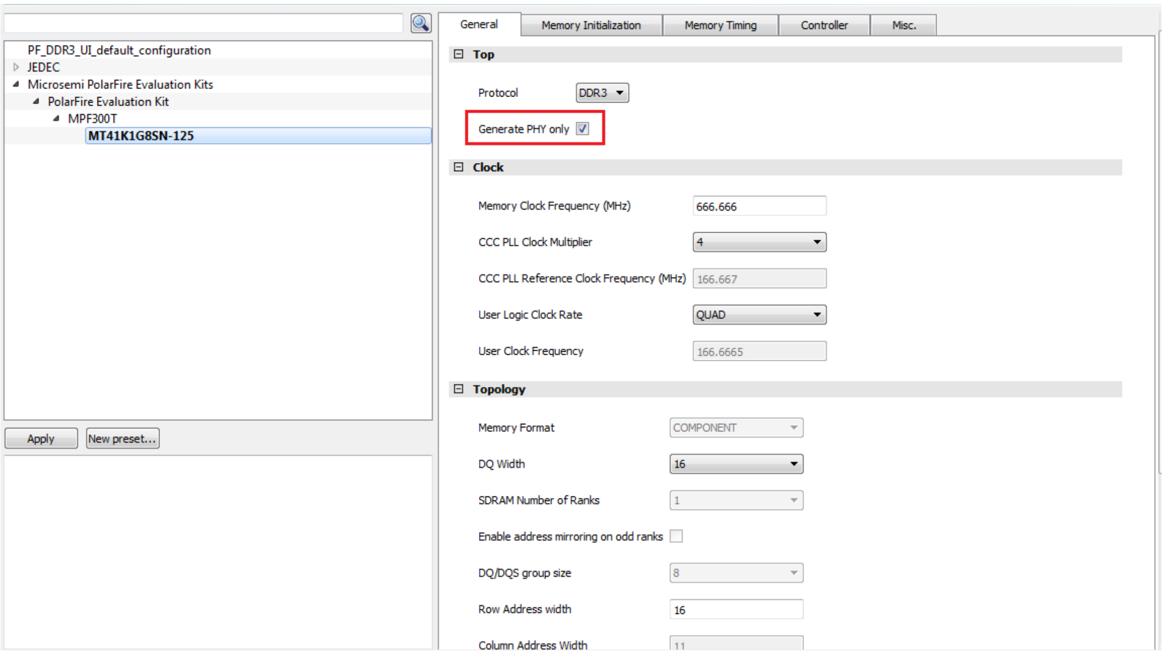
Table 3-23. DFI PORT WIDTH VS DRAM DQ BUS WIDTH – 32 BIT INTERFACE

DFI Signal name	Interface	DRAM DQ Width	Remarks
dfi_rddata_w0[63:0]	READ	—	dfi_rddata_w* width is [63:0] for 32 bit DRAM interface. (4 x 64 = 256 bit Data transferred to controller in a single SYS_CLK)
dfi_rddata_w1[63:0]		—	
dfi_rddata_w2[63:0]		—	
dfi_rddata_w3[63:0]		—	
dfi_rddata_valid_w0[3:0]		—	dfi_rddata_valid width is 3:0 for four byte lanes.
dfi_rddata_valid_w1[3:0]		—	
dfi_rddata_valid_w2[3:0]		DQ[31:0]	
dfi_rddata_valid_w3[3:0]		DQS[3:0]	
dfi_rddata_cs_n_p0**		DQS_N[3:0]	cs_n_p** width is 1. These signals are always asserted in Libero SoC design.
dfi_rddata_cs_n_p1**		—	
dfi_rddata_cs_n_p2**		—	
dfi_rddata_cs_n_p3**		—	
dfi_rddata_en_p0[3:0]*		—	*dfi_rddata_en_p* is 31:0. Out of this only 3:0 are used for four byte lanes.
dfi_rddata_en_p1[3:0]*		—	
dfi_rddata_en_p2[3:0]*		—	
dfi_rddata_en_p3[3:0]*		—	
dfi_wrdata_p0[63:0]	—	—	dfi_wrdata_p* width is [63:0] for 32 bit DRAM interface. (4 x 64 = 256 bit Data transferred from controller in a single SYS_CLK)
dfi_wrdata_p1[63:0]	—	—	
dfi_wrdata_p2[63:0]	—	—	
dfi_wrdata_p3[63:0]	—	—	
dfi_wrdata_mask_p0[7:0]	—	—	Data on dfi_wrdata is valid only when wrdata_mask is ‘0’ (8 bytes of wrdata[63:0] have 8 bits of mask)
dfi_wrdata_mask_p1[7:0]	—	—	
dfi_wrdata_mask_p2[7:0]	—	—	
dfi_wrdata_mask_p3[7:0]	—	DQ[31:0]	
dfi_wrdata_cs_n_p0**	WRITE	DQS[3:0]	cs_n_p** width is 1. These signals are always asserted in Libero SoC design
dfi_wrdata_cs_n_p1**	—	DQS_N[3:0]	
dfi_wrdata_cs_n_p2**	—	—	
dfi_wrdata_cs_n_p3**	—	—	
dfi_wrdata_en_p0[3:0]*	—	—	*dfi_wrdata_en_p* is 31:0. Out of this only 3:0 are used for four byte lanes.
dfi_wrdata_en_p1[3:0]*	—	—	
dfi_wrdata_en_p2[3:0]*	—	—	
dfi_wrdata_en_p3[3:0]*	—	—	

The configuration of the controller and the DDR PHY must be matched. For example, if the controller is configured for 32-bit DQ width, PHY also must be configured for the 32-bit DQ width.

The DDR3 PHY is instantiated by selecting the Generate PHY Only check box in the PolarFire DDR3 configurator > General Tab settings. When this check box is selected as shown in the following figure, the remaining tabs are grayed out because these settings are pertaining to the controller.

Figure 3-32. DDR3 PHY-Only Configurator



⊟ **Top**

Protocol	DDR3
Generate PHY only	☑

⊟ **Clock**

Memory Clock Frequency (MHz)	666.666
CCC PLL Clock Multiplier	4
CCC PLL Reference Clock Frequency (MHz)	166.667
User Logic Clock Rate	QUAD
User Clock Frequency	166.6665

⊟ **Topology**

Memory Format	COMPONENT
DQ Width	16
SDRAM Number of Ranks	1
Enable address mirroring on odd ranks	□
DQ/DQS group size	8
Row Address width	16
Column Address Width	11

The following parameters are configurable:

- Protocol
- Memory Clock Frequency
- PLL Reference Clock
- User Logic Clock Rate
- CCC PLL Clock Multiplier
- DQ Width
- Enable DM
- Enable ECC

3.7.3 Initialization and Bring-Up of the DDR Subsystem

The DDR subsystem initializes in the following sequence:

1. DFI Initialization (DFI_INIT_START & DFI_INIT_COMPLETE)
2. DRAM Initialization
3. ZQCL
4. Write Leveling
5. DQS Gate Training
6. Read Data Eye Training
7. Write Calibration
8. Simple Burst Write
9. Simple Burst Read

Note: Any DFI compliant controller must follow this sequence.

3.7.3.1 DFI Initialization

In this step, the PHY de-asserts the **DFI_INIT_COMPLETE** signal (1’b0) to indicate that the PHY is not ready to accept DFI commands. After the RESET release, the PHY goes through the following PLL Lock and Clock Training sequence, which does not require any controller or user intervention.

1. Initializes the PLL and waits for the lock signal.
2. Switches SYS_CLK to PLL HSIO_CLK/4 clock.
3. Sets the IOA buffer Vref to 50%.
   Note: V_{REF} is set either internally or externally by a pin at device power-up.
4. Aligns HSIO_CLK to SYS_CLK.
5. Aligns REF_CLK (CK0/CK0_N) to ADD/CMD bus.

This completes the DFI and IOD initialization. The Training IP asserts the **DFI_INIT_COMPLETE** signal (1) indicating the controller to start the DDR training procedure. The PHY is now ready to accept DFI commands.

Note: After dfi_init_complete assertion, the controller must keep DFI_RESET_N_P* / DFI_CKE_P* low for 200 µs, and then drive DFI_RESET_N_P* high. After that, the controller must wait for another 500 µs and assert DFI_CKE_P* high. This is required as per the JEDEC specification.

3.7.3.2 DRAM Initialization

After the first step, the controller starts the JEDEC compliant initialization sequence. The controller initializes the DRAM devices by writing to their mode registers. The DRAM device is initialized by issuing MRS command to mode registers as per the JEDEC specification. For example, the MRS command is issued on the DRAM interface by asserting CS_N=0, RAS_N=0, CAS_N=0, and WE_N=0. The Mode register address is sent on BA pins and the Mode register data is sent on DRAM ADDR pins.

The controller must drive these signals accordingly on the DFI interface. The latency in the PHY is identical on all of the control interface signals. The controller must meet the timing of the DRAM interface on the DFI interface.

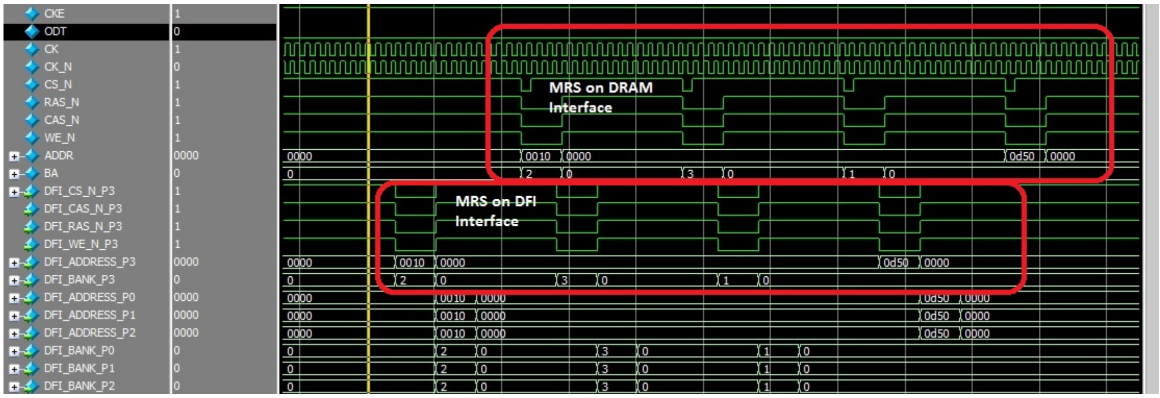
Figure 3-33 shows the DFI to DRAM latency and is an example of how the memory controller drives the DFI interface to obtain the desired functionality and timing on the DRAM interface.

In the DFI Specification, there is a single timing parameter (tCTRL_DELAY, which needs to be set in the DDR controller) for all of the control interface. The controller must implement this DFI timing inside it.

The latency seen in Figure 3-33 is the minimal latency observed from DFI command to DRAM command. For this example tCTRL_DELAY was set to 0. Any tCTRL_DELAY value set by the controller is added to the latency shown in Figure 3-33.

Note: The DDR3 example is used to show the simulation.

Figure 3-33. DFI To DRAM Control Interface Timing Relationship



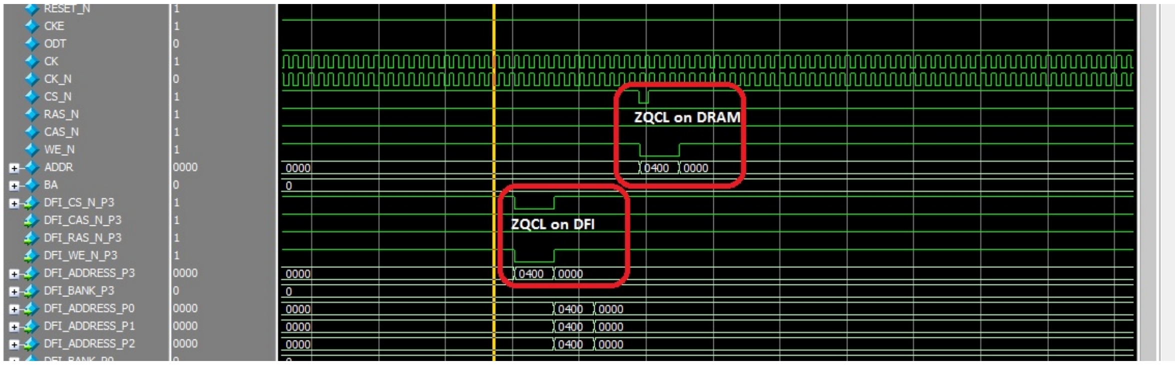
3.7.3.3 ZQ Calibration

After the second step, the controller performs the ZQ calibration. This command takes 512 REF_CLK cycles to complete the ZQ calibration. After issuing this command, the controller must wait for 512 REF_CLK cycles. The

ZQCL command is issued (by asserting CS_N=0, WE_N=0, and DRAM ADDR=0x400 for DDR3). The controller drives these signals accordingly on the DFI interface.

Figure 3-34 shows ZQCL on DFI followed by ZQCL on DRAM interface. There is a single timing parameter (tCTRL_DELAY) for all of the control Interface. The latency seen Figure 3-34 is minimal latency observed from the DFI command to DRAM command. For this example, tCTRL_DELAY was set to 0. Any tCTRL_DELAY value set by the controller is added to the latency shown in Figure 3-34.

Figure 3-34. ZQCL On DFI Followed By ZQCL On DRAM INTERFACE

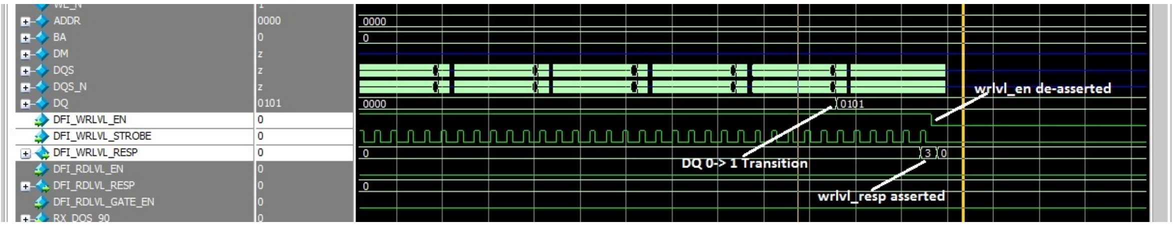


3.7.3.4 Write Leveling

After the third step, the controller enters the write leveling mode. Any DFI compliant controller must follow this Write Leveling sequence to complete the training step:

1. To enter the Write Leveling mode, the controller must issue Mode Register Set (MRS) to MR1 for DDR3/ DDR4, MR2 for LPDDR3, and set the write leveling enable bit (DFI_WRLVL_EN = 1).
2. After detecting DFI_WRLVL_EN =1, the Training IP starts the training algorithm and aligns CLK to DQS. Write leveling is a JEDEC defined method used to detect 0 to 1 transition on CLK by DQS.
3. DQS samples the CLK value and feeds it back on the prime DQ on the DRAM DQ byte lane.
4. When the prime DQ value transitions from 0 to 1, CLK is aligned to DQS within the JEDEC specified timing limits. The Training IP performs this procedure on all byte lanes concurrently.
5. The controller sends continuous strobe (DFI_WRLVL_STROBE) pulses to capture DFI_WRLVL_RESP. DFI_WRLVL_STROBE initiates the capture of DFI_WRLVL_RESP within the PHY DQ bus.
6. The Training IP asserts DFI_WRLVL_RESP = 1 indicating the completion of Write Leveling to the controller.
   Note: In the Fast Simulation Mode, PHY ignores the WRLVL request without responding. The controller must bypass the write leveling, read gate leveling, and read eye training (Read Leveling) stages.
7. When the controller receives the DFI_WRLVL_RESP asserted signal, it de-asserts the write to mode register MR1 and also de-asserts DFI_WRLVL_EN to disable write leveling enable bit.
8. The delay values detected for each byte lane are stored in the PHY lane controller. The following figure shows the controller sampling DFI_WRLVL_RESP and disabling DFI_WRLVL_EN.

Figure 3-35. DFI Write Leveling Complete

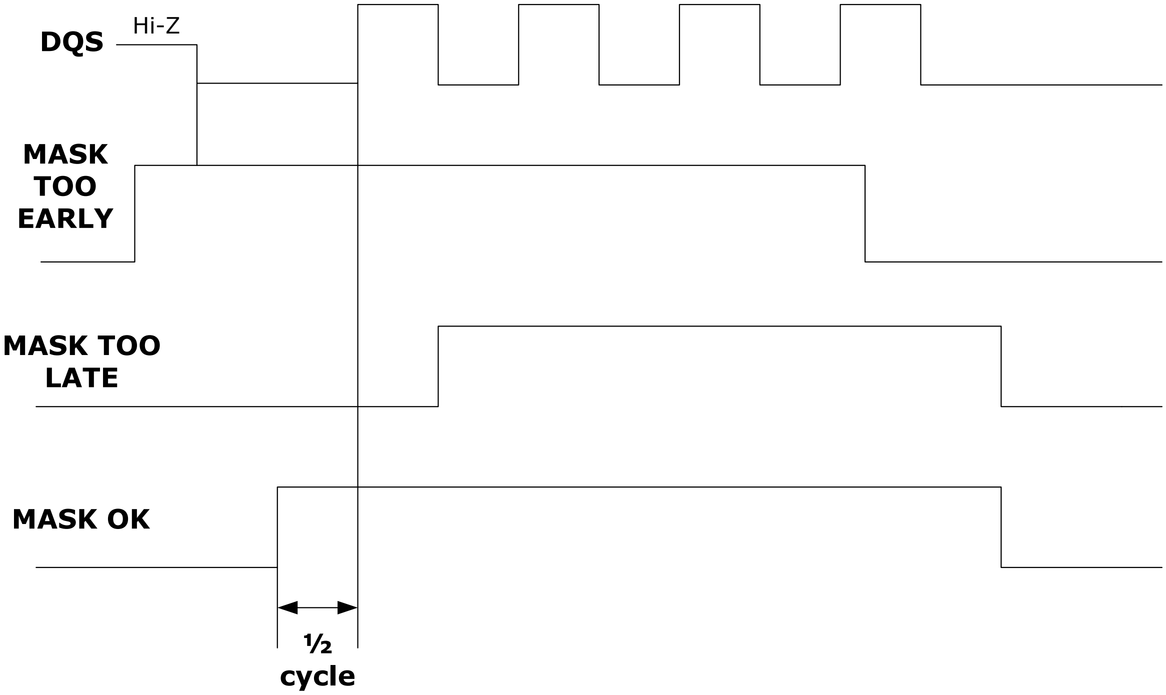


3.7.3.5 DQS Gate Training

After the fourth step, Controller initiates the DQS Gate Training in the following sequence:

1. The controller initiates the DQS Gate Training by asserting DFI_RDLVL_GATE_EN. The controller must write 0 to MR1 (for DDR3/4)/MR2 (for LPDDR3) bit 7 by issuing the MRS command for disabling the Write Leveling of Step 4.
2. The controller writes 1 to MR3 bit 2 for enabling data on DQ bus flow from MPR (Multi-Purpose Register) for DDR3 and DDR4 SDRAMs.
3. The controller must assert DFI_RDLVL_GATE_EN = 1 and wait for dfi_rdlvl_resp. After detecting DFI_RDLVL_GATE_EN = 1, the Training IP starts the DQS Gate Training algorithm.
4. The Training IP positions the DQS_GATE mask to enable READ_DQ and READ_DQS as shown in the following figure.

Figure 3-36. DQS Gate Training Conceptual View



When the DQS GATE Mask is positioned correctly, the read data matches the pattern selected in the MPR register. Then, the Training IP will assert DFI_RDLVL_RESP. After sampling DFI_RDLVL_RESP = 1, the controller can de-assert DFI_RDLVL_GATE_EN = 0. For DDR3/4, the controller must write 0 to MR3 bit 2 for restoring normal data flow on the DQ bus. (MR3 bit 2 = 0 > Normal Data flow; MR3 bit 2 = 1 > Data flow from MPR)

Otherwise, DQS strobe pulses will be missing and the read data will not match the pattern selected in the MPR register.

Note: For DDR3/4, the controller must issue regular READ commands to read the data from MPR. For LPDDR3, the controller must issue mode register read to MR32 or MR40. The time between reads must at least be 62 SYS_CLK cycles long so that the Training IP can stabilize. The DFI READ time during the MPR reads must be the same as the normal data read operations. During the DQS Gate training, the timing of DFI_RDDATA_VALID is not trained.

3.7.3.6 Read Data Eye Training

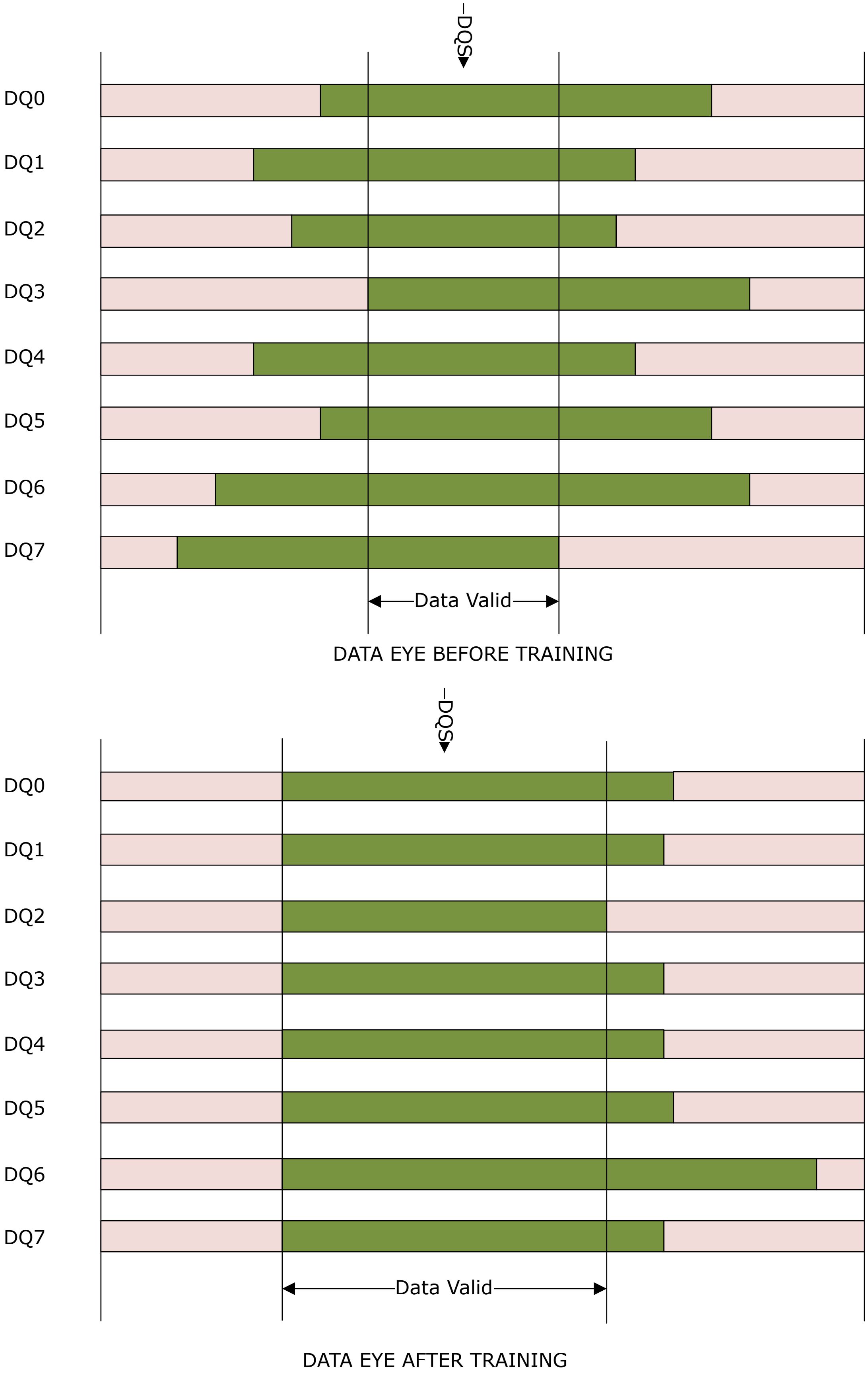
After the fifth step, the controller initiates The READ Data Eye training in the following sequence.

1. The controller reads a continuous data pattern of ‘00’ followed by ‘FF’ from DRAM. This is same as enabling MPR and reading a fixed pattern from MPR for DDR3/4. DDR3 supports only one pattern in MPR.
2. Enabling the data flow for DDR3/4 from MPR is similar to what is done in DQS Gate training. For DDR3/4, the controller must set MR3 bit 2 to 1 (data flow from MPR) by issuing the MRS command.
3. The controller must assert DFI_RDLVL_EN = 1 to initiate READ DATA EYE training. The Training IP sweeps the input tap delay, reads the data and compares it with the expected results.
4. The left and right margins of the passing data are stored for all the bits in the byte lane. The left and right margins of each bit are analyzed and the appropriate delay value is applied to each bit to maximize the valid data window.
5. After the completion of READ Data eye training, the Training IP asserts DFI_RDLVL_RESP. After sampling DFI_RDLVL_RESP = 1, the controller can de-assert DFI_RDLVL_EN = 0. For DDR3/4, the controller must write 0 to MR3 bit 2 for restoring the normal data flow on the DQ bus.

Note: For DDR3/4, the controller must issue the regular ‘READ’ commands to read data from the MPR register. For LPDDR3, the controller must issue mode register read to MR32 or MR40.

The following figure shows READ DATA EYE before and after training.

Figure 3-37. READ DATA EYE BEFORE AND AFTER TRAINING



3.7.3.7 Write Calibration

The write leveling described in step 4 addresses only timing. It does not address latency. In the fly-by topology, there is a latency from the first DRAM device to the last DRAM device. Write calibration is not defined in the DFI Specification. This implementation of Write calibration is specific to PolarFire and PolarFire SoC designs.

Correcting for latency is accomplished by pushing both DQ and DQS by a full clock cycle (memory clock, CK0/CK0_N). The Training IP handles the Write calibration. The Training IP writes and reads known patterns through the controller and pushes DQ/DQS by a full clock cycle until Writes and Reads match. This step is also called as coarse correction because DQS is pushed by a full clock cycle. The Write calibration is done on a per byte lane basis.

The Write calibration is handled by a group of signals called CAL_TRAINING interface.

After the completion of READ Data eye training (Step 6), the Training IP asserts the DFI_TRAINING_COMPLETE signal. The Write Calibration sequence is as follows.

1. Training IP issues TRAINING_COMPLETE.
2. Controller asserts CTRLR_READY_IN.
3. For DDR3/4, the controller writes to ‘MR3’ ‘0x0’ to restore normal data flow. This restores the DQ data flow in the normal mode.
4. Controller de-asserts CAL_L_BUSY (0x0). Training IP asserts CAL_SELECT (0x1, ~CAL_L_BUSY).
5. Training IP issues CAL_L_W_REQ to issue ‘WRITE’ Command.
6. Controller issues CAL_L_D_REQ one clock before Data required.
7. Training IP provides ‘CAL_L_DATAIN’ & ‘CAL_L_DM_IN’ in the following clock.
8. Controller issues ‘WRITE’ command on DFI to be written to DRAM.
9. After waiting for 19 SYS_CLK clock cycles, the Training IP issues CAL_L_R_REQ to issue ‘READ’ command.
10. Controller Reads Data from DRAM and presents it on ‘CAL_L_DATAOUT’ with ‘CAL_L_R_VALID’.
11. Training IP compares DATA written and read.
12. If DATA written and read do not match, Training IP pushes DQ/DQS by full clock cycle at a time, until DATA written matches DATA read.
13. Upon successful completion of this sequence, the Training IP asserts CTRLR_READY_OUT and de-asserts CAL_SELECT in the following clock.
14. Controller will de-assert CAL_L_BUSY (0x1) to indicate Write calibration complete.

The following figures show the Write calibration sequence.

Figure 3-38. Write And Read Of Known Patterns During Write Calibration

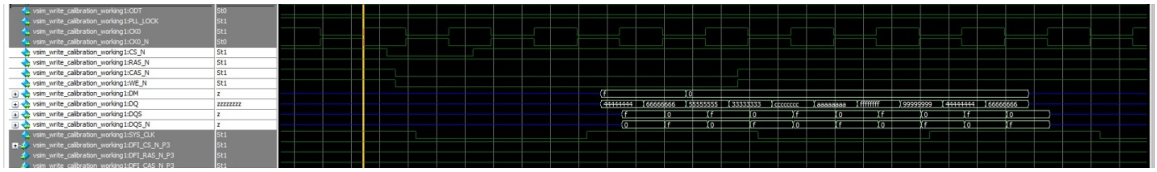
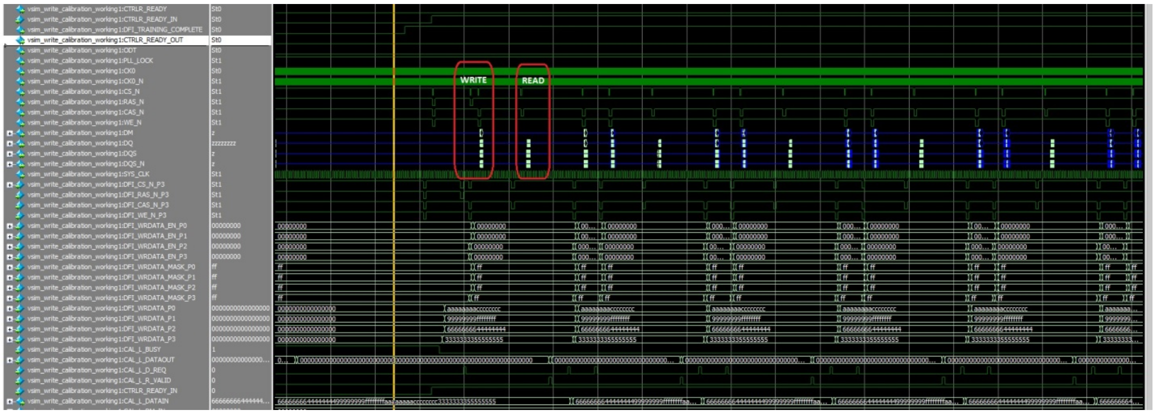
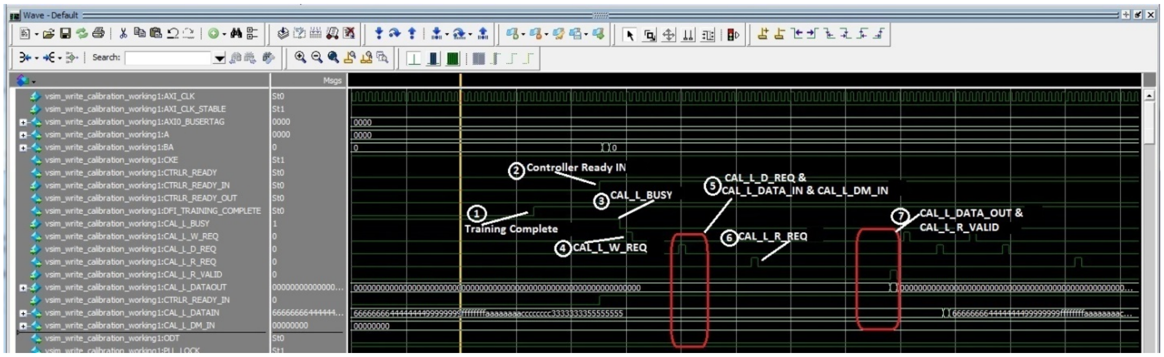

vsim_write_calibration_working1:ODT St0
vsim_write_calibration_working1:PLL_LOCK St1
vsim_write_calibration_working1:CK0 St1
vsim_write_calibration_working1:CK0_N St0
vsim_write_calibration_working1:CS_N St1
vsim_write_calibration_working1:RAS_N St1
vsim_write_calibration_working1:CAS_N St1
vsim_write_calibration_working1:WE_N St1
vsim_write_calibration_working1:DM z
vsim_write_calibration_working1:DQ zzzzzzzz
vsim_write_calibration_working1:DQS z
vsim_write_calibration_working1:DQS_N z
vsim_write_calibration_working1:SYS_CLK St1
vsim_write_calibration_working1:DFI_CS_N_P3 St1
vsim_write_calibration_working1:DFI_RAS_N_P3 St1
vsim_write_calibration_working1:DFI_CAS_N_P3 St1
f
0
44444444
66666666
55555555
33333333
cccccccc
aaaaaaaa
ffffffff
99999999
44444444
66666666
f 0 f 0 f 0 f 0 f 0
0 f 0 f 0 f 0 f 0 f

Figure 3-39. Write And Read Sequence During Write Calibration



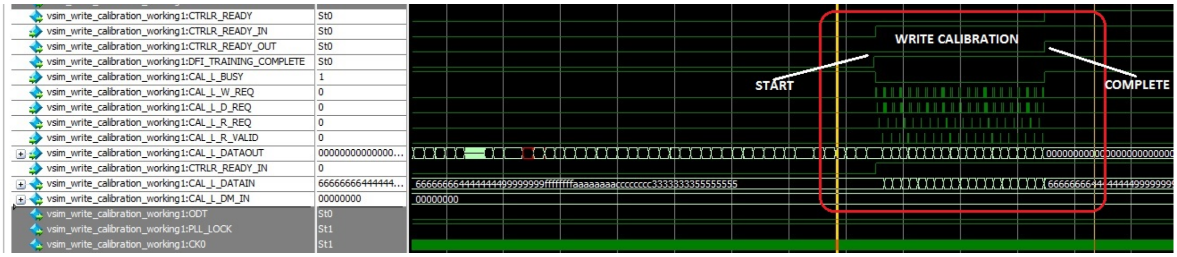
vsim_write_calibration_working1:CTRLR_READY	St0
vsim_write_calibration_working1:CTRLR_READY_IN	St0
vsim_write_calibration_working1:DFI_TRAINING_COMPLETE	St0
vsim_write_calibration_working1:CTRLR_READY_OUT	St0
vsim_write_calibration_working1:ODT	St0
vsim_write_calibration_working1:PLL_LOCK	St1
vsim_write_calibration_working1:CK0	St0
vsim_write_calibration_working1:CK0_N	St1
vsim_write_calibration_working1:CS_N	St1
vsim_write_calibration_working1:RAS_N	St1
vsim_write_calibration_working1:CAS_N	St1
vsim_write_calibration_working1:WE_N	St1
vsim_write_calibration_working1:DM	z
vsim_write_calibration_working1:DQ	zzzzzzzz
vsim_write_calibration_working1:DQS	z
vsim_write_calibration_working1:DQS_N	z
vsim_write_calibration_working1:SYS_CLK	St1
vsim_write_calibration_working1:DFI_CS_N_P3	St1
vsim_write_calibration_working1:DFI_RAS_N_P3	St1
vsim_write_calibration_working1:DFI_CAS_N_P3	St1
vsim_write_calibration_working1:DFI_WE_N_P3	St1
vsim_write_calibration_working1:DFI_WRDATA_EN_P0	00000000
vsim_write_calibration_working1:DFI_WRDATA_EN_P1	00000000
vsim_write_calibration_working1:DFI_WRDATA_EN_P2	00000000
vsim_write_calibration_working1:DFI_WRDATA_EN_P3	00000000
vsim_write_calibration_working1:DFI_WRDATA_MASK_P0	ff
vsim_write_calibration_working1:DFI_WRDATA_MASK_P1	ff
vsim_write_calibration_working1:DFI_WRDATA_MASK_P2	ff
vsim_write_calibration_working1:DFI_WRDATA_MASK_P3	ff
vsim_write_calibration_working1:DFI_WRDATA_P0	0000000000000000
vsim_write_calibration_working1:DFI_WRDATA_P1	0000000000000000
vsim_write_calibration_working1:DFI_WRDATA_P2	0000000000000000
vsim_write_calibration_working1:DFI_WRDATA_P3	0000000000000000
vsim_write_calibration_working1:CAL_L_BUSY	1
vsim_write_calibration_working1:CAL_L_DATAOUT	00000000000000...
vsim_write_calibration_working1:CAL_L_D_REQ	0
vsim_write_calibration_working1:CAL_L_R_VALID	0
vsim_write_calibration_working1:CTRLR_READY_IN	0
vsim_write_calibration_working1:CAL_L_DATAIN	66666666444444...

Figure 3-40. Write Calibration Sequence



vsim_write_calibration_working1:AXI_CLK	St0
vsim_write_calibration_working1:AXI_CLK_STABLE	St1
vsim_write_calibration_working1:AXI0_BUSERTAG	0000
vsim_write_calibration_working1:A	0000
vsim_write_calibration_working1:BA	0
vsim_write_calibration_working1:CKE	St1
vsim_write_calibration_working1:CTRLR_READY	St0
vsim_write_calibration_working1:CTRLR_READY_IN	St0
vsim_write_calibration_working1:CTRLR_READY_OUT	St0
vsim_write_calibration_working1:DFI_TRAINING_COMPLETE	St0
vsim_write_calibration_working1:CAL_L_BUSY	1
vsim_write_calibration_working1:CAL_L_W_REQ	0
vsim_write_calibration_working1:CAL_L_D_REQ	0
vsim_write_calibration_working1:CAL_L_R_REQ	0
vsim_write_calibration_working1:CAL_L_R_VALID	0
vsim_write_calibration_working1:CAL_L_DATAOUT	00000000000000...
vsim_write_calibration_working1:CTRLR_READY_IN	0
vsim_write_calibration_working1:CAL_L_DATAIN	66666666444444...
vsim_write_calibration_working1:CAL_L_DM_IN	00000000
vsim_write_calibration_working1:ODT	St0
vsim_write_calibration_working1:PLL_LOCK	St1

Figure 3-41. Write Calibration Complete



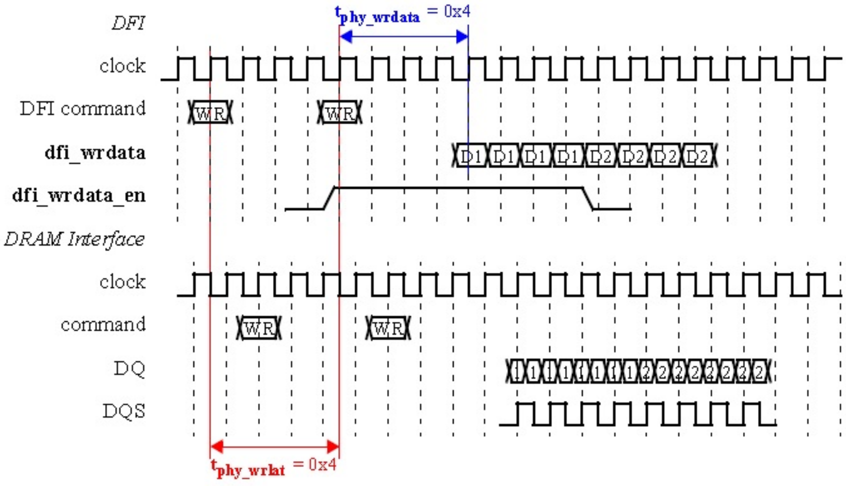
vsim_write_calibration_working1:CTRLR_READY	St0
vsim_write_calibration_working1:CTRLR_READY_IN	St0
vsim_write_calibration_working1:CTRLR_READY_OUT	St0
vsim_write_calibration_working1:DFI_TRAINING_COMPLETE	St0
vsim_write_calibration_working1:CAL_L_BUSY	1
vsim_write_calibration_working1:CAL_L_W_REQ	0
vsim_write_calibration_working1:CAL_L_D_REQ	0
vsim_write_calibration_working1:CAL_L_R_REQ	0
vsim_write_calibration_working1:CAL_L_R_VALID	0
vsim_write_calibration_working1:CAL_L_DATAOUT	00000000000000...
vsim_write_calibration_working1:CTRLR_READY_IN	0
vsim_write_calibration_working1:CAL_L_DATAIN	66666666444444...
vsim_write_calibration_working1:CAL_L_DM_IN	00000000
vsim_write_calibration_working1:ODT	St0
vsim_write_calibration_working1:PLL_LOCK	St1
vsim_write_calibration_working1:CK0	St1

3.7.3.8 Simple Burst Write

After Write Calibration, the DDR subsystem is ready for a normal operation. Typically, the controller asserts a signal like CTRLR_READY to let the host system know that the DDR subsystem is now ready for normal DDR operations.

This step describes a simple Burst Write operation along the DFI interface. This step describes how the controller must drive the DFI interface for a simple Burst Write operation and the associated timing parameters. There are two timing parameters associated with the WRITE Command, tphy_wrdata, and tphy_wrlat. Figure 3-42 shows how these timing parameters are used by the controller. The controller must be aware, must parameterize and hold constant all of the PHY related timing during operations. The controller must know the PHY protocol related timing defined in the DFI 3.1 specification. For more information on timing, see the DFI 3.1 sequence.

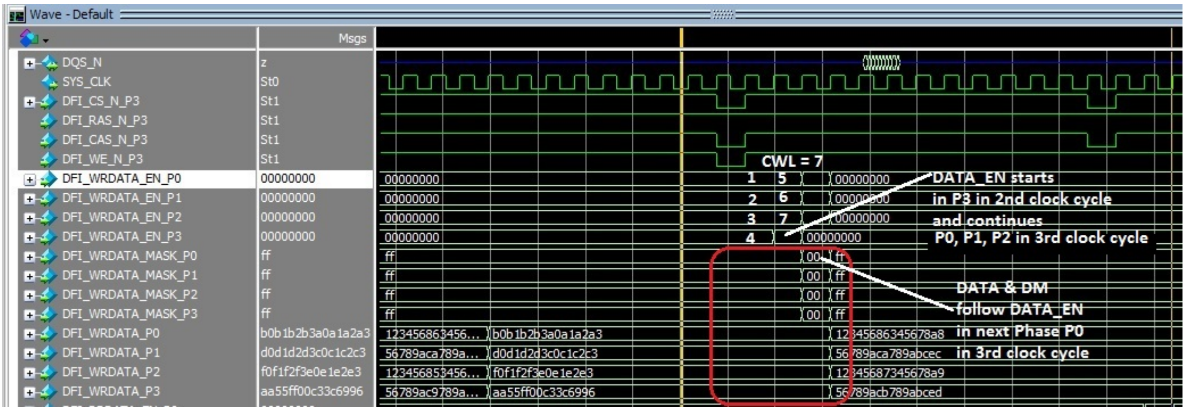
Figure 3-42. DFI Write Burst Example



In the frequency ratio (4:1) system, the controller must follow the phase relationship and timing. The following examples for CWL = 7 & CWL = 8 where the controller must calculate the phase to drive WRDATA_EN, WRDATA, and WRDATA_MASK.

If CWL = 7 and DFI_WRDATA_EN {P3} are asserted in two DFI clocks (SYS_CLK) after issuing the Write command. DFI_WRDATA_EN {P0, P1, P2} are asserted in the third clock cycle. WRDATA & WRDATA_MASK are issued in the next phase after the issuing WRDATA_EN, as shown in the following figure.

Figure 3-43. DFI WRITE BURST EXAMPLE FREQUENCY RATIO SYSTEM CWL = 7


Wave - Default

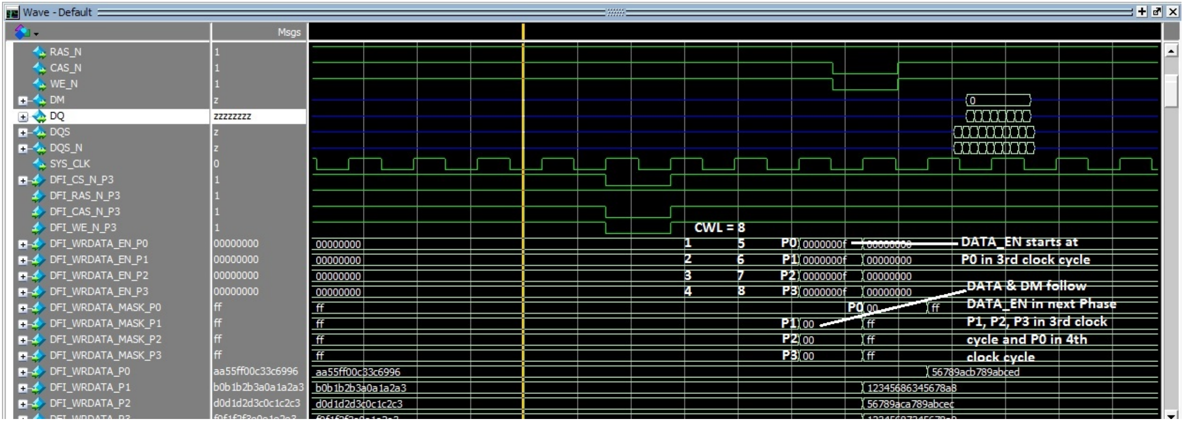
	Msgs
DQS_N	z
SYS_CLK	St0
DFI_CS_N_P3	St1
DFI_RAS_N_P3	St1
DFI_CAS_N_P3	St1
DFI_WE_N_P3	St1
DFI_WRDATA_EN_P0	00000000
DFI_WRDATA_EN_P1	00000000
DFI_WRDATA_EN_P2	00000000
DFI_WRDATA_EN_P3	00000000
DFI_WRDATA_MASK_P0	ff
DFI_WRDATA_MASK_P1	ff
DFI_WRDATA_MASK_P2	ff
DFI_WRDATA_MASK_P3	ff
DFI_WRDATA_P0	b0b1b2b3a0a1a2a3
DFI_WRDATA_P1	d0d1d2d3c0c1c2c3
DFI_WRDATA_P2	f0f1f2f3e0e1e2e3
DFI_WRDATA_P3	aa55ff00c33c6996

DATA_EN starts in P3 in 2nd clock cycle and continues P0, P1, P2 in 3rd clock cycle

DATA & DM follow DATA_EN in next Phase P0 in 3rd clock cycle

If CWL = 8 and DFI_WRDATA_EN {P0, P1, P2, P3} are asserted in three DFI clocks (SYS_CLK) after issuing the Write command. WRDATA & WRDATA_MASK are issued in the next phase after issuance of WRDATA_EN, as shown in the following figure.

Figure 3-44. DFI WRITE BURST EXAMPLE FREQUENCY RATIO SYSTEM CWL = 8



The following figures show simple burst write sequence for 16 and 32 bit interfaces.

Figure 3-45. ACTIVATE AND WRITE COMMANDS ON DFI & DRAM (16BIT INTERFACE)

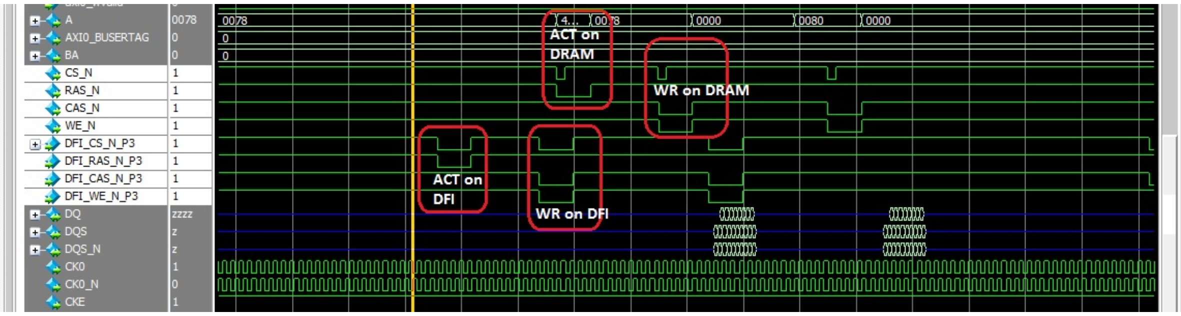


Figure 3-46. WRDATA ON DFI & DRAM - DFI 128 BITS (4 X 32) TRANSLATE TO 16 BIT BL8 ON DRAM

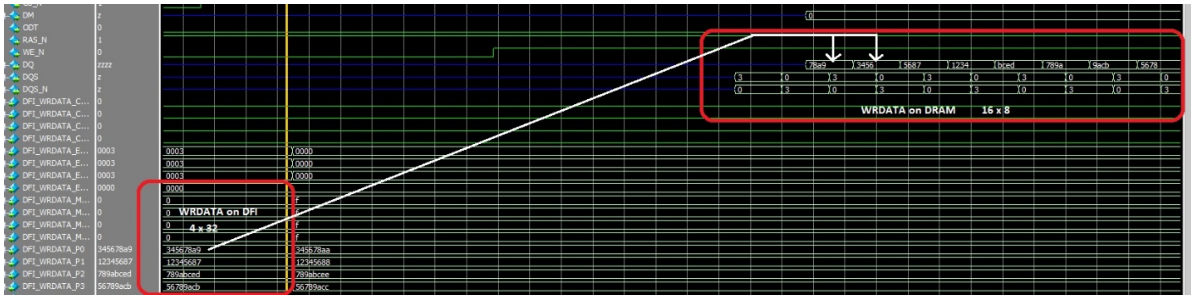
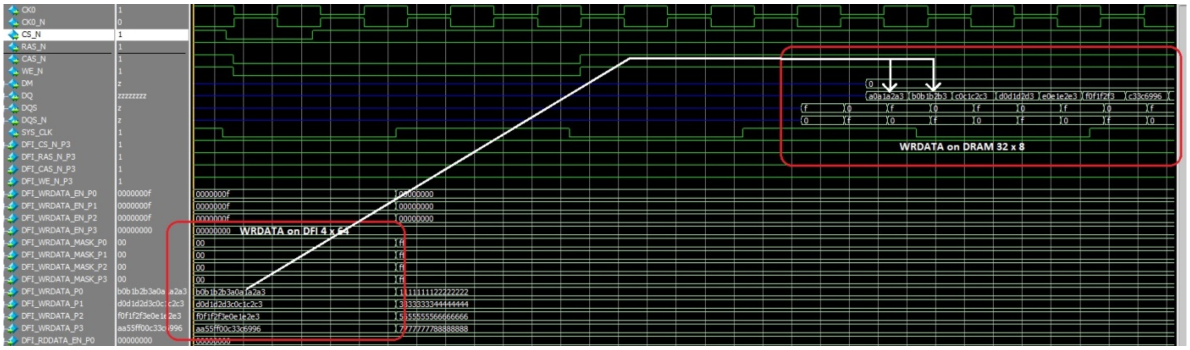


Figure 3-47. WRDATA ON DFI & DRAM - DFI 256 BITS (4 X 64) TRANSLATE TO 32 BIT BL8 ON DRAM



3.7.3.9 Simple Burst Read

This step describes simple burst read operation. Similar to WRITE there two timing parameters the controller need to be aware of trddata_en and tphy_rdlat. After issuing DFI READ command on control interface, Controller must assert DFI_RDDATA_EN after trddata_en clocks. The PHY responds to the READ command and DFI_RDDATA_EN with DFI_RDDATA and DFI_RDDATA_VALID. The following figure shows the sequence of events for a simple burst READ operation.

Figure 3-48. DFI Read Burst Example

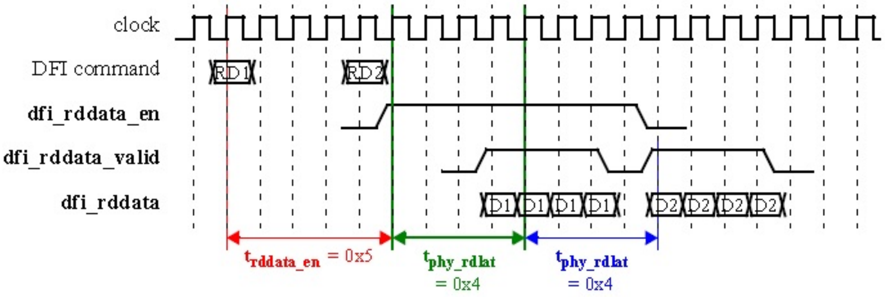


Figure 3-49. RDDATA ON DFI - 16 BIT BL8 ON DRAM TRANSLATE TO DFI 128 BITS (4 X 32)

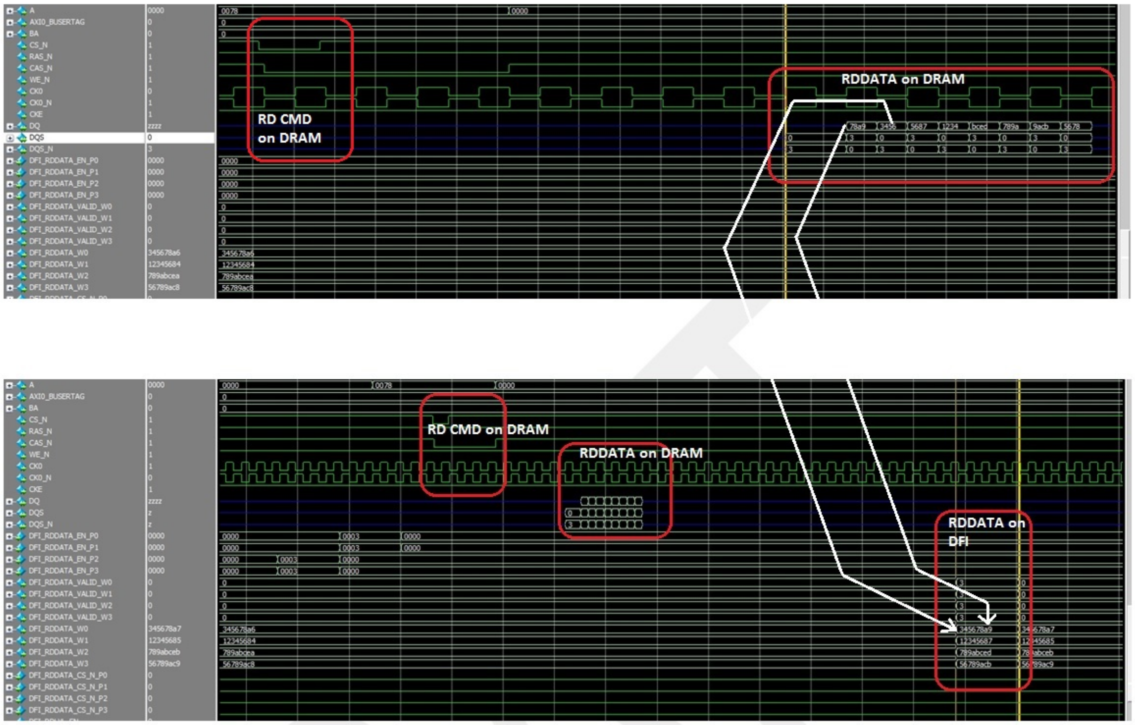
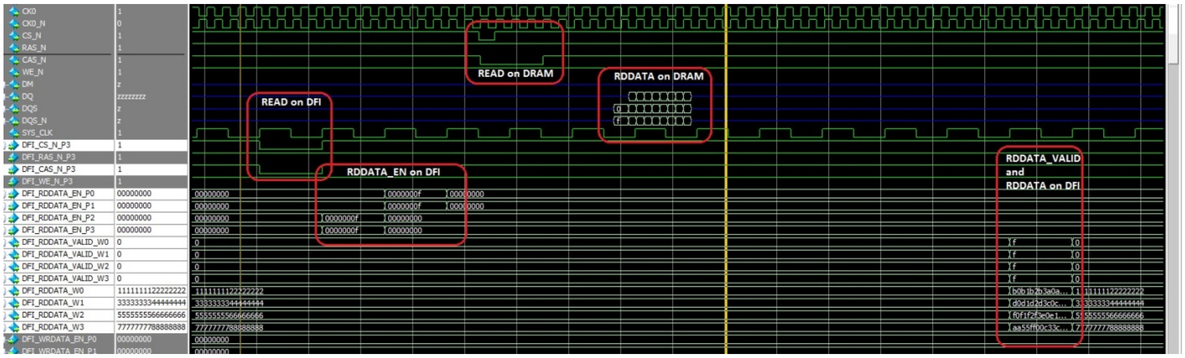


Figure 3-50. RDDATA ON DFI - 32 BIT BL8 ON DRAM TRANSLATE TO DFI 256 BITS (4 X 64)



Data on DFI_RDDATA is valid only when DFI_RDDATA_VALID is asserted. After receiving READ command, PHY responds with Read Data and Read Data Valid signals within (trddata_en + tphy_rdlat) cycles. PHY latencies associated with DFI-to-DRAM and DRAM-to-DFI are added to DFI timing, trddata_en and tphy_rdlat. For 16-bit interface, 128-bits of Data (16 x 8 (BL8)) is transferred to Controller on a single DFI clock cycle.

The timing parameters trddata_en and tphy_rdlat together define maximum number of cycles from assertion of READ command to assertion of DFI_RDDATA_VALID signal.

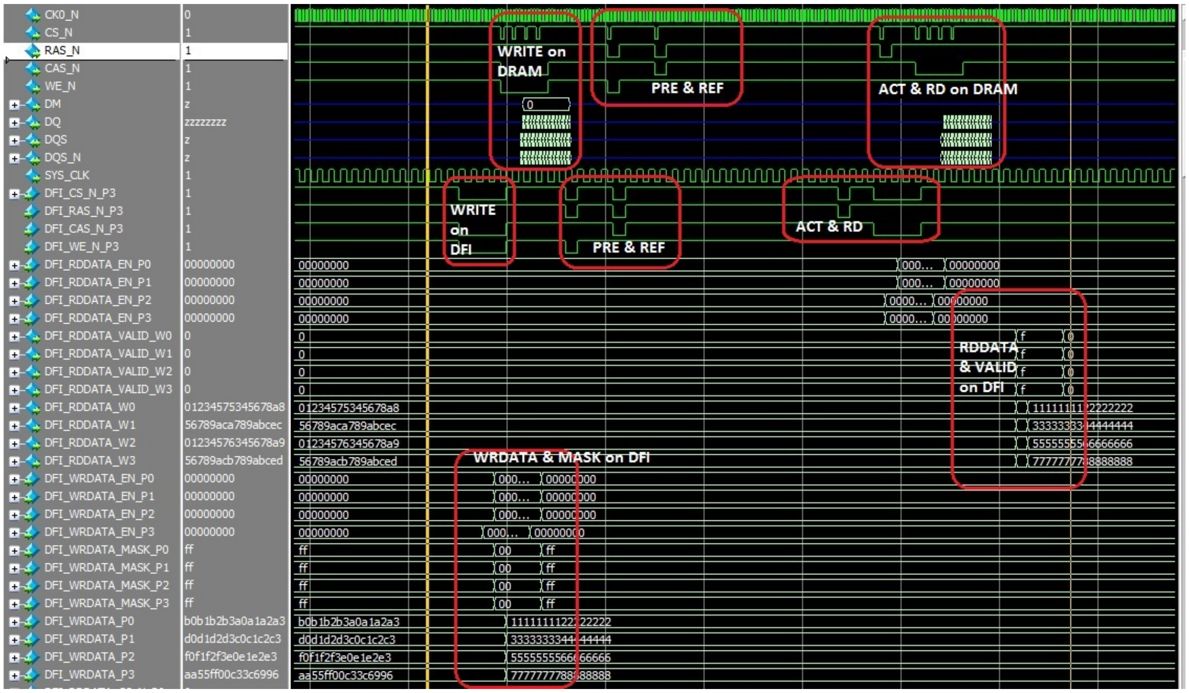
DFI dictates a timing relationship between DFI_RDDATA_EN and DFI_RDDATA_VALID specified by tphy_rdlat. DFI does not dictate exact number of cycles. Typically DFI_RDDATA_VALID can assert any time prior to max delay. DFI_RDDATA_VALID is trained and DFI_RDDATA is valid only when DFI_RDDATA_VALID is asserted.

3.7.4 Timing

Figure 3-51 shows a typical DDR3 protocol sequence and behavior of DFI and DRAM interfaces. All of the DDR protocol timing must be taken care by the controller. PHY is just an I/O and is responsible for timing parameters like setup time, hold time, clock duty cycle, and jitter. The PHY Timing Parameters are defined in the DFI 3.1 specification.

The following figure shows the continuous multi-burst operation on WRITE and READ, and also includes commands like PRECHARGE and REFRESH.

Figure 3-51. CONTINUOUS MULTI-BURST WRITE AND READ ON DFI/DRAM



CK0_N	0
CS_N	1
RAS_N	1
CAS_N	1
WE_N	1
DM	z
DQ	zzzzzzzz
DQS	z
DQS_N	z
SYS_CLK	1
DFI_CS_N_P3	1
DFI_RAS_N_P3	1
DFI_CAS_N_P3	1
DFI_WE_N_P3	1
DFI_RDDATA_EN_P0	00000000
DFI_RDDATA_EN_P1	00000000
DFI_RDDATA_EN_P2	00000000
DFI_RDDATA_EN_P3	00000000
DFI_RDDATA_VALID_W0	0
DFI_RDDATA_VALID_W1	0
DFI_RDDATA_VALID_W2	0
DFI_RDDATA_VALID_W3	0
DFI_RDDATA_W0	01234575345678a8
DFI_RDDATA_W1	56789aca789abcec
DFI_RDDATA_W2	01234576345678a9
DFI_RDDATA_W3	56789acb789abced
DFI_WRDATA_EN_P0	00000000
DFI_WRDATA_EN_P1	00000000
DFI_WRDATA_EN_P2	00000000
DFI_WRDATA_EN_P3	00000000
DFI_WRDATA_MASK_P0	ff
DFI_WRDATA_MASK_P1	ff
DFI_WRDATA_MASK_P2	ff
DFI_WRDATA_MASK_P3	ff
DFI_WRDATA_P0	b0b1b2b3a0a1a2a3
DFI_WRDATA_P1	d0d1d2d3c0c1c2c3
DFI_WRDATA_P2	f0f1f2f3e0e1e2e3
DFI_WRDATA_P3	aa55ff00c33c6996

3.8 Octal DDR PHY-Only Solution

Libero SoC Catalog includes the PolarFire Octal DDR PHY IP, which allows the generation of a PHY with following hard and soft blocks:

- I/O
- IOD logic
- PLL
- Soft fabric logic

The Octal DDR PHY supports xSPI (JESD251), HyperBUS, and ONFI. These memories use an 8-bit serial interface operating Double Data Rate (DDR) mode. These memories are used in applications such as the automotive semiconductor market, Industrial Ethernet, AI, Wearable device market, and IoT market. The Octal DDR PHY-Only solution supports up to 200 MHz operations and is available in GPIO and HSIO banks.

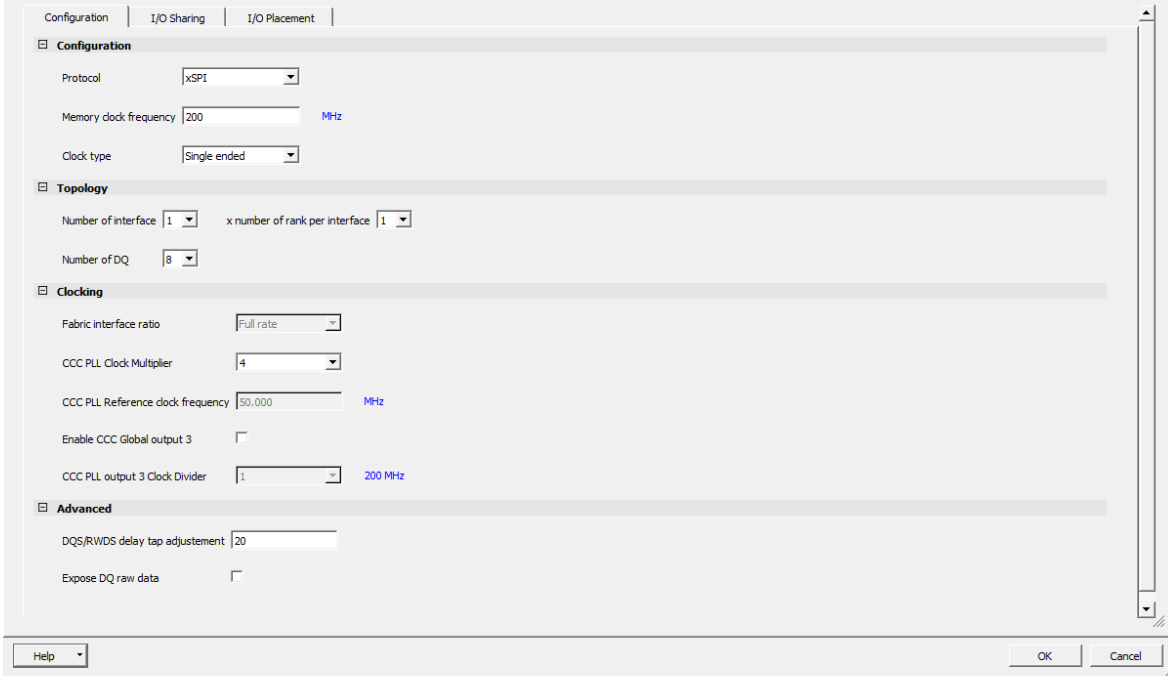
The Octal DDR PHY configurator supports up to eight shared parallel interfaces and up to two ranks per interface. The PHY relies on input static delay settings and static timing analysis for interface I/O timing closure. The interface is not intended to be a trained interface.

3.8.1 Configurator

The following figure shows the main GUI window of the Octal DDR PHY Configurator.

© 2021 Microchip Technology Inc.
and its subsidiaries

Figure 3-52. Octal DDR PHY Configurator Main GUI



⊟ **Configuration**

Protocol: xSPI

Memory clock frequency: 200 MHz

Clock type: Single ended

⊟ **Topology**

Number of interface: 1 x number of rank per interface: 1

Number of DQ: 8

⊟ **Clocking**

Fabric interface ratio: Full rate

CCC PLL Clock Multiplier: 4

CCC PLL Reference clock frequency: 50.000 MHz

Enable CCC Global output 3: □

CCC PLL output 3 Clock Divider: 1 200 MHz

⊟ **Advanced**

DQS/RWDS delay tap adjustement: 20

Expose DQ raw data: □

The following table lists the Octal DDR PHY Configurator options and parameters.

Table 3-24. Configurator Options and Parameters

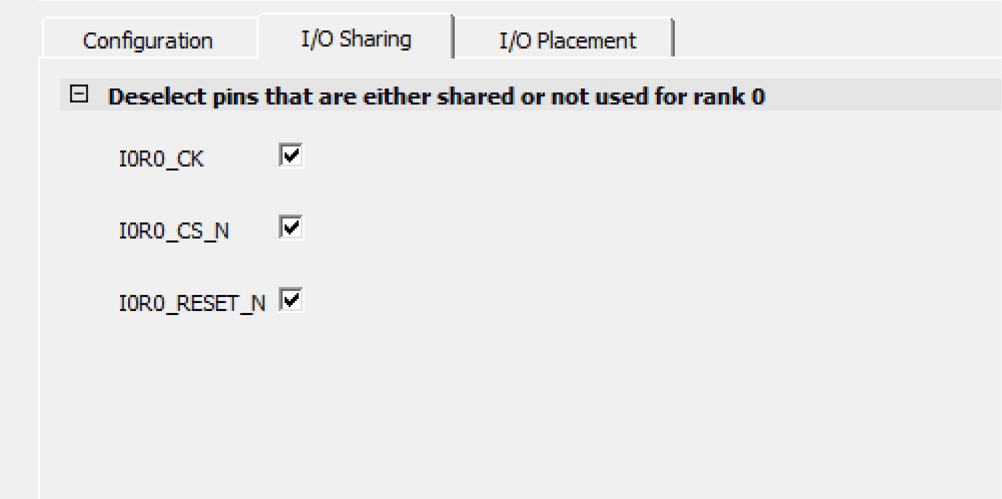
Parameter	Range	Description
PROTOCOL	XSPI	Protocol used to build the PHY
	ONFI	
	HYPERBUS	
	APMEMORY_PSRAM	
CLOCK_FREQ	1-200 MHz	Memory clock frequency
CLOCK_TYPE	SINGLE_ENDED	1 clock
	DIFFERENTIAL	True differential clock pair
	PSEUDO_DIFFERENTIAL	Two single ended clock in phase opposition
NUMBER_OF_DQ	4–8	Number of DQ bits
EXPOSE_DQ_RAW_DATA	0–1	When 1, this adds I*_DQ_RAW_DATA output to the fabric. Direct connection of DQ input to the fabric bypassing the PHY. It can be used to sample input data in 1-1-1 mode for xSPI.
FABRIC_CLOCK_RATIO	FULL	In full rate, the fabric run at the memory frequency.
CCC_PLL_MULTIPLIER	1–40	Multiplication factor from the external reference clock to the memory frequency.
CCC_PLL_REF_FREQ	—	Resulting reference clock frequency

...........continued

Parameter	Range	Description
DQS_DELAY_TAP	1 - 255	Delay tap set on DQS for read. Libero default value of 70 is optimized for 200 MHz performance as long as PCB routing is closely matched. Value below 20 will fail in simulation.
INTERFACE_NB	1–8	Number of shared interface
RANK_NB	1–2	Number of rank per interface
I{INTERFACE#}R{RANK#}_{IO}_ENABLED	0–1	For all I/O except DQ/DQS: Expose the I/O and corresponding signals. It can be used to disable I/O that are optional in the protocol (RESET_N for instance) or that are shared between interfaces/ ranks. This parameter is set using the related I/O Sharing tab in the Configurator.
I{INTERFACE#}R{RANK#}_{IO}_IN_DQ_LANE	0–1	For all I/O except DQ/DQS: Indicates that, the corresponding I/O will be placed in the DQ Lane. P and R will fail if an I/O is placed in DQ/DQS lane if this parameter is not set. By default, we assume all I/O other than DQ/DQS outside of the DQ/DQS lane. This parameter is set using the related I/O Placement tab in the Configurator.

The I/O Sharing tab sets the (I{INTERFACE#}R{RANK#}_{IO}_ENABLED parameter for CK, CS_N, INT_N, and RESET_N/RSTO pins allowing the user to deselect a port of the interface. Once set, that port would not be in the generated netlist. This is used to deselect the not used in the design or a port that is tied-off (that is, INT# input of HypeRAM), or for multi-rank deselect ports that are shared between ranks.

Figure 3-53. I/O Sharing Configurator Tab



⊟ **Deselect pins that are either shared or not used for rank 0**

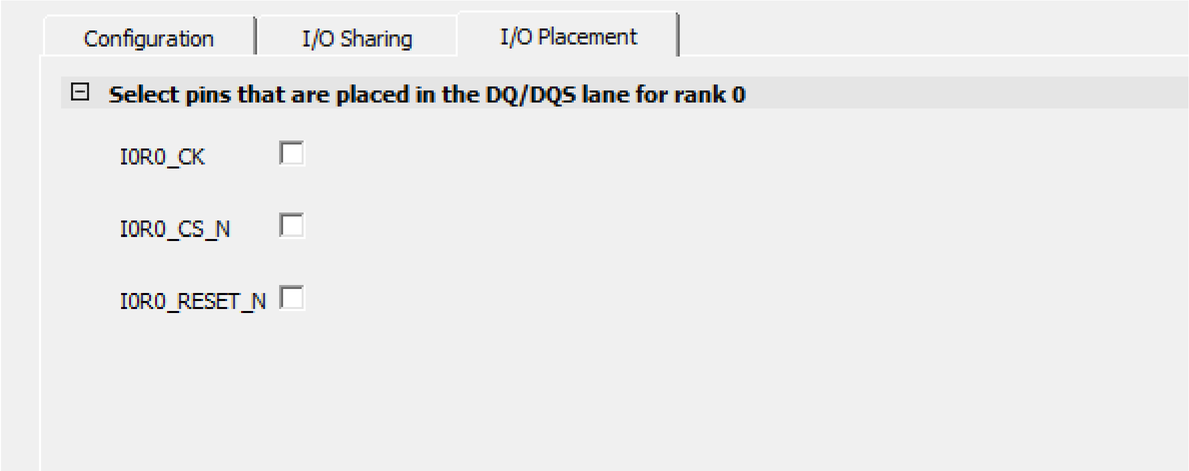
I0R0_CK ☑

I0R0_CS_N ☑

I0R0_RESET_N ☑

The I{INTERFACE#}R{RANK#}_{IO}_IN_DQ_LANE parameter is set with the I/O Placement tab in the configurator. It is used to indicate that a CK, CS_N, and RESET ports are in the DQ lane enabling Libero to correctly optimize the connections.

Figure 3-54. I/O Placement Configurator Tab



⊟ **Select pins that are placed in the DQ/DQS lane for rank 0**

IOR0_CK ☐

IOR0_CS_N ☐

IOR0_RESET_N ☐

Figure 3-55. Generated Octal DDR PHY SmartDesign Component

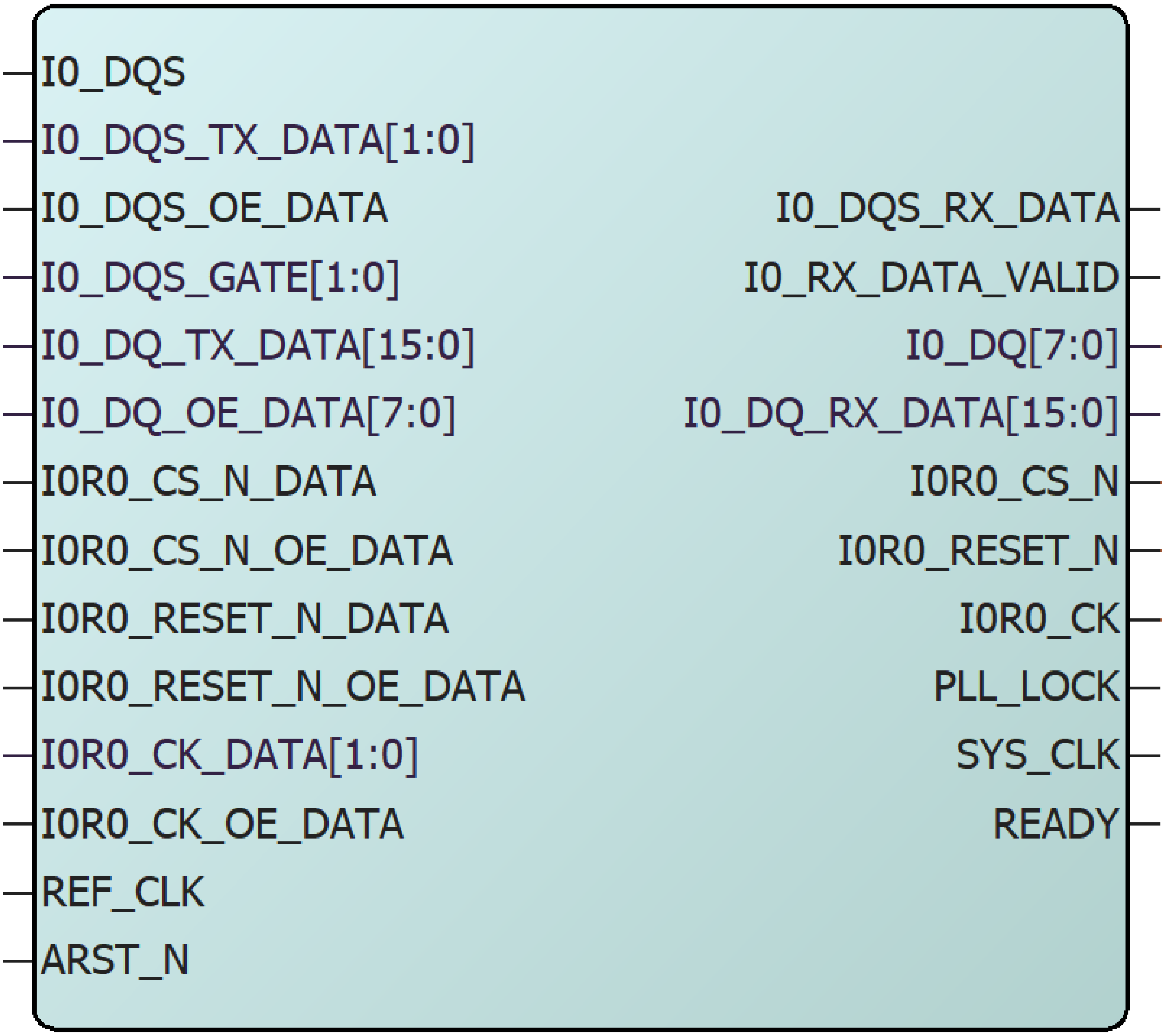


Table 3-25. Port List

Port	Width	Direction	Visibility	Description	Notes
General					
REF_CLK	1	IN	—	Reference clock to the CCC. It must be at CCC_PLL_REF_FREQ frequency.	—
ARST_N	1	IN	—	Asynchronous reset to the PHY. Note that, while ARST_N is asserted, the CCC inside the PHY is powerdown.	—
PLL_LOCK	1	OUT	—	Internal CCC PLL lock.	—
SYS_CLK	1	OUT	—	System clock. All input/output of the PHY from/to the fabric are synchronous to this clock (unless mentioned otherwise). SYS_CLK is generated by OUT0 of the CCC through CLKINT.	—
READY	1	OUT	—	When high indicates that the PHY initialization is complete and can receive commands.	—
Output				Clock can be any of: C, SCK, CK, CLK depending on protocol.	—
I{INTERFACE#}R{RANK#}_{CLOCK}	1	OUT	I{INTERFACE#}R{RANK#}_{CLOCK}_ENABLED== 1. This parameter is set using the related I/O Sharing tab in the Configurator.	Output pad to the memory, DDR.	CCC OUT1 through HS_IO_CLK, 90deg phase shift
I{INTERFACE#}R{RANK#}_{CLOCK}_N	1	OUT	I{INTERFACE#}R{RANK#}_{CLOCK}_ENABLED== 1. Protocol dependent. This parameter is set using the related IO Sharing tab in the Configurator.	Output pad N side to the memory, DDR.	CCC OUT1 through HS_IO_CLK, 90deg phase shift

...........continued

Port	Width	Direction	Visibility	Description	Notes
I{INTERFACE#}R{RANK#}_{CLOCK}_DATA	2	IN	I{INTERFACE#}R{RANK#}_{CLOCK}_ENABLED== 1. This parameter is set using the related IO Sharing tab in the Configurator.	Output data from the fabric to the I/O pad. Tie to 2'b01 to send a clock pattern. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}R{RANK#}_{CLOCK}_OE_DATA	2	IN	I{INTERFACE#}R{RANK#}_{CLOCK}_ENABLED == 1. This parameter is set using the related IO Sharing tab in the Configurator.	Output enable data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
Command				Output can be any of: S_N, CS_N, CE_N, RESET_N, ALE, CLE, WR_N, WP_N.	
I{INTERFACE#}R{RANK#}_{OUTPUT}	1	OUT	I{INTERFACE#}R{RANK#}_{OUTPUT}_ENABLED == 1. This parameter is set using the related I/O Sharing tab in the Configurator.	Output pad to the memory, SDR.	CCC OUT0 through HS_IO_CLK
I{INTERFACE#}R{RANK#}_{OUTPUT}_DATA	1	IN	I{INTERFACE#}R{RANK#}_{OUTPUT}_ENABLED== 1. This parameter is set using the related I/O Sharing tab in the Configurator.	Output data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}R{RANK#}_{OUTPUT}_OE_DATA	1	IN	I{INTERFACE#}R{RANK#}_{OUTPUT}_ENABLED== 1. This parameter is set using the related IO Sharing tab in the Configurator.	Output enable data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
Asynchronous Inputs					
I{INTERFACE#}R{RANK#}_{INPUT}	1	IN	I{INTERFACE#}R{RANK#}_{INPUT}_ENABLED == 1. This parameter is set using the related I/O Sharing tab in the Configurator.	Input pad from the memory.	ASYNC
I{INTERFACE#}R{RANK#}_{INPUT}_DATA	1	OUT	I{INTERFACE#}R{RANK#}_{INPUT}_ENABLED == 1. This parameter is set using the related I/O Sharing tab in the Configurator.	Input data to the fabric.	ASYNC
DQS					

...........continued

Port	Width	Direction	Visibility	Description	Notes
I{INTERFACE#}_{DQS}	1	INOUT	—	Inout pad from/to the memory, DDR.	CCC OUT0 through HS_IO_CLK
I{INTERFACE#}_{DQS}_N	1	INOUT	Protocol dependent	Inout pad N side from/to the memory, DDR.	CCC OUT0 through HS_IO_CLK
I{INTERFACE#}_{DQS}_TX_DATA	2	IN	protocol !=XSPI	Output data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}_{DQS}_OE_DATA	2	IN	—	Output enable data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}_{DQS}_RX_DATA	1	OUT	—	Input data from the I/O pad to the fabric. Asynchronous. It is not gated by the GATE signal. Note that, this signal will toggle when DQS is driven by the fabric.	ASYNC
I{INTERFACE#}_{DQS}_GATE	2	IN	—	DQS gating signal to the PHY. It is used to gate DQS going into the PHY RX FIFO. The gate must be low when the memory does not drive strobes (must be low whenever DQS is 'bZ or one DQS is driven by the FPGA or when DQS from the memory is not strobe. See timing section for the latency.	SYS_CLK
DQ					

...........continued

Port	Width	Direction	Visibility	Description	Notes
I{INTERFACE#}_DQ	4 or 8	INOUT	—	Inout pad to the memory, DDR INOUT except for XSPI where it is IN DQS can be any of: DQS, DS, and RWDS depending on protocol.	CCC OUT0 through HS_IO_CLK
I{INTERFACE#}_DQ _TX_DATA	8 or 16	IN	protocol !=XSPI	Output data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}_DQ _OE_DATA	4 or 8	IN	—	Output enable data from the fabric to the I/O pad. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}_DQ _RX_DATA	8 or 16	OUT	—	Input data from the I/O pad to the fabric. Synchronous. See the timing section for the latency through the PHY.	SYS_CLK
I{INTERFACE#}_RX _DATA_VALID	8 or 16	OUT	—	Indicate that received data are valid.	SYS_CLK
I{INTERFACE#}_DQ _RX_RAW_DATA	4 or 8	OUT	—	Input data from the I/O pad to the fabric. Asynchronous. This port will also toggle when DQ is driven by the fabric.	ASYNC

Figure 3-56. Command Timing

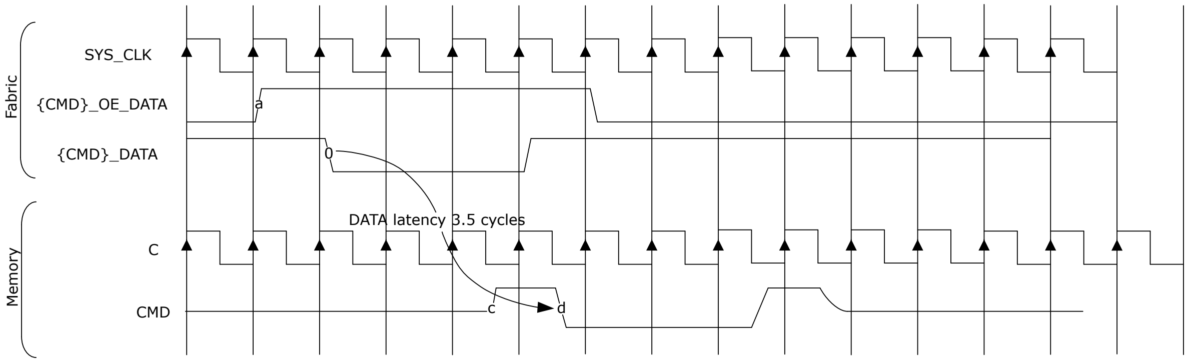
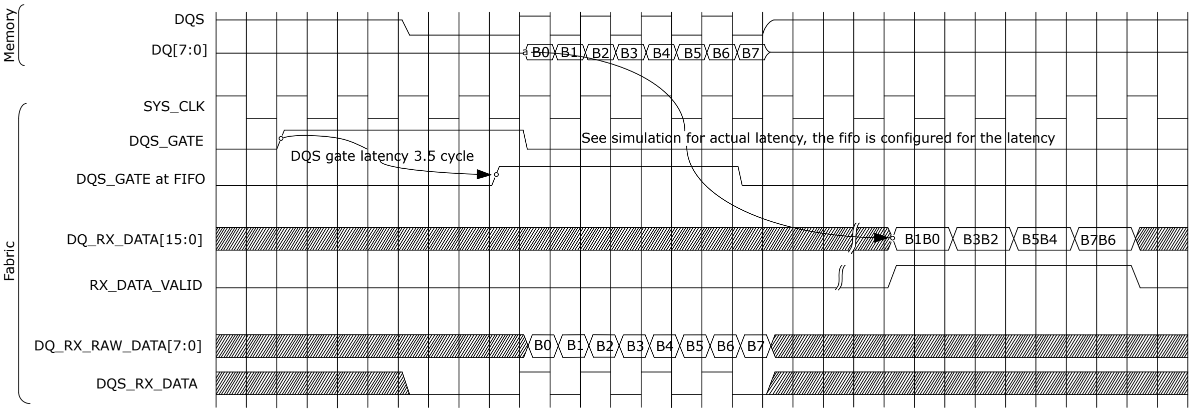


Figure 3-57. DQ/DQS Receive



Note: DQS preamble width differs depending on the protocol and configuration of the memory device.

3.8.2 Timing Constraints

The configurator generates a timing constraint for the internal CCC. Users can add or modify the I/O delay a timing constraint based on the part, and board layout. DQ input delay can be changed using **Libero SoC** > **I/O Editor**.

3.8.3 Place and Route Pin-Out Rules

Consider the following points during Place and Route:

- I/O Bridging is allowed.
- I/Os cannot share a lane with another independent IF.
- All DQ must be in the same lane.
- DQS/RWDS must be on the DQ lane on the P side I/O with the DQS function.
- The N side I/O with the DQS function must not be used. It is re-used internally to create the DQS Gate, and no other signal can be connected to pin.
- Addr/command/CLK can be placed anywhere. However, if they are placed in the DQ lane, the parameter I{INTERFACE#}R{RANK#}_{IO}_IN_DQ_LANE must be set to one.
- Asynchronous input can be placed anywhere.

Note: For a burst of multiple of two strobes cycle, any burst shorted than two cycles will not trigger RX_DATA_VALID and corrupt the receive FIFO. Any burst longer than two cycles but not multiple of two will keep RX_DATA_VALID high and corrupt the receive FIFO.

3.9 Implementation

This section describes how to use the DDR subsystem in the design. It includes information about selecting memory devices and configuring the subsystem to interface with the chosen devices.

3.9.1 Selecting the Memory Device

The Fabric DDR subsystem supports DDR3, DDR3L, DDR4, and LPDDR3 memory devices.

The memory that is used with the system depends on the application requirements. No single memory device can fulfill all system requirements; therefore, the trade-offs associated with each memory type must be considered for a balanced design.

Consider the following factors while selecting the memory device:

- Cost
- Bandwidth and speed
- Power consumption
- Bus width
- Latency
- Capacity

The following table lists the memory parameters for DDR3, DDR4, and LPDDR3 memory devices. Based on the system requirements (bandwidth, speed, bus width, density, and cost), the appropriate memory device is selected.

Table 3-26. Memory Selection Parameters

Parameter	DDR3	DDR3L	DDR4	LPDDR3
DDR clock	666	666	800	666
Data width (in bits)	16, 32, 64	16, 32, 64	16, 32, 64	16, 32
Burst length (in bytes)	8	8	8	8
Number of banks	8	8	8,16	8
Density	512 Mb–8192 Mb	512 Mb–8192 Mb	2 Gb–16 Gb	32 Gb
I/O standard	SSTL-15 Class I, II	SSTL-135 Class I, II	POD-12 Class I, II*	HSUL-12

Note: HSTL_12 IO standard is used for DDR4 address, command, and control signals.

3.9.1.1 DDR4 Optimization Guidelines

The following points provides guidance in selecting the proper speed bin memory component when using the PolarFire DDR PHY with the DDR4 Controller IP from Libero SoC:

1. The PolarFire DDR4 interface can operate up to 1600 Mbps. To achieve maximum efficiency at this speed, a 2400 Mbps speed bin DDR4 SDRAM must be used. Maximum efficiency is achieved with the tCCD_L = 4 (4 clock cycles) parameter setting. The following calculations can be used as examples to determine the memory component speed bin to be selected when running at a user's desired memory clock rate:
   - For DDR4-2400 memory components, tCCD_L must be ≥ 5 ns.
     - 4clock cycles * (1/800 MHz) = 5 ns, hence, an 800 MHz memory clock can be used to achieve 1600 Mbps data rate.
   - For DDR4-2133 memory components, tCCD_L must be ≥ 5.355 ns
     - 4clock cycles * (1/746 MHz) = 5.36 ns, hence, a 746 MHz memory clock can be used to achieve 1492 Mbps data rate.
   - For DDR4-1866 memory components, tCCD_L must be ≥ 5.355 ns
     - 4 clock cycles * (1/746 MHz) = 5.36 ns, hence, a 746 MHz memory clock can be used to achieve 1492 Mbps data rate.
2. tCCD_L delay values are obtained from Micron's 8Gb_DDR4_SDRAM datasheet.
3. The tCCD_L delay values are the same for DDR4-2133 and DDR4-1866.

4. In the above calculations, a slower memory clock can also be used.

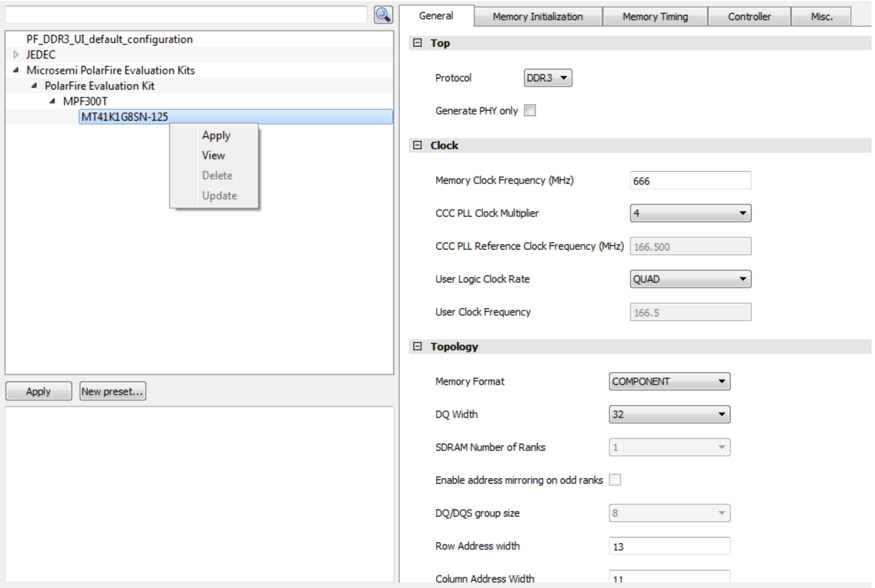
These examples may not apply if the embedded DDR PHY is interfaced to a proprietary DDR Controller IP. In this case, the user must develop appropriate set of optimized parameters.

3.9.2 Configuring the Fabric DDR Subsystem

The Fabric DDR subsystem macro (DDR3, DDR4, and LPDDR3) located in the Libero IP catalog must be instantiated in SmartDesign to access the DDR memory from the FPGA fabric through the subsystem. The DDR Configurator, shown in Figure 3-60, configures the DDR subsystem. It supports the following modes:

- Preset configuration—allows selection from a list of memory vendors and devices to preset all of the memory initialization and timing parameters into the DDR Configurator, as shown in Figure 3-58.
  Note: For DDR3, preset configuration is supported for 1333, 1600, 1866, and 2133 at 666.67 MHz (minimum tCK ≥ 1.5 ns). For DDR4, preset configuration is supported for 1600, 1866, 2133, 2400, 2666, 2933, and 3200 at 800 MHz (minimum tCK ≥ 1.25 ns).

Figure 3-58. DDR3 Configurator—Preset Configuration



PF_DDR3_UI_default_configuration
JEDEC
Microsemi PolarFire Evaluation Kits
PolarFire Evaluation Kit
MPF300T
MT41K1G8SN-125

⊟ Top

Protocol DDR3
Generate PHY only □

⊟ Clock

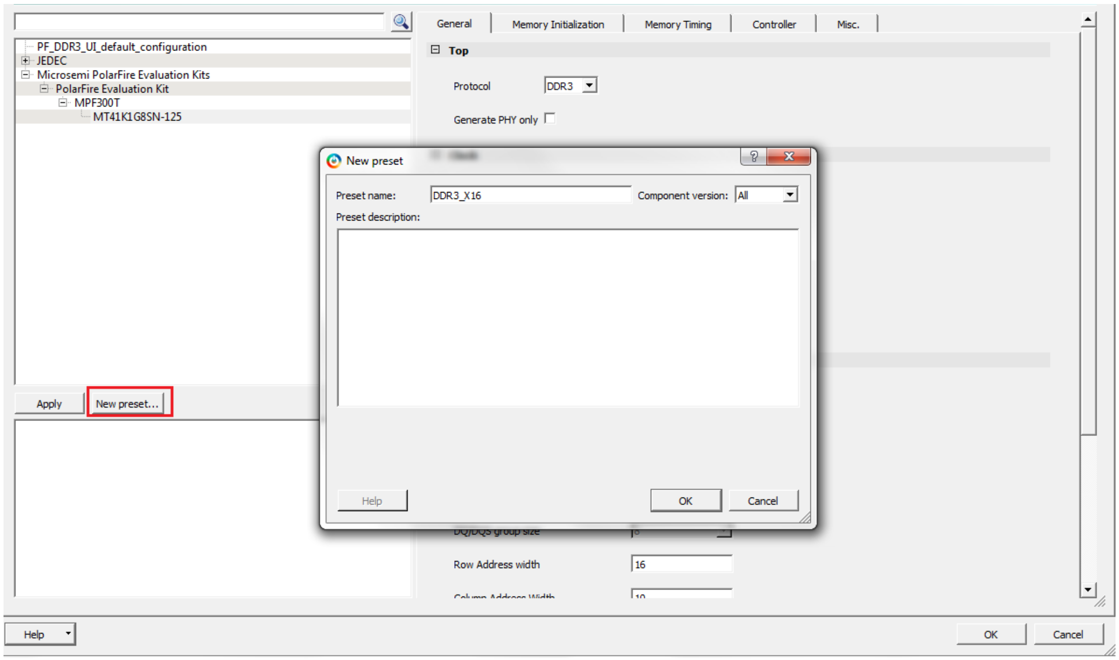
Memory Clock Frequency (MHz) 666
CCC PLL Clock Multiplier 4
CCC PLL Reference Clock Frequency (MHz) 166.500
User Logic Clock Rate QUAD
User Clock Frequency 166.5

⊟ Topology

Memory Format COMPONENT
DQ Width 32
SDRAM Number of Ranks 1
Enable address mirroring on odd ranks □
DQ/DQS group size 8
Row Address width 13
Column Address Width 11

- User configuration—allows manual configuration of all memory initialization and timing parameters. This can be saved as a preset configuration, as shown in Figure 3-59.
  Note: See the DDR vendor datasheets before configuring DDR parameters.

Figure 3-59. User-Defined Configuration



Protocol: DDR3

Generate PHY only □

New preset

Preset name: DDR3_X16

Component version: All

Preset description:

Row Address width: 16

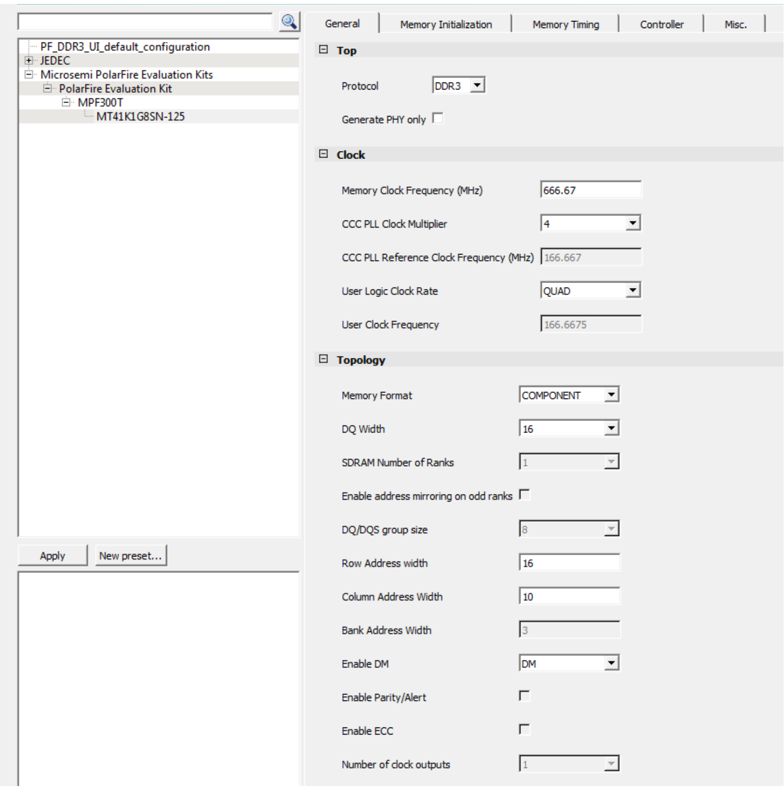
The following sections describe the configuration options available in the DDR Configurator.

3.9.2.1 General Options

The General tab contains top-level options for configuring the DDR subsystem, as shown in Figure 3-60. The fields available on this tab are:

- Clock—allows you to set the DDR memory clock frequency and user logic clock rate. The user clock frequency is automatically populated.
- Topology—allows you to set the topology of the memory system, including the address bits, bank configurations, DM modes, and ECC.

Figure 3-60. DDR Configurator—General Tab



Top

Protocol	DDR3
Generate PHY only	□

Clock

Memory Clock Frequency (MHz)	666.67
CCC PLL Clock Multiplier	4
CCC PLL Reference Clock Frequency (MHz)	166.667
User Logic Clock Rate	QUAD
User Clock Frequency	166.6675

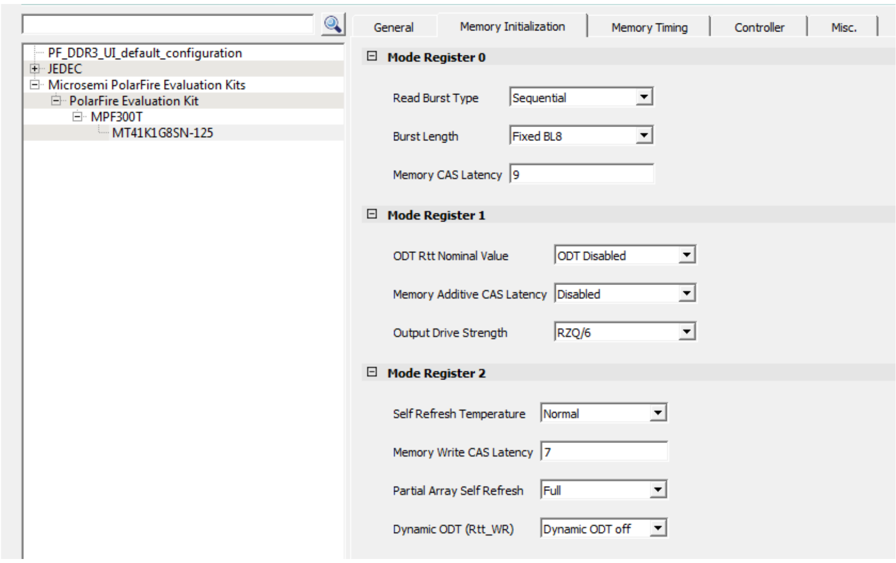
Topology

Memory Format	COMPONENT
DQ Width	16
SDRAM Number of Ranks	1
Enable address mirroring on odd ranks	□
DQ/DQS group size	8
Row Address width	16
Column Address Width	10
Bank Address Width	3
Enable DM	DM
Enable Parity/Alert	□
Enable ECC	□
Number of clock outputs	1

3.9.2.2 Memory Initialization

The Memory Initialization tab sets the DDR mode register configuration according to the JEDEC specification. In the Fabric DDR subsystem architecture, these parameters are passed to the DDR subsystem IP, which then performs the DDR initialization sequence and configures the mode registers.

Figure 3-61. DDR Configurator—Memory Initialization Tab



Mode Register 0

Read Burst Type: Sequential
Burst Length: Fixed BL8
Memory CAS Latency: 9

Mode Register 1

ODT Rtt Nominal Value: ODT Disabled
Memory Additive CAS Latency: Disabled
Output Drive Strength: RZQ/6

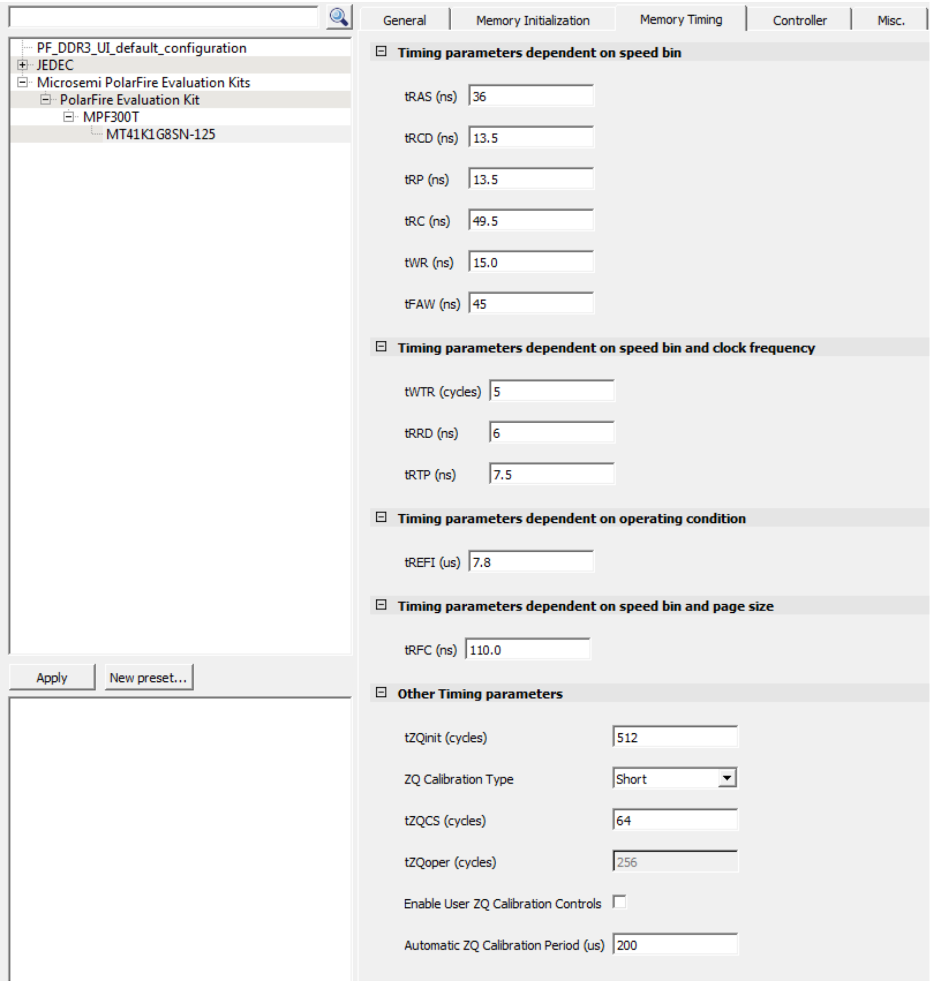
Mode Register 2

Self Refresh Temperature: Normal
Memory Write CAS Latency: 7
Partial Array Self Refresh: Full
Dynamic ODT (Rtt_WR): Dynamic ODT off

3.9.2.3 Memory Timing

The Memory Timing tab sets the timing parameters, which are then translated to the appropriate configuration values for the DDR subsystem IP, as shown in the following figure.

Figure 3-62. DDR Configurator—Memory Timing Tab



Timing parameters dependent on speed bin

tRAS (ns) 36

tRCD (ns) 13.5

tRP (ns) 13.5

tRC (ns) 49.5

tWR (ns) 15.0

tFAW (ns) 45

Timing parameters dependent on speed bin and clock frequency

tWTR (cycles) 5

tRRD (ns) 6

tRTP (ns) 7.5

Timing parameters dependent on operating condition

tREFI (us) 7.8

Timing parameters dependent on speed bin and page size

tRFC (ns) 110.0

Other Timing parameters

tZQinit (cycles) 512

ZQ Calibration Type Short

tZQCS (cycles) 64

tZQoper (cycles) 256

Enable User ZQ Calibration Controls □

Automatic ZQ Calibration Period (us) 200

3.9.2.4 Controller Options

DDR subsystem features are set using the Controller tab, as shown in Figure 3-63.

- Instance Select—enables selection of the DDR subsystem instance number that is used to unify the controller instance. PolarFire and PolarFire SoC devices support a maximum of six DDR subsystem instances.
- User Interface—options are AMBA AXI3/4 and native interface. The following table lists the supported AXI bus data widths with respect to the DQ widths.

Table 3-27. Supported AXI Bus Data Width

DQ Width	Supported AXI Data Bus Width
x16	64, 128
x32	64, 128, 256
x64	512

- Efficiency—provides control over memory refresh, precharge, and address ordering options. Queue depth can be set to 3 or 4 for DDR3 using the Command queue depth option on DDR Configurator > Controller tab.

- Data Bus Turnaround—provides additional bus turnaround time between different SDRAM ranks. The turnaround time is the number of clock cycles it takes to change the access from one rank to another rank for back-to-back memory accesses (read to read, read to write, write to write, and write to read). The AXI interface address is mapped based on the type of the Address Ordering selected in the PF_DDR3 configurator. For example, if Chip-Row-Bank-Col is selected, and if a row address width and column address width is configured for 13 and 11, the AXI address is mapped as shown in the following table.

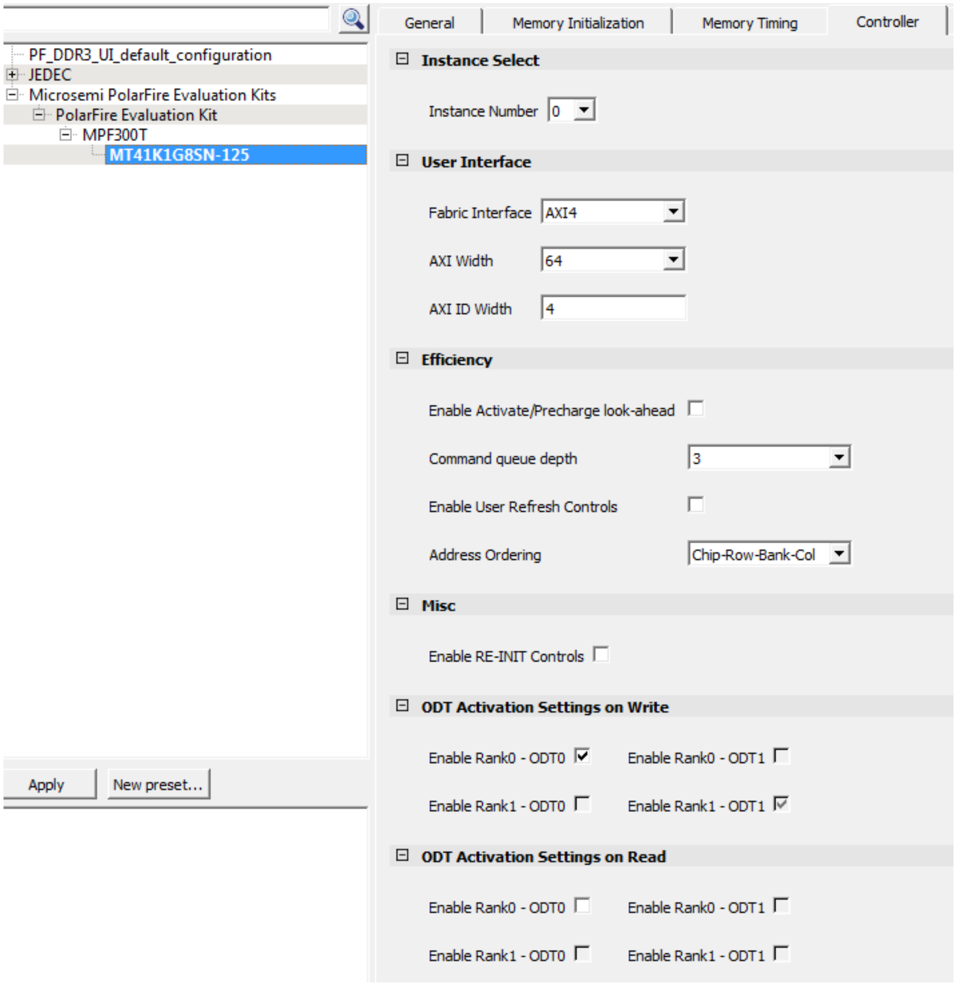
Table 3-28. AXI Address Mapping

AXI Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Column Address																				C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0		
Bank Address																	BA2	BA1	BA0													
Row Address				R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0																

Note: The address mapping shown in the preceding table is applicable to Native interface also.

- Misc—provides the following options:
  - Enable RE-INIT Controls—when selected, the configurator exposes the CTRLR_INIT signal. Initialization begins when a low-to-high transition is detected on this signal.
  - ODT activation settings on read/write—provides options to enable ODT on read and write operations.

Figure 3-63. DDR Configurator—Controller Options



⊟ Instance Select

Instance Number 0

⊟ User Interface

Fabric Interface AXI4

AXI Width 64

AXI ID Width 4

⊟ Efficiency

Enable Activate/Precharge look-ahead □

Command queue depth 3

Enable User Refresh Controls □

Address Ordering Chip-Row-Bank-Col

⊟ Misc

Enable RE-INIT Controls □

⊟ ODT Activation Settings on Write

Enable Rank0 - ODT0 ☑ Enable Rank0 - ODT1 □

Enable Rank1 - ODT0 □ Enable Rank1 - ODT1 ☑

⊟ ODT Activation Settings on Read

Enable Rank0 - ODT0 □ Enable Rank0 - ODT1 □

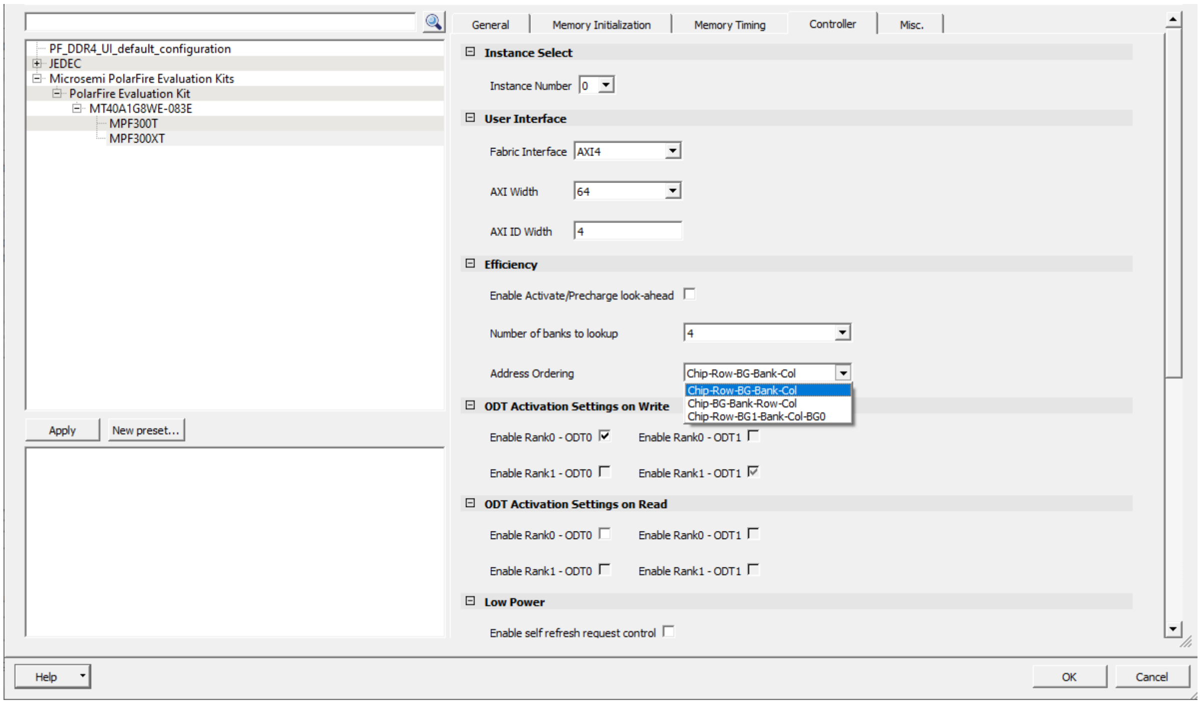
Enable Rank1 - ODT0 □ Enable Rank1 - ODT1 □

3.9.2.4.1 DDR4 Additional Controller Options

For DDR4, the Controller tab includes the following additional settings under Efficiency.

- The Address Ordering includes the additional setting Chip-Row-BG1-Bank-Col-BG0.
  This selection will map the LSBs of an address to the BG0 field of the memory. This results in alternating Bank Groups during a burst operation based on the tCCD_S timing parameter (see memory device datasheet for tCCD_S settings). The tCCD_S parameter is the timing parameter which results in a better efficient operation for the DDR IP Configurator in the DDR4 mode.

Figure 3-64. Address Ordering—Chip-Row-BG1-Bank-Col-BG0



PF_DDR4_UI_default_configuration
JEDEC
Microsemi PolarFire Evaluation Kits
PolarFire Evaluation Kit
MT40A1G8WE-083E
MPF300T
MPF300XT

Instance Select

Instance Number 0

User Interface

Fabric Interface AXI4

AXI Width 64

AXI ID Width 4

Efficiency

Enable Activate/Precharge look-ahead ☐

Number of banks to lookup 4

Address Ordering Chip-Row-BG-Bank-Col
- Chip-Row-BG-Bank-Col
- Chip-BG-Bank-Row-Col
- Chip-Row-BG1-Bank-Col-BG0

ODT Activation Settings on Write

Enable Rank0 - ODT0 ☑ Enable Rank0 - ODT1 ☐

Enable Rank1 - ODT0 ☐ Enable Rank1 - ODT1 ☑

ODT Activation Settings on Read

Enable Rank0 - ODT0 ☐ Enable Rank0 - ODT1 ☐

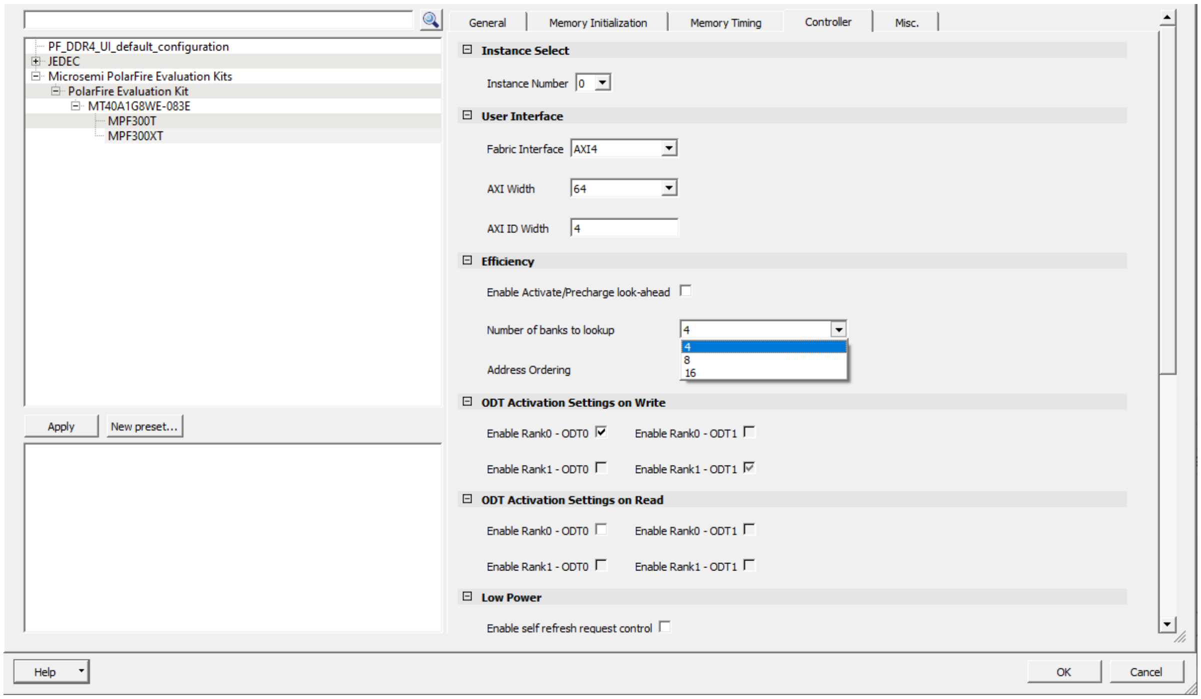
Enable Rank1 - ODT0 ☐ Enable Rank1 - ODT1 ☐

Low Power

Enable self refresh request control ☐

- The **Number of banks to look up** option includes additional selections for 8 and 16. The default selection is 4. This should be set to the number of banks in a Bank Group of the DDR4 memory device being used. Selecting a higher value is more challenging in terms if timing closure on the controller. A setting of 8 works for all but the most congested designs. A setting of 16 works for designs that have 15% timing margin or more.

Figure 3-65. Number of Banks to Look Up



PF_DDR4_UI_default_configuration
JEDEC
Microsemi PolarFire Evaluation Kits
PolarFire Evaluation Kit
MT40A1G8WE-083E
MPF300T
MPF300XT

Instance Select

Instance Number 0

User Interface

Fabric Interface AXI4

AXI Width 64

AXI ID Width 4

Efficiency

Enable Activate/Precharge look-ahead ☐

Number of banks to lookup 4
- 4
- 8
- 16

Address Ordering

ODT Activation Settings on Write

Enable Rank0 - ODT0 ☑ Enable Rank0 - ODT1 ☐

Enable Rank1 - ODT0 ☐ Enable Rank1 - ODT1 ☑

ODT Activation Settings on Read

Enable Rank0 - ODT0 ☐ Enable Rank0 - ODT1 ☐

Enable Rank1 - ODT0 ☐ Enable Rank1 - ODT1 ☐

Low Power

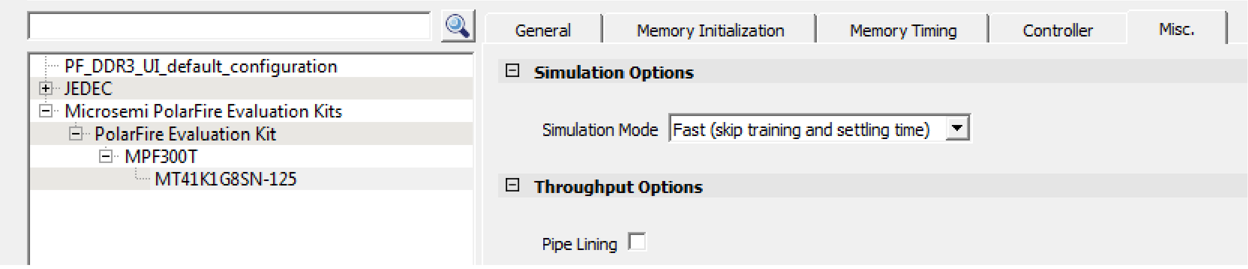
Enable self refresh request control ☐

3.9.2.5 Simulation Options

The **Misc** tab is used to select the simulation mode and expose the debug interface. The following simulation modes are available:

- **Fast (skip training and settling time)**: In this mode, the DDR subsystem skips the DDR training sequence and asserts CTRLR_READY without waiting for the settling time. CTRLR_READY is asserted within 8 µs.
- **Training (skip settling time)**: In this mode, the DDR subsystem performs the DDR training sequence but asserts CTRLR_READY without waiting for settling time. CTRLR_READY is asserted within 240 µs.
- **Full (training and settling time)**: In this mode, the DDR subsystem performs the DDR training sequence and also waits for settling time before asserting CTRLR_READY. In this mode, CTRLR_READY is asserted within 10 ms.
- **Pipe Lining**: Adds pipeline registers to Training IP (TIP) write and read data path to improve static timing closure.

Figure 3-66. Misc Tab



Simulation Mode: Fast (skip training and settling time)

Pipe Lining ☐

3.9.3 Simulating the DDR Subsystem

Libero SoC provides a simulation model for the DDR subsystem, which includes a training sequence (three simulation modes are supported as mentioned in 3.9.2.5 Simulation Options). For simulating the DDR subsystem, the DDR DRAM interface must be connected to a third-party DDR memory simulation model in a testbench. Memory vendors such as Micron, Samsung, and Hynix provide downloadable, JEDEC-compliant simulation models for memory devices. For information about setting up and running the simulation, see 3.10.1 Accessing DDR Subsystem Through AXI4 Interface.

Note: SmartDesign Testbench does not support instantiation of SystemVerilog testbenches for DDR models. To use SystemVerilog, create an HDL testbench instead of a SmartDesign testbench, and select **System Verilog** in Libero Project settings.

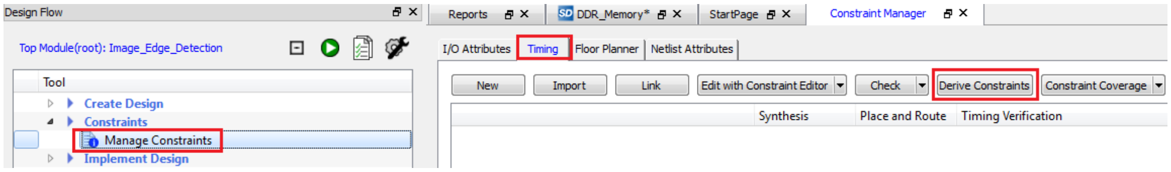
3.9.4 Design Constraints

This section describes physical constraints and timing constraints of the DDR subsystem.

3.9.4.1 Timing Constraints

The REF_CLK, HS_IO_CLK, HS_IO_CLK_270, and SYS_CLK clocks generated using the dedicated PLL require timing constraints for synthesis, place and route, and timing verification. To generate these timing constraints, select the **Timing** tab in **Constraint Manager**, and click **Derive Constraints** as shown in the following figure.

Figure 3-67. Constraint Manager—Derive Constraints Tab



When prompted, click **Yes** to apply the derived constraints for synthesis, place and route, and timing verification.

3.9.4.2 Physical Constraints

DDR memory controllers can be placed only in the pre-defined locations that are optimized for maximum performance. The maximum width of the DDR memory varies based on die/package combination and placement. For information about the DDR placement, see the respective PPATs as follows:

- PolarFire Package Pin Assignment Tables (PPAT)

- PolarFire SoC Package Pin Assignment Tables (PPAT)

The DDR subsystem location is selected using the physical design constraint (PDC) or the I/O Editor. Libero assigns the I/Os based on the selection of the DDR memory location. Before placing the DDR subsystem to the required I/O Bank location, the VDDI and VREF of that I/O Bank must be set to the appropriate value. In I/O Editor, the VDDI and VREF are set using the I/O Bank Settings option.

3.9.4.2.1 Using PDC

To select the DDR subsystem location, add the following PDC constraint to the floor planner constraints:

```
set_location -inst_name <memory component inst name> -location <edge>_<anchor>
```

For example, set_location –inst_name {DDR3_TOP/DDR3_0} –location {NORTH_NE}.

The number of I/Os per bank varies from package to package; therefore, the maximum DDR memory width supported by each bank is not same for all PolarFire and PolarFire SoC FPGA packages. For more information, see the respective user guides as follows:

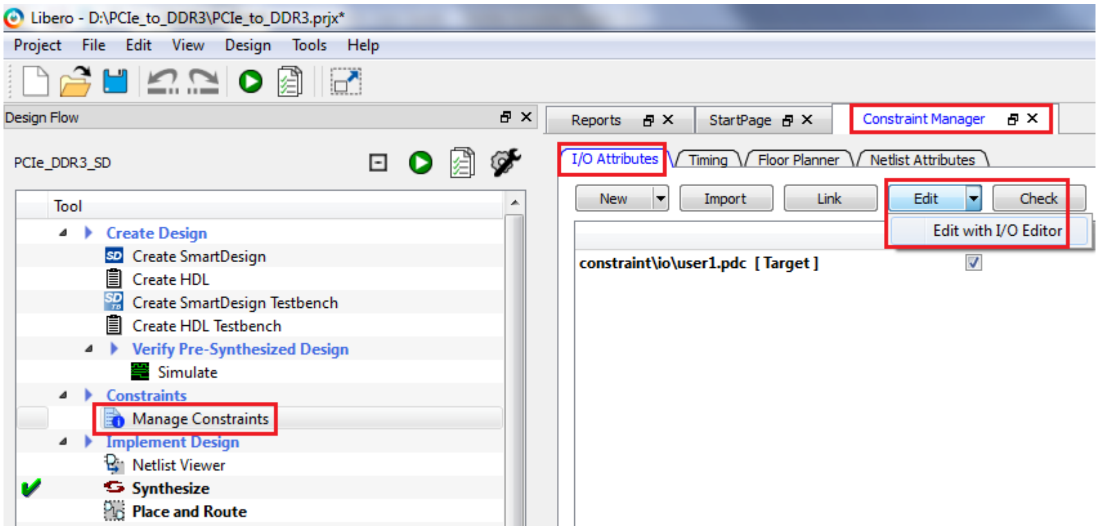
- UG0722: PolarFire FPGA Packaging and Pin Descriptions User Guide
- UG0902: PolarFire SoC FPGA Packaging and Pin Descriptions User Guide

3.9.4.2.2 Using I/O Editor

To select the DDR subsystem location, follow these steps:

1. Click **Design Flow** > **Constraint Manager** > **I/O Attributes** > **Edit with I/O Editor**.

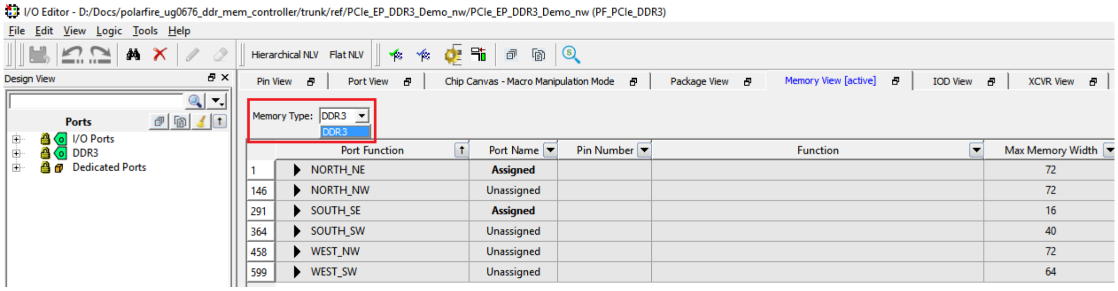
Figure 3-68. I/O Editor



constraint\io\user1.pdc [Target] ☑

2. Select the **Memory View** tab in **I/O Editor**. Select the appropriate **Memory Type** to get the valid locations for the memory type.

Figure 3-69. I/O Editor—Memory View



I/O Editor - D:/Docs/polarfire_ug0676_ddr_mem_controller/trunk/ref/PCIe_EP_DDR3_Demo_nw/PCIe_EP_DDR3_Demo_nw (PF_PCIe_DDR3)

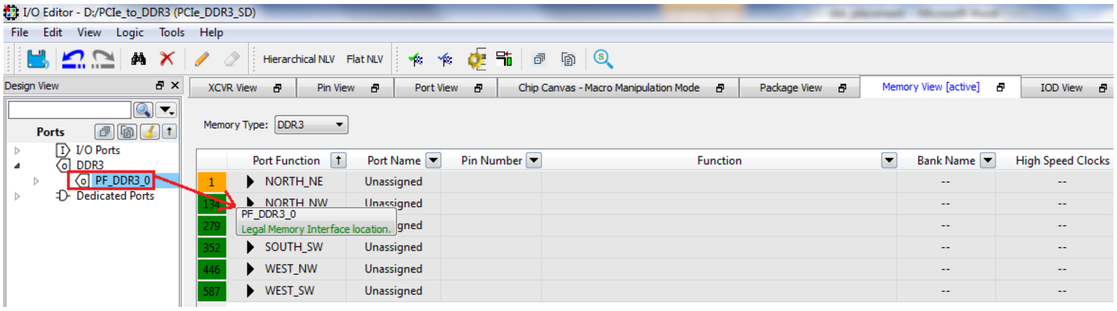
Memory Type: DDR3

	Port Function	Port Name	Pin Number	Function	Max Memory Width
1	NORTH_NE	**Assigned**			72
146	NORTH_NW	Unassigned			72
291	SOUTH_SE	**Assigned**			16
364	SOUTH_SW	Unassigned			40
458	WEST_NW	Unassigned			72
599	WEST_SW	Unassigned			64

Note: The difference between DDR3 and DDR3L is the I/O standard, SSTL15 and SSTL135 I/O. I/O editor allows to select SSTL15I, SSTL15II, SSTL135I, SSTL135II I/O standards. User needs to select SSTL135 I/Os in the I/O editor for DDR3L configuration.

3. Select the appropriate DDR instance from the **Design View** window, and drag it onto **Port Function**. For the selected DDR instance, the I/O Editor shows the valid locations in green.

Figure 3-70. I/O Editor—Valid DDR Locations



I/O Editor - D:/PCIe_to_DDR3 (PCIe_DDR3_SD)

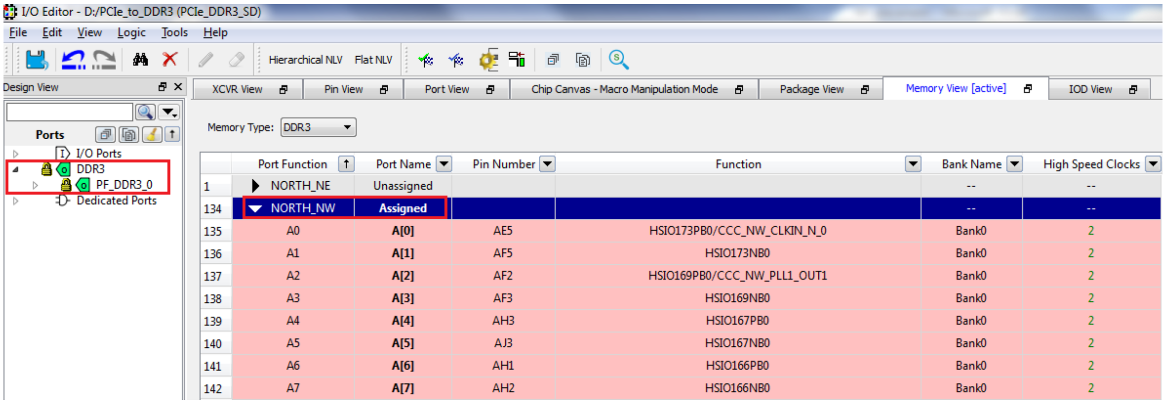
Memory Type: DDR3

	Port Function	Port Name	Pin Number	Function	Bank Name	High Speed Clocks
1	NORTH_NE	Unassigned			--	--
134	NORTH_NW	Unassigned			--	--
279		gned			--	--
352	SOUTH_SW	Unassigned			--	--
446	WEST_NW	Unassigned			--	--
587	WEST_SW	Unassigned			--	--

PF_DDR3_0
Legal Memory Interface location.

In **I/O Editor**, **Design View** window, the DDR instances that are assigned and locked are shown in green. **I/O Editor** does not assign a location if the selected location is unavailable, has insufficient resources or is impacted by any other issue. If an issue is detected, **I/O Editor** displays an error.

Figure 3-71. I/O Editor—Assigned DDR Location



I/O Editor - D:/PCIe_to_DDR3 (PCIe_DDR3_SD)

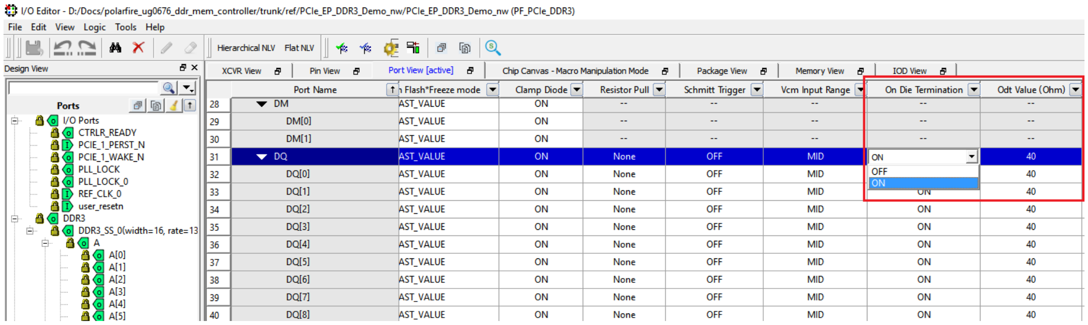
Memory Type: DDR3

	Port Function	Port Name	Pin Number	Function	Bank Name	High Speed Clocks
1	NORTH_NE	Unassigned			--	--
134	NORTH_NW	**Assigned**			--	--
135	A0	**A[0]**	AE5	HSIO173PB0/CCC_NW_CLKIN_N_0	Bank0	2
136	A1	**A[1]**	AF5	HSIO173NB0	Bank0	2
137	A2	**A[2]**	AF2	HSIO169PB0/CCC_NW_PLL1_OUT1	Bank0	2
138	A3	**A[3]**	AF3	HSIO169NB0	Bank0	2
139	A4	**A[4]**	AH3	HSIO167PB0	Bank0	2
140	A5	**A[5]**	AJ3	HSIO167NB0	Bank0	2
141	A6	**A[6]**	AH1	HSIO166PB0	Bank0	2
142	A7	**A[7]**	AH2	HSIO166NB0	Bank0	2

To turn ODT on for a DQ and DQS, select the Port View tab, select the signal, and select ON from the On Die Termination list, as shown in the following figure.

Note: 60 Ω ODT is the recommended ODT setting for DDR4 DIMM/on-board DDR4 component. 30 Ω ODT is the recommended ODT setting for on-board DDR3 component. 60 Ω ODT is the recommended ODT setting for DDR3 DIMM.

Figure 3-72. Setting ODT



	Port Name	n Flash*Freeze mode	Clamp Diode	Resistor Pull	Schmitt Trigger	Vcm Input Range	On Die Termination	Odt Value (Ohm)
28	DM	AST_VALUE	ON	--	--	--	--	--
29	DM[0]	AST_VALUE	ON	--	--	--	--	--
30	DM[1]	AST_VALUE	ON	--	--	--	--	--
31	DQ	AST_VALUE	ON	None	OFF	MID	ON	40
32	DQ[0]	AST_VALUE	ON	None	OFF	MID	OFF	40
33	DQ[1]	AST_VALUE	ON	None	OFF	MID	ON	40
34	DQ[2]	AST_VALUE	ON	None	OFF	MID	ON	40
35	DQ[3]	AST_VALUE	ON	None	OFF	MID	ON	40
36	DQ[4]	AST_VALUE	ON	None	OFF	MID	ON	40
37	DQ[5]	AST_VALUE	ON	None	OFF	MID	ON	40
38	DQ[6]	AST_VALUE	ON	None	OFF	MID	ON	40
39	DQ[7]	AST_VALUE	ON	None	OFF	MID	ON	40
40	DQ[8]	AST_VALUE	ON	None	OFF	MID	ON	40

Note: If VREF is not specified, the internal VREF is used. When external VREF is used, any I/O pin available on the same bank can be selected as VREF using the I/O Editor. If the DDR Subsystem uses two banks, two external VREF pins are required (one per each bank). Microchip recommends using the internal VREF.

If there is DDR, QDR, and DLL in a design, be aware of the following DLL placement rules:

- If DDR/QDR is placed in NORTH pinouts, the DLL cannot be placed in the location 2,377 (DLL0_NW)
- If DDR/QDR is placed in SOUTH pinouts, the DLL cannot be placed in the location 2462,5 (DLL0_SE)
- If DDR/QDR is placed in WEST pinouts, the DLL cannot be placed in the location 2,5 (DLL0_SW)

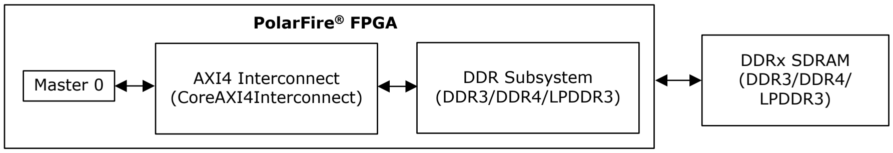
3.10 Functional Examples

This section describes how to use the DDR subsystem in the design in both AXI4 and native interface modes.

3.10.1 Accessing DDR Subsystem Through AXI4 Interface

In AXI4 interface mode, the AXI4 master implemented in the FPGA fabric accesses the DDR memory through the AXI4 interconnect and the DDR subsystem, as shown in the following figure.

Figure 3-73. DDR Subsystem Accessed Through AXI4 Interface



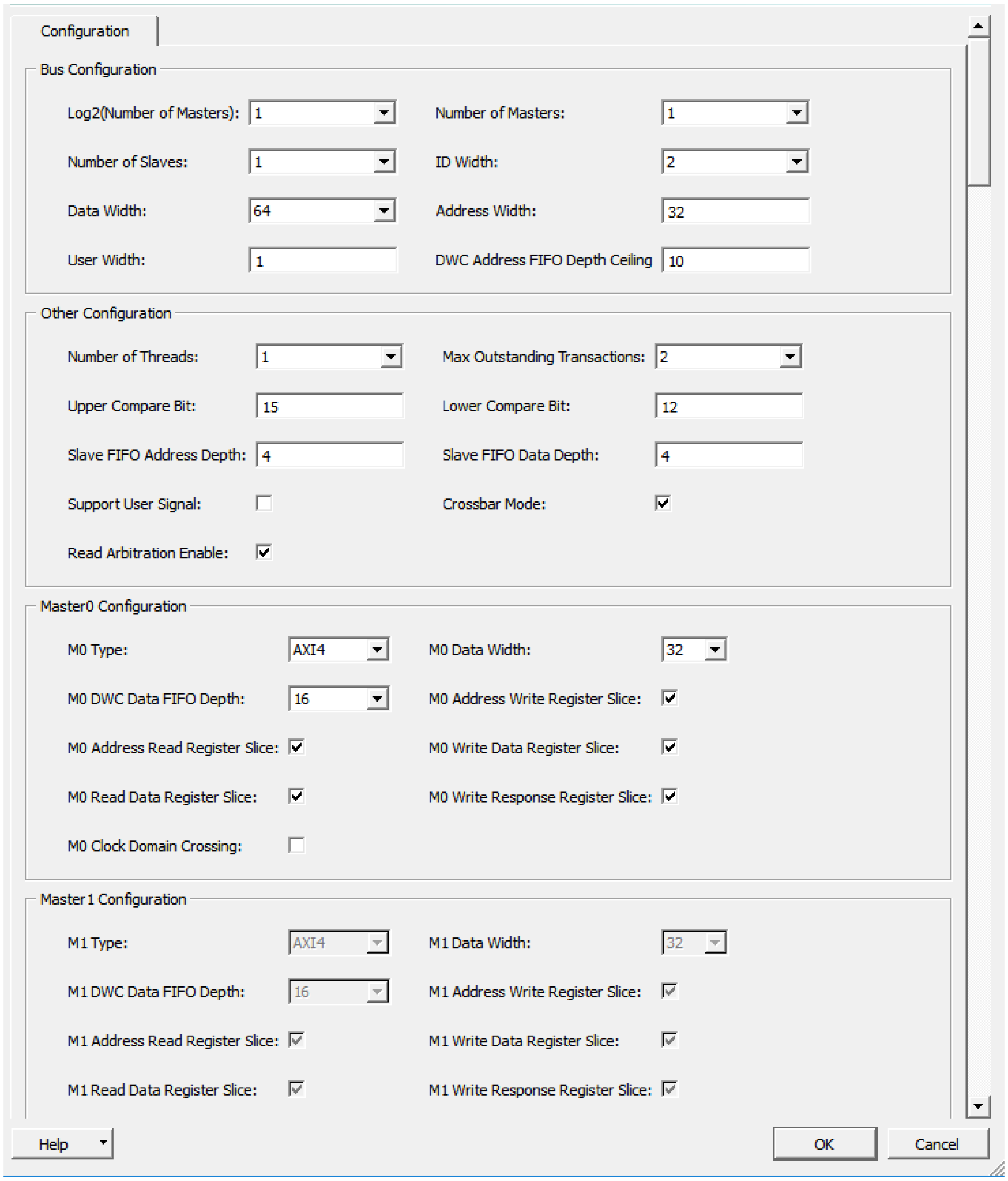
AXI4 interconnect functions in pass-through mode for a single-master-and-slave configuration. The AXI4 interconnect IP (CoreAXI4Interconnect) is available for download from the Libero SoC IP catalog.

After successful DDR initialization, the AXI master initiates reads from and writes to the DDR memory. The following steps describe how to create a design to access the DDR3 memory from the AXI master in the FPGA fabric:

1. Create a SmartDesign, and instantiate the DDR3 macro.
2. Configure the DDR3 subsystem as described in 3.9 Implementation, or apply the preset configuration (if any), as shown in Figure 3-58. The design shown in this figure is created to access the DDR3 memory with a 32-bit data width through the AXI4 interface.
3. Instantiate the user AXI4 master logic in the SmartDesign canvas. Ensure that the AXI master logic accesses the DDR3 subsystem only after CTRLR_READY is high.

4. Instantiate the CoreAXI4Interconnect IP, and configure single master and slave, as shown in the following figure.

Figure 3-74. CoreAXI4Interconnect IP Configurator



Bus Configuration

Log2(Number of Masters): 1
Number of Masters: 1
Number of Slaves: 1
ID Width: 2
Data Width: 64
Address Width: 32
User Width: 1
DWC Address FIFO Depth Ceiling 10

Other Configuration

Number of Threads: 1
Max Outstanding Transactions: 2
Upper Compare Bit: 15
Lower Compare Bit: 12
Slave FIFO Address Depth: 4
Slave FIFO Data Depth: 4
Support User Signal: ☐
Crossbar Mode: ☑
Read Arbitration Enable: ☑

Master0 Configuration

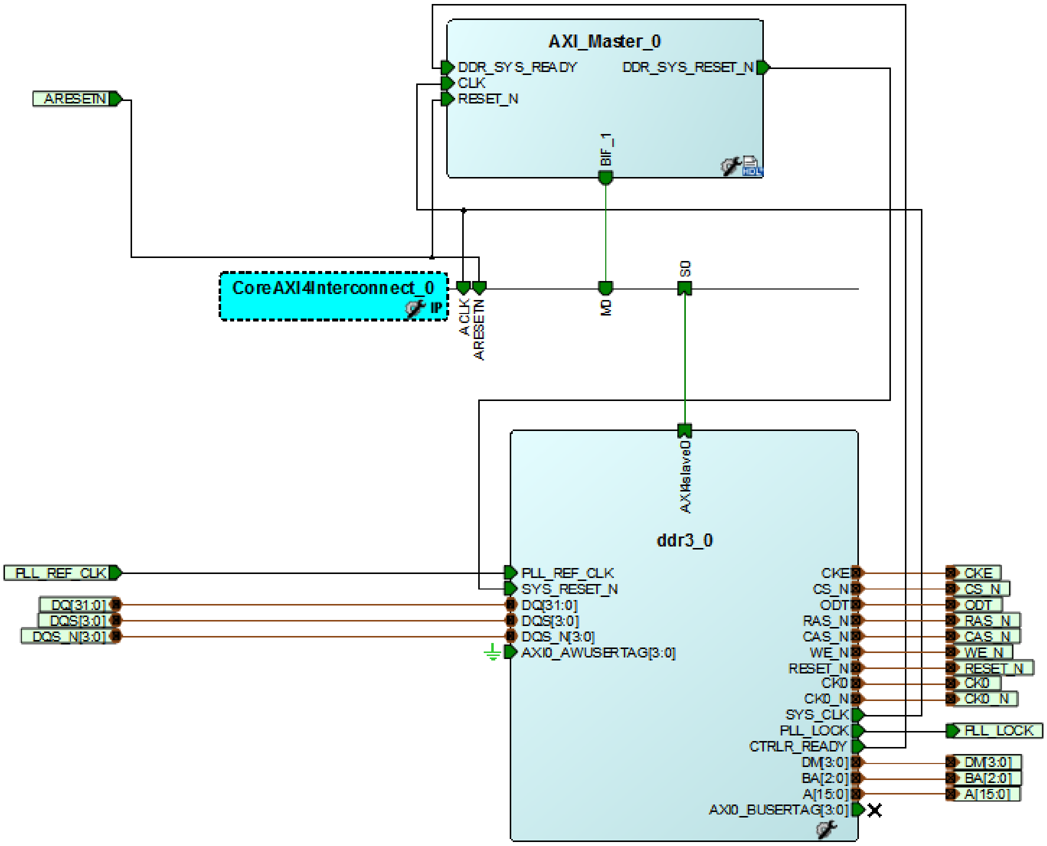
M0 Type: AXI4
M0 Data Width: 32
M0 DWC Data FIFO Depth: 16
M0 Address Write Register Slice: ☑
M0 Address Read Register Slice: ☑
M0 Write Data Register Slice: ☑
M0 Read Data Register Slice: ☑
M0 Write Response Register Slice: ☑
M0 Clock Domain Crossing: ☐

Master1 Configuration

M1 Type: AXI4
M1 Data Width: 32
M1 DWC Data FIFO Depth: 16
M1 Address Write Register Slice: ☑
M1 Address Read Register Slice: ☑
M1 Write Data Register Slice: ☑
M1 Read Data Register Slice: ☑
M1 Write Response Register Slice: ☑

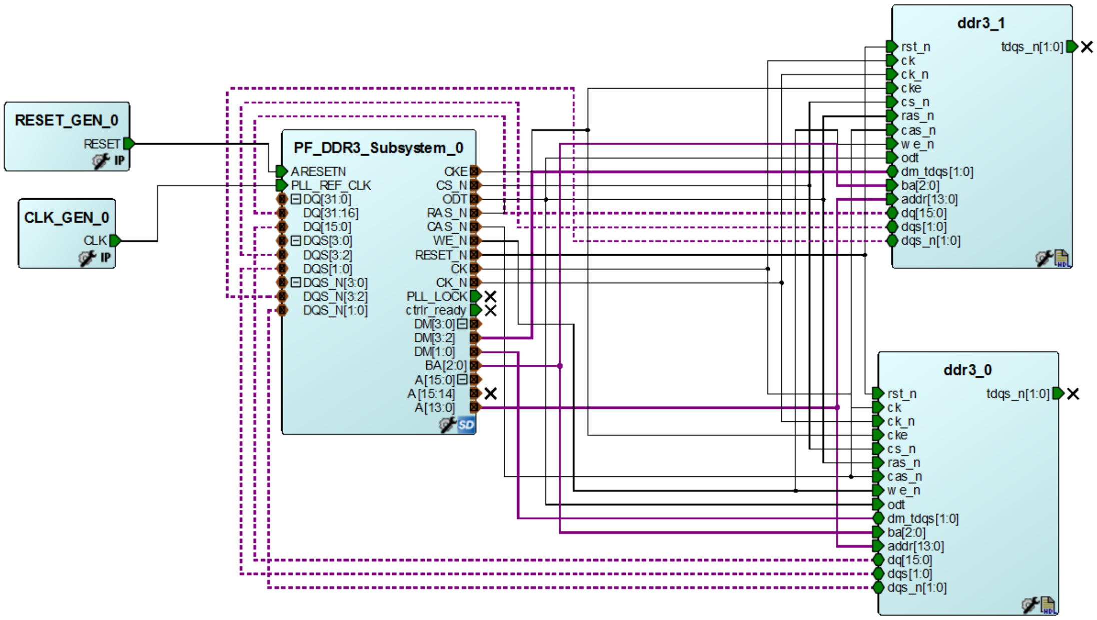
5. The AXI master and CoreAXI4Interconnect IP clocks must be driven from the SYS_CLK clock of the DDR3 subsystem.
6. In the SmartDesign canvas, connect the blocks as shown in the following figure.

Figure 3-75. SmartDesign Connection—AXI



7. Create a new SmartDesign testbench to simulate the design.
8. Instantiate the top-level design component and the DDR memory simulation models.
9. Configure CLK_GEN to generate the PLL reference clock, and connect to PLL_REF_CLK.
10. Connect the blocks in SmartDesign testbench, as shown in the following figure.

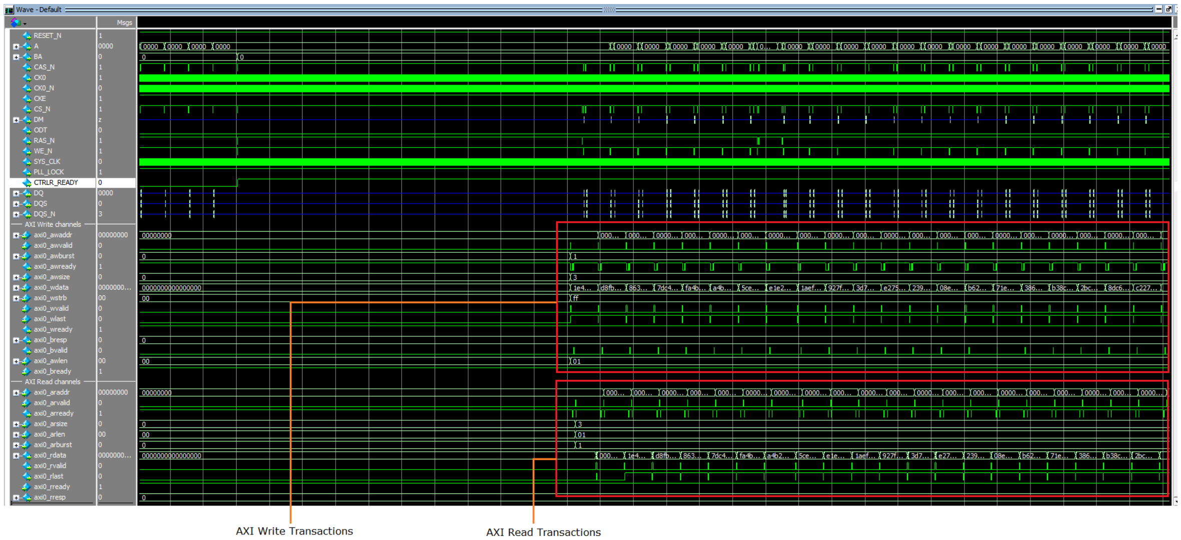
Figure 3-76. SmartDesign Testbench



11. Select **Simulate** from Libero **Design Flow** -> **Verify Pre-synthesized Design**. The DDR subsystem initializes the memory model, and sequence complete messages are displayed on the transcript window.

    The following figure shows the AXI read and write transactions and the corresponding SDRAM transactions.

Figure 3-77. AXI Read and Write Transactions

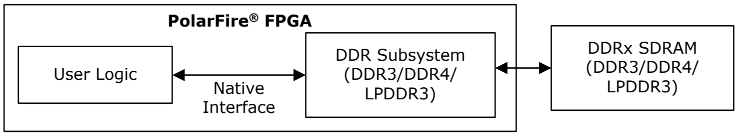


	Msgs
RESET_N	1
A	0000
BA	0
CAS_N	1
CK0	1
CK0_N	0
CKE	1
CS_N	1
DM	z
ODT	0
RAS_N	1
WE_N	1
SYS_CLK	0
PLL_LOCK	1
CTRLR_READY	0
DQ	0000
DQS	0
DQS_N	3
AXI Write channels	
axi0_awaddr	00000000
axi0_awvalid	0
axi0_awburst	0
axi0_awready	1
axi0_awsize	0
axi0_wdata	0000000...
axi0_wstrb	00
axi0_wvalid	0
axi0_wlast	0
axi0_wready	1
axi0_bresp	0
axi0_bvalid	0
axi0_awlen	00
axi0_bready	1
AXI Read channels	
axi0_araddr	00000000
axi0_arvalid	0
axi0_arready	1
axi0_arsize	0
axi0_arlen	00
axi0_arburst	0
axi0_rdata	0000000...
axi0_rvalid	0
axi0_rlast	0
axi0_rready	1
axi0_rresp	0

3.10.2 Accessing DDR Subsystem Through Native Interface

The DDR subsystem can be used to access SDRAMs directly, as shown in the following figure.

Figure 3-78. DDR Subsystem Accessed Through Native Interface

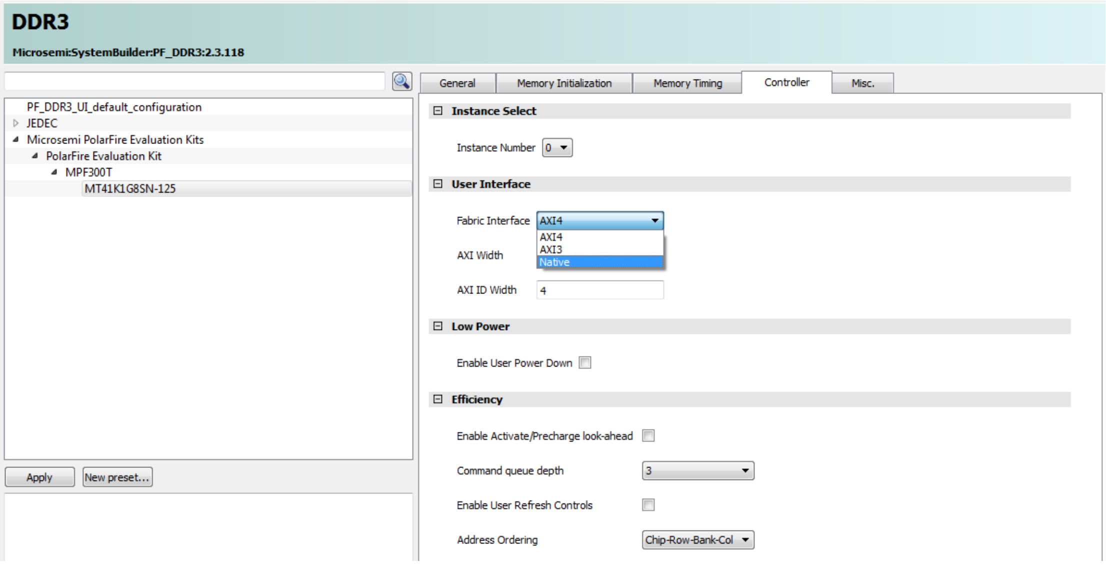

PolarFire® FPGA
User Logic
Native Interface
DDR Subsystem (DDR3/DDR4/ LPDDR3)
DDRx SDRAM (DDR3/DDR4/ LPDDR3)

User logic is connected directly to the DDR subsystem using the native interface.

After successful DDR initialization, the native interface master initiates reads from or writes to the DDR memory. The following steps describe how to create a design to access the DDR3 memory from the native interface master in the FPGA fabric:

1. Create a SmartDesign, and instantiate the DDR3 macro.
2. Configure the DDR3 subsystem as described in 3.9 Implementation, or apply the preset configuration (if any), and select the native interface, as shown in the following figure.
   In this example, the design is created to access the DDR3 memory with a 16-bit data width through the native interface.

   Figure 3-79. Native Interface Selection

   DDR3

   Microsemi:SystemBuilder:PF_DDR3:2.3.118

   PF_DDR3_UI_default_configuration
   JEDEC
   Microsemi PolarFire Evaluation Kits
   PolarFire Evaluation Kit
   MPF300T
   MT41K1G8SN-125

   Instance Select

   Instance Number: 0

   User Interface

   Fabric Interface: AXI4 (options: AXI4, AXI3, Native)

   AXI Width

   AXI ID Width: 4

   Low Power

   Enable User Power Down ☐

   Efficiency

   Enable Activate/Precharge look-ahead ☐

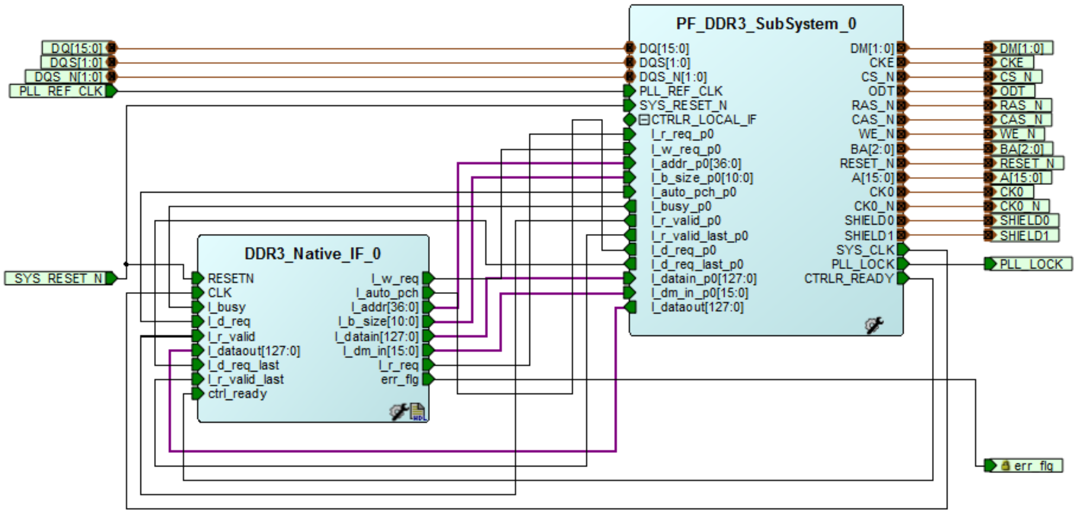
   Command queue depth: 3

   Enable User Refresh Controls ☐

   Address Ordering: Chip-Row-Bank-Col

3. Instantiate the user native interface master logic in the SmartDesign canvas. Ensure that the native interface master logic accesses the DDR3 subsystem only after CTRLR_READY is high.
4. In the SmartDesign canvas, connect the blocks, as shown in the following figure.

Figure 3-80. SmartDesign Connection—Native Interface

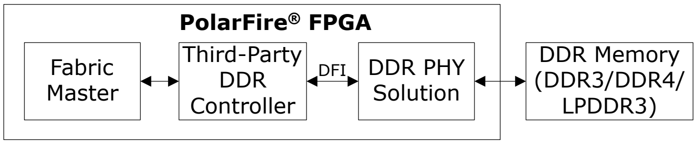


5. Follow steps 7–11 of 3.10.1 Accessing DDR Subsystem Through AXI4 Interface to simulate the design and see the read and write transactions.

3.10.3 Accessing DDR Memory Using a Third-Party DDR Controller and the DDR PHY-Only Solution

The DDR PHY-only solution can be used to access the SDRAMs using a third-party DDR controller or an open-source DDR controller. A fabric master accesses the DDR memory through a third-party DDR controller and the DDR PHY-only solution, as shown in the following figure.

Figure 3-81. DDR Memory Accessed Through Third-Party DDR Controller



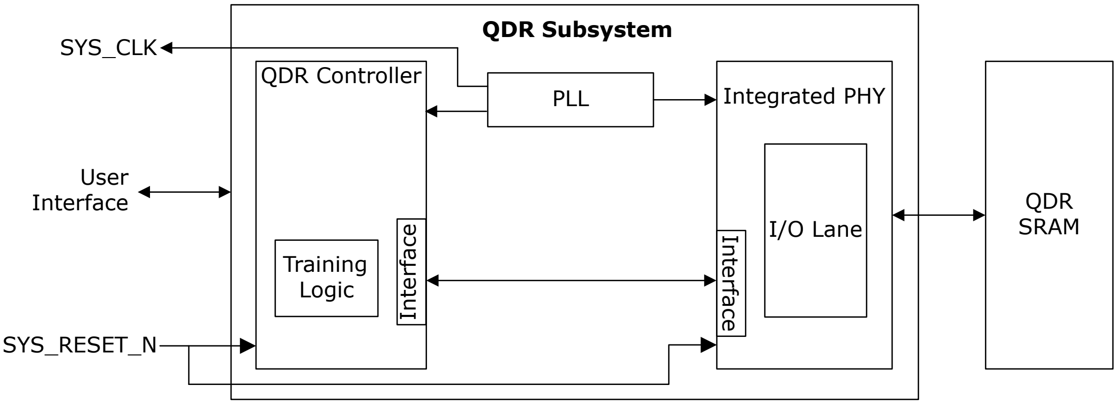
4. QDR Memory Controller

The PolarFire QDR subsystem addresses memory solution requirements for communication applications. The subsystem can be configured to support QDR II+ and QDR II+ Xtreme memory devices. The subsystem is intended for accessing QDR memories for applications that require high-speed data transfer. QDR SRAMs are a family of SRAMs with separate read and write channels, each operating at double date rate, optimized for high-performance communication applications (Example: Data Plane Memory of Routers performing functions such as packet buffering, statistics counting, and flow rate control).

The PolarFire QDR subsystem is made up of the following soft and hard blocks as shown in the following figure:

- QDRC (Soft Block)
- I/O Lane (Hard Block)
- Phase-Locked Loop (PLL) (Hard Block)

Figure 4-1. QDR Controller Block Diagram



4.1 Features

The QDR subsystem provides the following features:

- Integrated Hard PHY
- Supports QDR II+ and QDR II+ Xtreme Interface
  - Up to 1100 Mbps data rate (550 MHz clock)
  - Separate read and write channels, supporting concurrent transactions
  - Single address channel
  - Burst of 2 and burst of 4 support (configurable)
- Configurable QDR data width (9, 18, 36 bits)
- Configurable address width (18-21 bits)

4.2 Performance

The following table lists the performance of QDR families.

Table 4-1. Performance of QDR Families

Parameters		QDR II+	QDR II+ Xtreme
Frequency	2-Word Burst	333 MHz	400 MHz
	4-Word Burst	500 MHz	550 MHz
Density		18 / 36 / 72 / 144 Mb	36 / 72 Mb

4.2.1 Read Latency

The following table lists the QDR Read latency.

Table 4-2. Read Latency

Data Rate	Sys Clock Frequency (MHz)	User Interface Data Width	Memory Data Width	Latency in Cycles (User Interface)	Latency in ns
1100 Mbps	137.5	288	36	15 cycles	109.09
800 Mbps	100	288	36	14/15 cycles	140-150
500 Mbps	62.5	288	36	14 cycles	224

4.3 Functional Description

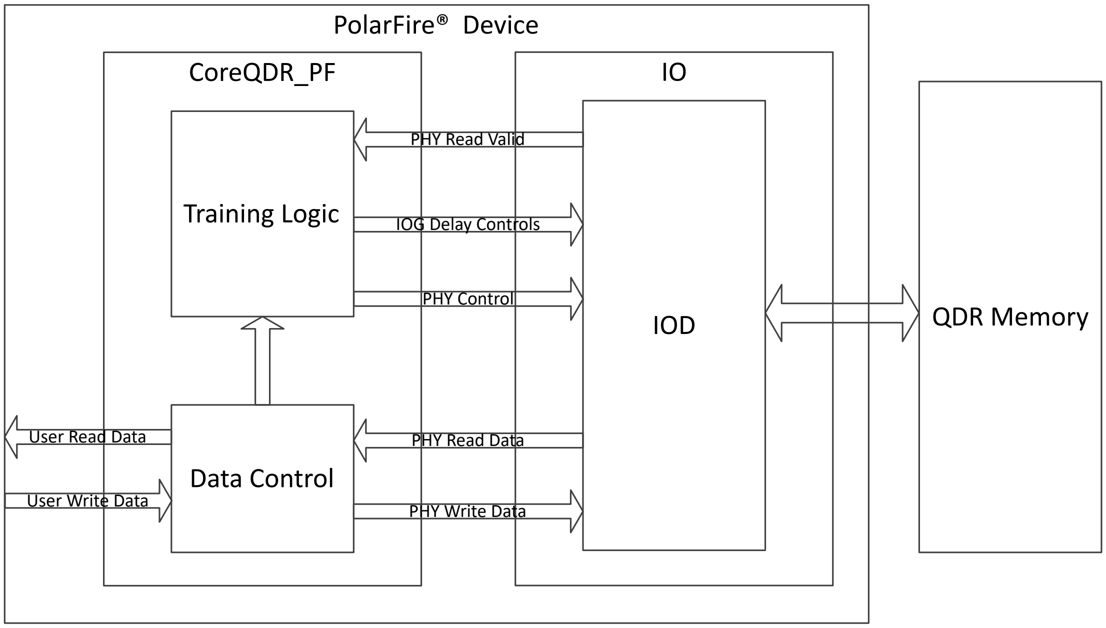
The QDR Subsystem translates the User Interface requests into command sequences required by QDR SRAM devices using the QDR Controller and PHY subsystem modules.

4.3.1 QDR Controller

QDR Controller is a Soft IP Core that consists of the following blocks:

- Data Control: Accepts the Write and Read commands from User Interface and convert them to PHY commands. It controls the PHY to generate QDR memory interface signals such that, the memory device look into data with the correct latency-with respect to Address and Command (WPS_N and RPS_N).
- Training Logic: Controls PHY Delay Lines & Deskew (due to board and internal delays) Input data, Input Clocks, and Input Valid Signals.

Figure 4-2. QDR Memory Controller



4.3.1.1 Training Logic

The training process is as follows:

1. CQP, Q delay controls are initialized with default values.

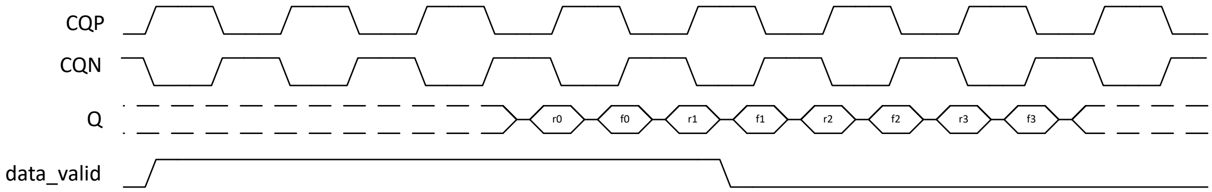
2. The test data patterns [0x96, 0x2d, 0x5a, 0xb4, 0x69, 0xd2, 0xa5, 0x4b] are written to Q.
3. The test data patterns are read at falling edges of CQP.
4. If all 8 test data patterns are read, the following conditions are verified for all data patterns:
   - If Q (data pattern) matches on falling edge of CQP. Q is moved to check the next data pattern.
   - If Q (data pattern) does not match, the training logic adjusts the IOD tap delay values as shown in Figure 4-4 and Figure 4-5.
   - If less than 8 patterns are matched, the training process is repeated from Step 2.
   - If Q_DELAY_LINE_OUT_OF_RANGE is asserted, TRAINING_ERROR and TRAINING_COMPLETE flags are asserted and then, the training process is repeated from Step 2.
5. Q is moved back to center of falling edge of CQP.
6. A sweep on CQP is performed to find the center of the rising edge data similar to step 3, 4 and 5. This covers all the Q bits per lane (9 bits per lane).
7. Step 2 is repeated for all lanes.

Note: Training is not shown on a per-Q basis, for brevity and clarity. CQN is generated in QDR IP.

Note: QDR training remains valid across PVT.

The timing relationships between CQP, CQN, data (Q), and data valid (before training begins) is shown in the following figure.

Figure 4-3. Data Capture before Training begins

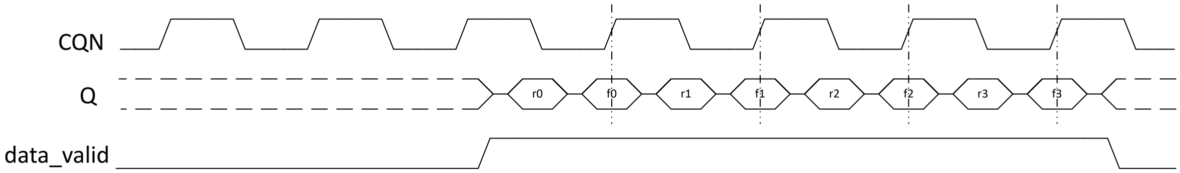


At first, the falling edge data alignment is done concurrently with the data_valid alignment. As a result, the data_valid properly frames the 8-bit burst coming from the QDR device and the CQN centers the falling edge data (f0, f1, f2, f3).

Note: CQN is not moved because it cannot be moved in the specified architecture; instead, Q is moved to align with CQN by performing write and read of 8 different patterns and selecting a match if and only if all patterns match. This is done on a per-Q basis, with the exception of data framing, which is only performed on Q<0>.

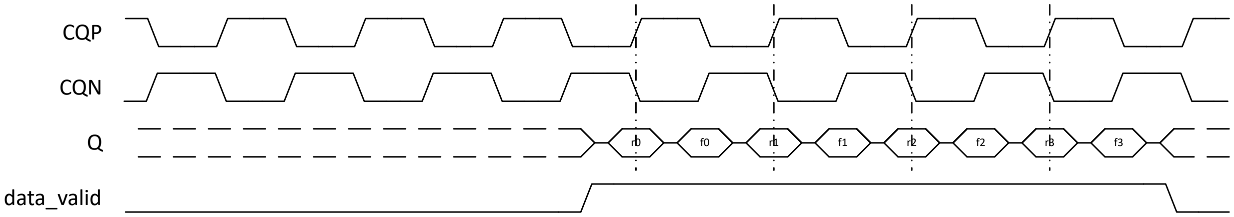
A high-level timing diagram of alignment is shown in the following figure.

Figure 4-4. Data capture after aligning data valid and falling edge data to CQN



Next, CQP is aligned with the rising edge data by performing write and read of 8 different patterns and selecting a match if and only if all patterns match.

Figure 4-5. Data capture after moving CQP to center rising edge data



The process is then repeated for all Q bits (9 bits per lane) and for all lanes.

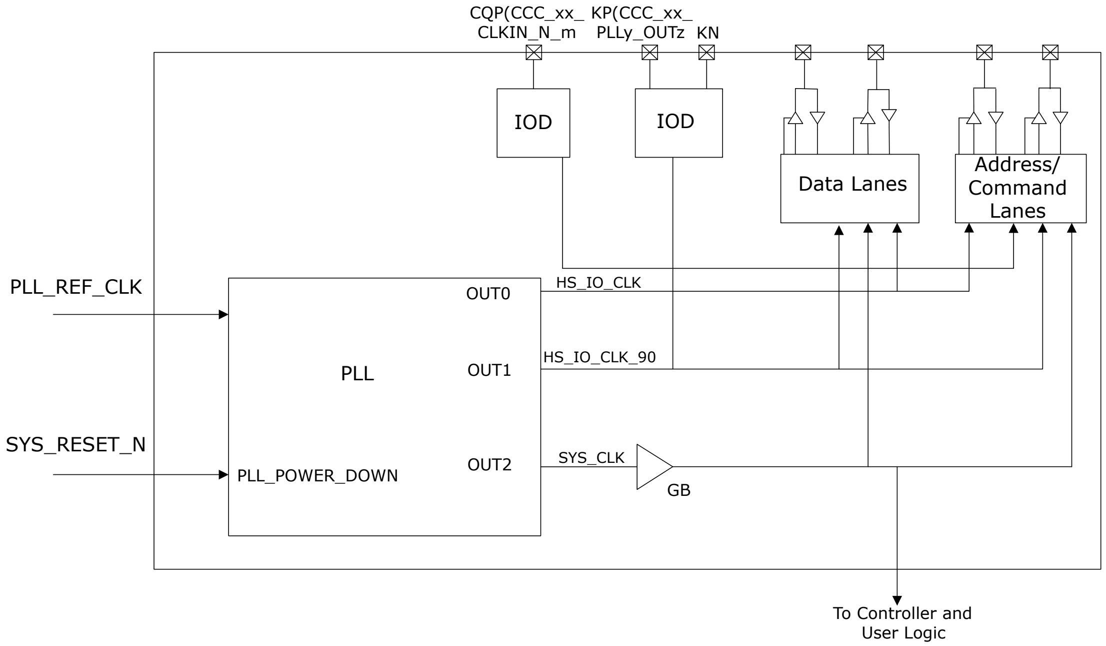
© 2021 Microchip Technology Inc.
and its subsidiaries

4.3.2 Clocking Structure

The QDR subsystem require a dedicated PLL to generate the clocks, which are then distributed throughout the subsystem using HS_IO_CLK routes, dedicated pads, and fabric clock routing. This PLL generates aligned clocks for all sub-blocks for smooth operation and synchronous communication with the user logic. The PLL generates three required clocks-HS_IO_CLK, HS_IO_CLK phase shifted by 90° (HS_IO_CLK_90), and controller clock (SYS_CLK). The two HSIO clocks (HS_IO_CLK and HS_IO_CLK_90) are routed to the PHY using HS_IO_CLK routing resources (for low skew) and multiplexers. These two clocks generate the QDR SRAM interface signals. SYS_CLK is routed to the QDR controller and user logic in the fabric. The QDR Memory Clock to SYS_CLK ratio is 4:1.

The PLL generates the QDR memory clocks (KP and KN) as shown in the following figure.

Figure 4-6. QDR Clocking Structure



The dedicated Clock Input Pad (CCC_xx_CLKIN_N_m) and Output Pad (CCC_xx_PLLy_OUTz) is selected depending on the QDR subsystem placement constraint (where xx = SE/NE/NW/SW, y = 0/1, z = 0/1 and m=0/1/2/3).

Note: Timing constraints are not required for output signals (Address, Command and Data). Because, these output signals are shifted 90° from the K clock.

4.3.3 Integrated PHY

The integrated PHY, which consists of the I/O lane provides a physical interface to QDR II+ and QDR II+ Xtreme SRAM devices. It receives commands from the QDR controller and generates the QDR memory signals required to access the external QDR memory. In QDR subsystem, the I/O Lane gear ratio is configured to 4:1.

4.3.3.1 I/O Lane

An I/O Lane contains: Twelve I/Os, a Lane Controller, I/O Gearing Logic, and a set of High-speed & Low-skew clock resources. The I/O Gearing Logic in a lane enables easy data transfer between the high-speed I/O pad and the Lower-speed FPGA core. The logic is used to either gear up the data rate from the FPGA Fabric to the memory device or gear down the data rate from the memory device to the FPGA fabric. For information about non-memory interface usage, see PolarFire FPGA and PolarFire SoC FPGA User I/O User Guide.

The Lane Controller contains the logic for managing the Read and Write signals and provides the Lane Clocks. Each Output Data (D) lane (x9) uses one I/O lane with 12 I/O pads each: Nine of the I/O pads are used for the Data (D) bits, and one for BWSx_N and the remaining I/O pads are left as spare.

Each Input Data (Q) lane(x9) uses one I/O lane with 12 I/O pads each-Nine of the I/O pads are used for the Data (D) bits and the remaining I/O pads are left as spare.

Address, Clock and Read/Write command (RPS_N and WPS_N) uses 3 I/O lanes.

Note: Depending on the configuration of the QDR memory, few of the I/Os in Address/Command Lanes are used. Some I/Os in these Lanes can be reused as normal I/Os. For Data Lanes, all 12 I/Os in the Lane are used.

4.3.4 Port List

The following table lists the QDR ports.

Table 4-3. QDR Signals and Descriptions

Name	Width	Type	Description
Generic Signals			
SYS_CLK	1	Output	Clock for User Logic generated by the embedded PLL. All User Interface signals are synchronous to this clock. Always on the global clock network.
RESET_N	1	Input	Active-low asynchronous system reset.
PLL_REF_CLK		Input	Reference clock to the PLL.
PLL_LOCK	1	Output	Lock signal generated by the PLL to indicate that the PLL is locked on to the PLL_REF_CLK signal.
TRAINING_COMPLETE	1	Output	Indicates that Training is complete. The user logic must check for TRAINING_COMPLETE = 1 and TRAINING_ERROR = 0 before accessing the QDR memory.
TRAINING_ERROR[2:0]	3	Output	– Bit 0: Reserved – Bits [2:1] • 0: No error. • 1: Timeout error, data VALID is received during Q training. • 2: No solution found during training. If no solution is found, restart the Training process described in 4.3.1.1 Training Logic.
User Interface Signals			
READ_N	4	Input	Read request with address specified on RADDR bus. 4-bit Wide if Burst Size is 2. 2-bit Wide if Burst Size is 4.
WRITE_N	4	Input	Write request with address specified on WADDR bus. 4-bit Wide if Burst Size is 2. 2-bit Wide if Burst Size is 4.
WSTRB_N	SRAM_DATA_WIDTH / 9 * 8	Input	Write strobe for write transaction.

...........continued

Name	Width	Type	Description

WADDR	(QDR SRAM Address Width)*4	Input	Write addresses for each beat. Width is (QDR SRAM Address Width)*4 if burst size is 2, (QDR SRAM Address Width)*2 if burst size is 4
			In Burst of 2: {ADDR3, ADDR2, ADDR1, ADDR0}
			In Burst of 4: {ADDR1, ADDR0}

RADDR	(QDR SRAM Address Width)*4	Input	Read addresses for each beat. Width is (QDR SRAM Address Width)*4 if burst size is 2, (QDR SRAM Address Width)*2 if burst size is 4
			In Burst of 2: {ADDR3, ADDR2, ADDR1, ADDR0}
			In Burst of 4: {ADDR1, ADDR0}

WDATA	(QDR SRAM Data Width)*8	Input	Write data
RDATA	(QDR SRAM Data Width)*8	Output	Read data, valid when RVALID is asserted
RVALID	1	Output	Read data valid
QDR SRAM Interface			
D[X:0]	9/18/36 (X=8,17, or 35)	Output	Data Output Bus
WPS_N	1	Output	Write port select - Active LOW
BWS0_N	1	Output	Byte write select (BWS) 0 - Active LOW.
BWS1_N	1	Output	Byte write select (BWS) 1 - Active LOW.
BWS2_N	1	Output	Byte write select (BWS) 2 - Active LOW.
BWS3_N	1	Output	Byte write select (BWS) 3 - Active LOW.
A[X:0]	18,19,20, or 21 (X=17,18,19, or 20)	Output	Address inputs
Q[X:0]	9,18, or 36 (X=8,17, or 35)	Input	Data input bus
RPS_N	1	Output	Read port select - Active LOW.
QVALID	1	Input	Valid output indicator. Not used in current QDR subsystem
KP	1	Output	Positive Clock Output.

© 2021 Microchip Technology Inc. and its subsidiaries

...........continued

Name	Width	Type	Description
KN	1	Output	Negative Clock Output.
CQP	1	Input	Synchronous echo Clock input.
DOFF_N	1	Output	Initially, DOFF_N is 0. After calibration, DOFF_N is set to 1.

4.4 Functional Timing Diagrams

This section describes the read/write rules and sequences in the User Interface. For information about User Interface signals, see 4.3.4 Port List.

4.4.1 Write Strobe Mapping

Mapping of the write strobe signal WSTRB to write data WDATA is done on a phased basis, by mapping WDATA to the QDR data bus D, as shown in Figure 7, page 12 and Figure 8, page 13. However, it is also aligned to BWSx_N signals on the QDR interface as follows. Note that the following example is for SRAM_DWIDTH of 36, or x36 mode.

Table 4-4. WSTRB to BWS Mapping

WSTRB Index	BWS Index
7:0	0
15:8	1
23:16	2
31:24	3

Within each WSTRB byte, each bit corresponds to a clock cycle (phase) and either rising or falling edge. A complete mapping for write data bus WDATA to WSTRB is listed in the following tables, for x36, x18, and x9 modes.

Table 4-5. Write Data to Write Strobe Mapping (x36)

WSTRB_MAPPING_INDEX	WDATA INDEX									RISING/FALLING	CLOCK CYCLE
0	0	1	2	3	4	5	6	7	8	RISING	1
8	36	37	38	39	40	41	42	43	44	FALLING	1
16	72	73	74	75	76	77	78	79	80	RISING	2
24	108	109	110	111	112	113	114	115	116	FALLING	2
1	144	145	146	147	148	149	150	151	152	RISING	3
9	180	181	182	183	184	185	186	187	188	FALLING	3
17	216	217	218	219	220	221	222	223	224	RISING	4
25	252	253	254	255	256	257	258	259	260	FALLING	4
2	9	10	11	12	13	14	15	16	17	RISING	1
10	45	46	47	48	49	50	51	52	53	FALLING	1
18	81	82	83	84	85	86	87	88	89	RISING	2
26	117	118	119	120	121	122	123	124	125	FALLING	2
3	153	154	155	156	157	158	159	160	161	RISING	3
11	189	190	191	192	193	194	195	196	197	FALLING	3
19	225	226	227	228	229	230	231	232	233	RISING	4
27	261	262	263	264	265	266	267	268	269	FALLING	4

...........continued

WSTRB_MAPPING_INDEX	WDATA INDEX									RISING/ FALLING	CLOCK CYCLE
4	18	19	20	21	22	23	24	25	26	RISING	1
12	54	55	56	57	58	59	60	61	62	FALLING	1
20	90	91	92	93	94	95	96	97	98	RISING	2
28	126	127	128	129	130	131	132	133	134	FALLING	2
5	162	163	164	165	166	167	168	169	170	RISING	3
13	198	199	200	201	202	203	204	205	206	FALLING	3
21	234	235	236	237	238	239	240	241	242	RISING	4
29	270	271	272	273	274	275	276	277	278	FALLING	4
6	27	28	29	30	31	32	33	34	35	RISING	1
14	63	64	65	66	67	68	69	70	71	FALLING	1
22	99	100	101	102	103	104	105	106	107	RISING	2
30	135	136	137	138	139	140	141	142	143	FALLING	2
7	171	172	173	174	175	176	177	178	179	RISING	3
15	207	208	209	210	211	212	213	214	215	FALLING	3
23	243	244	245	246	247	248	249	250	251	RISING	4
31	279	280	281	282	283	284	285	286	287	FALLING	4

[1] The mapping is not sequential. First bit of write_strobe corresponds to the first 9 wdata bits. The second bit does not correspond to the next 9 wdata bits.

Table 4-6. Write Data to Write Strobe Mapping (x18)

WSTRB_MAPPING_INDEX	WDATA INDEX									RISING/FALLING	CLOCK CYCLE
0	0	1	2	3	4	5	6	7	8	RISING	1
8	36	37	38	39	40	41	42	43	44	FALLING	1
1	72	73	74	75	76	77	78	79	80	RISING	2
9	108	109	110	111	112	113	114	115	116	FALLING	2
2	9	10	11	12	13	14	15	16	17	RISING	3
10	45	46	47	48	49	50	51	52	53	FALLING	3
3	81	82	83	84	85	86	87	88	89	RISING	4
11	117	118	119	120	121	122	123	124	125	FALLING	4
4	18	19	20	21	22	23	24	25	26	RISING	1
12	54	55	56	57	58	59	60	61	62	FALLING	1
5	90	91	92	93	94	95	96	97	98	RISING	2
13	126	127	128	129	130	131	132	133	134	FALLING	2
6	27	28	29	30	31	32	33	34	35	RISING	3
14	63	64	65	66	67	68	69	70	71	FALLING	3
7	99	100	101	102	103	104	105	106	107	RISING	4
15	135	136	137	138	139	140	141	142	143	FALLING	4

[1] The mapping is not sequential. First bit of write_strobe corresponds to the first 9 wdata bits. The second bit does not correspond to the next 9 wdata bits.

Table 4-7. Write Data to Write Strobe Mapping (x9)

WSTRB_MAPPING_INDEX	WDATA INDEX									RISING/ FALLING	CLOCK CYCLE
0	0	1	2	3	4	5	6	7	8	RISING	1
1	36	37	38	39	40	41	42	43	44	FALLING	2
2	9	10	11	12	13	14	15	16	17	RISING	3
3	45	46	47	48	49	50	51	52	53	FALLING	4
4	18	19	20	21	22	23	24	25	26	RISING	1
5	54	55	56	57	58	59	60	61	62	FALLING	2
6	27	28	29	30	31	32	33	34	35	RISING	3
7	63	64	65	66	67	68	69	70	71	FALLING	4

4.4.1.1 Burst of 2

In burst of 2 mode (SRAM_BURST=1), all bits of both WADDR/RADDR and WRITE_N/READ_N (the corresponding control signals) are used. A sample write transfer with SRAM_DWIDTH=18 and SRAM_AWIDTH=20 is shown in the following figure.

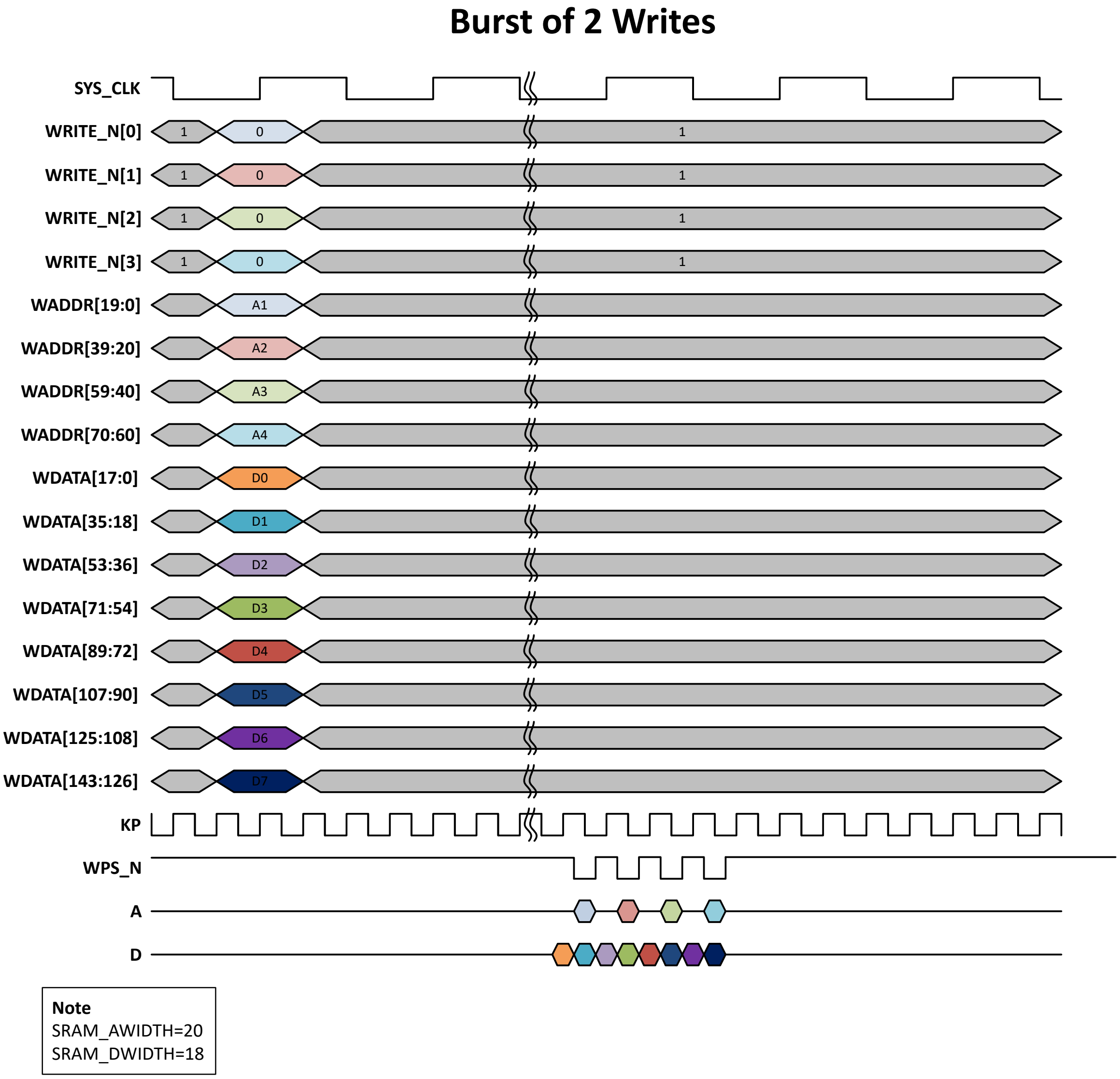
In burst of 2 mode, 4 write transactions (WRITE_N[0,1,2,3]) are completed in one SYS_CLK cycle. In each transaction data is 18-bit wide and address is 20-bit wide. The 4 write transactions must be triggered with respect to KP clock, which is derived from SYS_CLK internally. The first write (when WRITE_N[0] = 0) occurs at the first falling edge of the WPS_N write enable signal. The second write (when WRITE_N[1] = 0) occurs at the next falling edge of WPS_N. Similarly, the remaining writes are triggered.

Write transaction is performed as follows:

- When the WPS_N enable signal is '0', four 18-bit data are written in the specified address 'A' in a single system clock.
- A1= WADDR[19:0], A2= WADDR[39:20], A3= WADDR[59:40], A4= WADDR[79:60]
- During each write enable (WPS_N), 2 write transactions are performed as follows:
  - A1 address D0 = WDATA[17:0] and D1=WDATA[35:18]
  - A2, D2 = WDATA[53:36] and D3=WDATA[71:54]
  - A3, D4 = WDATA[89:72] and D5=WDATA[107:90]
  - A4, D6 = WDATA[125:108] and D7=WDATA[143:126]

Read transactions are not shown, but are similar, as each RPS_N assertion and the corresponding A bus values are derived from the four slices of READ_N and RADDR, respectively.

Figure 4-7. Burst of 2 Local Interface Timing



4.4.1.2 Burst of 4

In the burst of 4 mode (SRAM_BURST = 0), only half the WADDR/RADDR and WRITE_N/READ_N are utilized, because each SYS_CLK cycle transaction corresponds to only 2 (instead of 4) K clock transactions. A sample write transaction in burst of 4 mode with SRAM_AWIDTH=20 and SRAM_DWIDTH=36 is show in the following figure.

In burst of 4 mode, 2 write transactions (WRITE_N[0,1]) are completed in one SYS_CLK cycle. In each transaction data is 36-bit wide and address is 20-bit wide. The 4 write transactions must be triggered with respect to KP clock, which is derived from SYS_CLK internally. The first write (when WRITE_N[0] = 0) occurs at the first falling edge of the WPS_N write enable signal. The second write (when WRITE_N[1] = 0) occurs at the next falling edge of WPS_N.

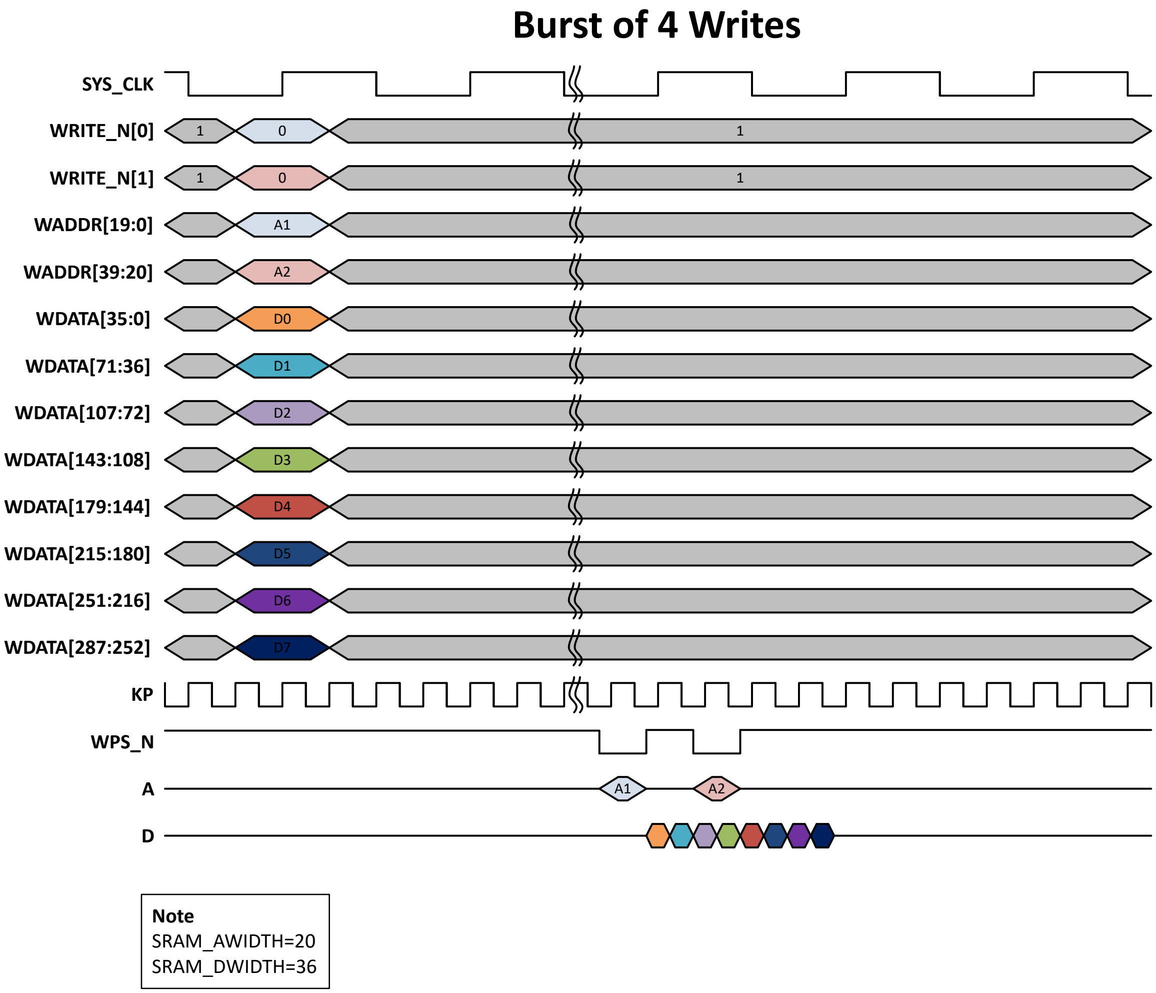
Write transaction is performed as follows:

- When the WPS_N enable signal is '0', 2 36-bit data are written in the specified address 'A' in a single system clock.
- A1= WADDR[19:0], A2= WADDR[39:20].
- During each write enable, 4 write transactions are performed in each burst, as follows:

© 2021 Microchip Technology Inc.
and its subsidiaries

- A1 address D0=WDATA[35:0], D1=WDATA[71:36], D2=WDATA[107:72] and D3=WDATA[143:108].
- A2, D4=WDATA[179:144] and D5=WDATA[215:180], D6=WDATA[251:216] and D7=WDATA[287:252]

Figure 4-8. Burst of 4 Local Interface Timing



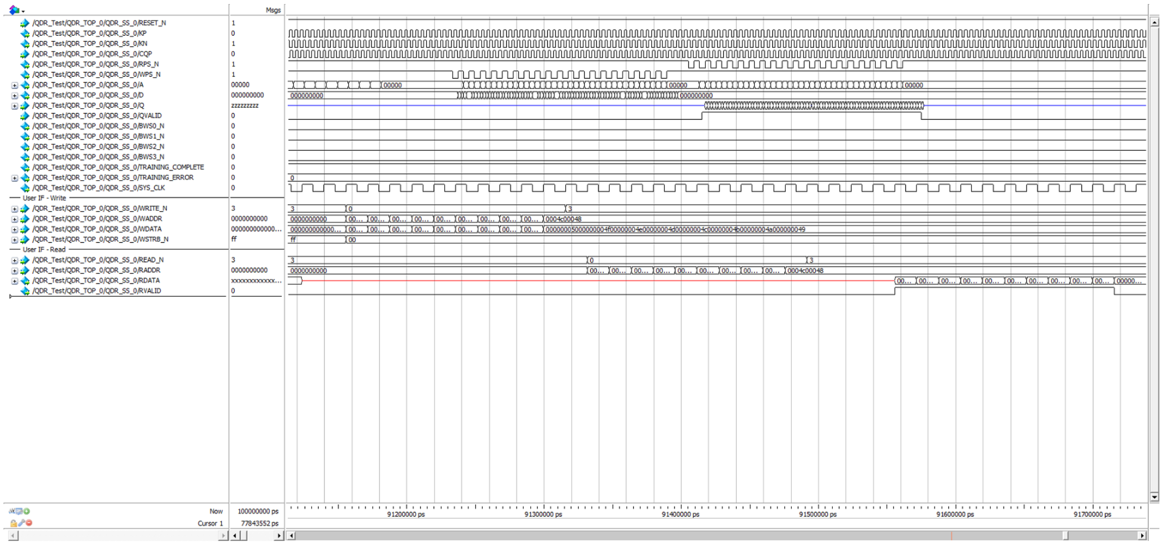
4.4.2 QDR SRAM Write

QDR SRAM Writes are requested by asserting the WRITE_N[x] signal low, and by driving the Address on the WADDR. The following rules are applicable for the write requests in the user interface:

1. A Write is accepted by the Subsystem on any SYS_CLK clock cycle, where the WRITE_N[x] signal is asserted to Low. Each Write Transaction performs the 2/4 Burst Write operations as per the QDR configuration.
2. Multiple Writes can be initiated by asserting the multiple WRITE_N[x] signals to low. The address WADDR[SRAM_ADDR_WIDTH*(x+1) -1: 0] and DATA[SRAM_DATA_WIDTH*(x+1)*burst_mode -1:0] will be the corresponding Write Address and Data for the Write Transaction associated with WRITE_N[x].
   - When QDR is configured for 2 Burst mode, four Burst Writes are initiated in single SYS_CLK clock cycle by using WRITE_N[3:0].
   - When QDR is configured for 4 Burst mode, two Burst Writes are initiated in single SYS_CLK clock cycle by using WRITE_N[1:0].
3. Maximum of 8 SRAM_DATA_WIDTH Bit writes to QDR SRAM are initiated in a single SYS_CLK clock cycle.

© 2021 Microchip Technology Inc.
and its subsidiaries

Figure 4-9. Write in 4 Burst Mode



	Msgs
/QDR_Test/QDR_TOP_0/QDR_SS_0/RESET_N	1
/QDR_Test/QDR_TOP_0/QDR_SS_0/KP	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/KN	1
/QDR_Test/QDR_TOP_0/QDR_SS_0/CQP	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/RPS_N	1
/QDR_Test/QDR_TOP_0/QDR_SS_0/WPS_N	1
/QDR_Test/QDR_TOP_0/QDR_SS_0/A	00000
/QDR_Test/QDR_TOP_0/QDR_SS_0/D	000000000
/QDR_Test/QDR_TOP_0/QDR_SS_0/Q	zzzzzzzzz
/QDR_Test/QDR_TOP_0/QDR_SS_0/QVALID	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS0_N	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS1_N	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS2_N	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS3_N	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/TRAINING_COMPLETE	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/TRAINING_ERROR	0
/QDR_Test/QDR_TOP_0/QDR_SS_0/SYS_CLK	0
User IF - Write	
/QDR_Test/QDR_TOP_0/QDR_SS_0/WRITE_N	3
/QDR_Test/QDR_TOP_0/QDR_SS_0/WADDR	0000000000
/QDR_Test/QDR_TOP_0/QDR_SS_0/WDATA	000000000000...
/QDR_Test/QDR_TOP_0/QDR_SS_0/WSTRB_N	ff
User IF - Read	
/QDR_Test/QDR_TOP_0/QDR_SS_0/READ_N	3
/QDR_Test/QDR_TOP_0/QDR_SS_0/RADDR	0000000000
/QDR_Test/QDR_TOP_0/QDR_SS_0/RDATA	xxxxxxxxxxxx...
/QDR_Test/QDR_TOP_0/QDR_SS_0/RVALID	0
Now	100000000 ps
Cursor 1	77843552 ps

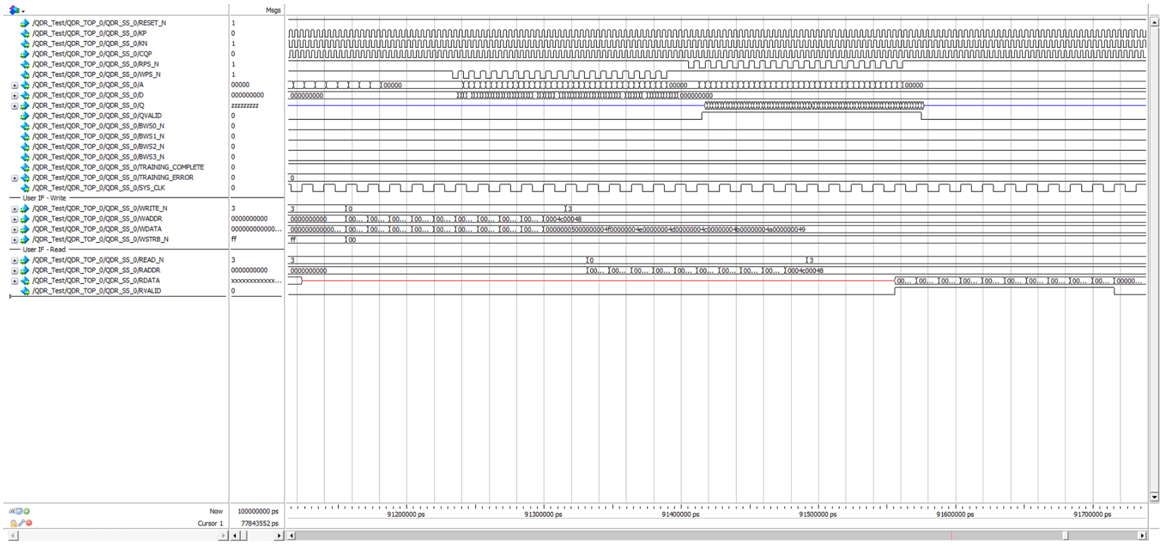
4.4.3 QDR SRAM Read

QDR SRAM Reads are requested by asserting the READ_N[x] signal low, and by driving the address on the RADDR. QDR controller asserts the RVALID to high when valid data is available on RDATA.

The following rules are applicable for the read requests in the user interface:

1. A Read is accepted by the Subsystem on any SYS_CLK clock cycle where the READ_N[x] signal is asserted to Low. Each Read Transaction performs the 2/4 burst Read Operations as per the QDR configuration.
2. Multiple Reads can be initiated by asserting the multiple READ_N[x] signals to Low. The address RADDR[SRAM_ADDR_WIDTH*(x+1) -1 : 0] and RDATA[SRAM_DATA_WIDTH*(x+1)*burst_mode - 1:0] will be the corresponding Read Address and Data for the Read Transaction associated with READ_N[x].
   – When QDR is configured for 2 Burst Mode, four burst Reads are initiated in single SYS_CLK clock cycle by using READ_N[3:0].
   – When QDR is configured for 4 Burst Mode, two burst Reads are initiated in single SYS_CLK clock cycle by using READ_N[1:0].

Figure 4-10. Read in 4 Burst Mode



```
                                                Msgs
/QDR_Test/QDR_TOP_0/QDR_SS_0/RESET_N            1
/QDR_Test/QDR_TOP_0/QDR_SS_0/KP                 0
/QDR_Test/QDR_TOP_0/QDR_SS_0/KN                 1
/QDR_Test/QDR_TOP_0/QDR_SS_0/CQP                0
/QDR_Test/QDR_TOP_0/QDR_SS_0/RPS_N              1
/QDR_Test/QDR_TOP_0/QDR_SS_0/WPS_N              1
/QDR_Test/QDR_TOP_0/QDR_SS_0/A                  00000
/QDR_Test/QDR_TOP_0/QDR_SS_0/D                  000000000
/QDR_Test/QDR_TOP_0/QDR_SS_0/Q                  zzzzzzzzz
/QDR_Test/QDR_TOP_0/QDR_SS_0/QVALID             0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS0_N             0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS1_N             0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS2_N             0
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS3_N             0
/QDR_Test/QDR_TOP_0/QDR_SS_0/TRAINING_COMPLETE  0
/QDR_Test/QDR_TOP_0/QDR_SS_0/TRAINING_ERROR     0
/QDR_Test/QDR_TOP_0/QDR_SS_0/SYS_CLK            0
User IF - Write
/QDR_Test/QDR_TOP_0/QDR_SS_0/WRITE_N            3
/QDR_Test/QDR_TOP_0/QDR_SS_0/WADDR              0000000000
/QDR_Test/QDR_TOP_0/QDR_SS_0/WDATA              000000000000...
/QDR_Test/QDR_TOP_0/QDR_SS_0/WSTRB_N            ff
User IF - Read
/QDR_Test/QDR_TOP_0/QDR_SS_0/READ_N             3
/QDR_Test/QDR_TOP_0/QDR_SS_0/RADDR              0000000000
/QDR_Test/QDR_TOP_0/QDR_SS_0/RDATA              xxxxxxxxxxxx...
/QDR_Test/QDR_TOP_0/QDR_SS_0/RVALID             0

Now        100000000 ps
Cursor 1   77843552 ps
```

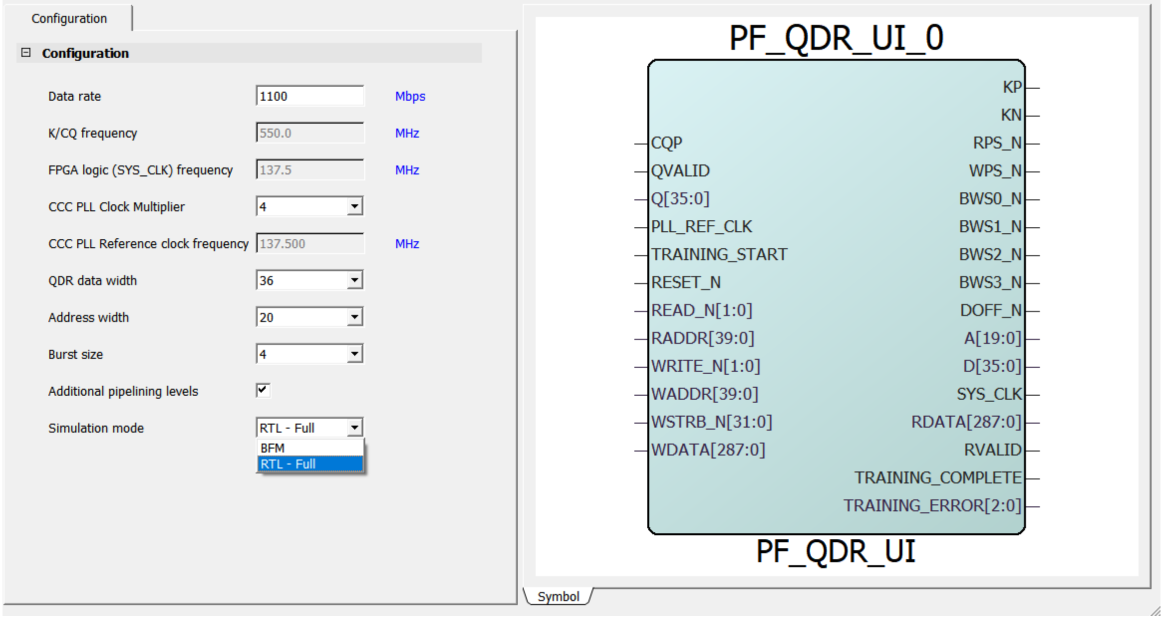
4.5 Implementation

This section describes how to use QDR Subsystem in the design.

4.5.1 Configuring QDR Memory Controller IP

The QDR Subsystem macro (PolarFire QDR) located in the Libero IP catalog must be instantiated in SmartDesign to access the QDR memory from the FPGA fabric through the subsystem. The QDR Configurator shown in the following figure configures the QDR Subsystem.

Figure 4-11. QDR Configurator



Configuration

⊟ **Configuration**

Data rate	1100	Mbps
K/CQ frequency	550.0	MHz
FPGA logic (SYS_CLK) frequency	137.5	MHz
CCC PLL Clock Multiplier	4	
CCC PLL Reference clock frequency	137.500	MHz
QDR data width	36	
Address width	20	
Burst size	4	
Additional pipelining levels	☑	
Simulation mode	RTL - Full	

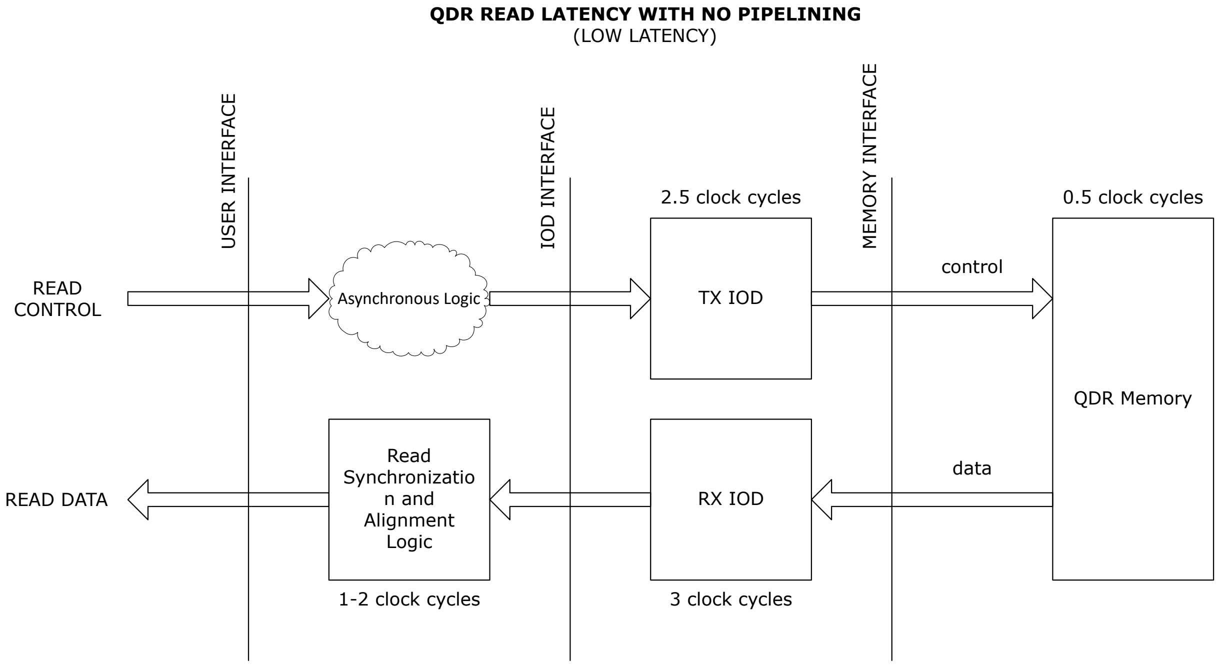
The following fields are available for configuration of QDR Subsystem:

- **Data rate**: Allows you to set the data rate of QDR SRAM memory. The K/CQ Clock frequencies are calculated using data rate value (divide by 2). The FPGA logic (SYS_CLK) frequency is automatically populated (data rate/8).
- **CCC PLL clock Multiplier**: Allows you to set the QDR Subsystem PLL Clock multiplier. The PLL Reference Clock frequency must be K/CQ frequency / CCC PLL Clock multiplier.
- **QDR data width**: Allows you to set 9,18,36-bit Data Width.
- **Address width**: Allows you to set 18 to 21-bit Address Width.
- **Burst size**: Allows you to select Burst Size of 2 or 4.
- **Simulation mode**: PolarFire QDR Subsystem supports following simulation modes:
  - **BFM**: In this mode, the QDR subsystem skips the training sequence and asserts the TRAINING_COMPLETE signal. QDR memory model is not required for BFM Simulation
  - **RTL_Full**: In this mode, the QDR subsystem performs the training sequence and asserts the TRAINING_COMPLETE signal. TRAINING_COMPLETE is asserted within 10 ms. QDR memory model is required for RTL_FULL Simulation.
- **Additional pipelining levels**: Optional pipelining to help with static timing closure, if necessary. Enabling this option increases the read latency by 2 - 3 SCLK cycles. The following section shows the latency with pipelining enabled and disabled.

4.5.1.1 Pipelining Latency

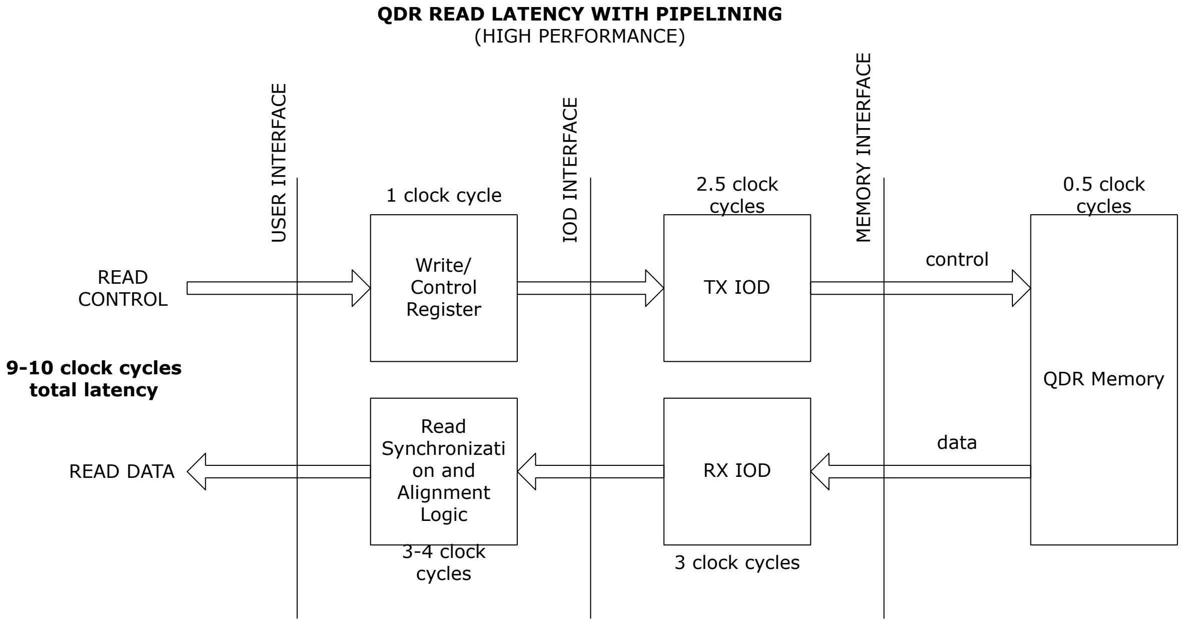
The following figure shows the QDR read latency with pipelining disabled.

Figure 4-12. Latency without Pipelining



The following figure shows the QDR read latency with pipelining enabled.

Figure 4-13. Latency with Pipelining



4.5.2 Simulating QDR Memory Controller

Libero SoC supports simulation for the QDR Subsystem, which includes a training sequence. For simulating this Subsystem, the QDR SRAM interface must be connected to the vendor-specific QDR memory simulation model in a testbench. Memory vendors (like Cypress Semiconductor) provide downloadable simulation models for memory devices. For more information on Simulations: Set-up and Running, see 4.6 Functional Example.

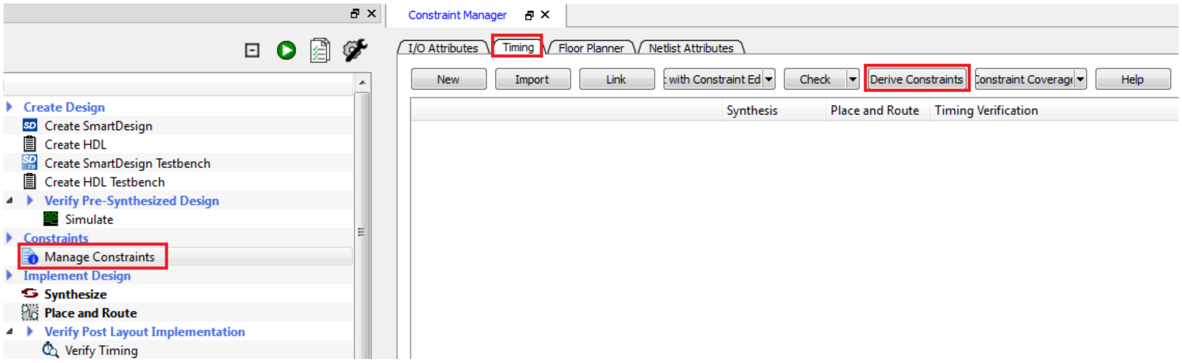
4.5.3 Design Constraints

This section describes Timing Constraints and Physical Constraints of the QDR Subsystem.

4.5.3.1 Timing Constraints

The HS_IO_CLK, HS_IO_CLK_90, and SYS_CLK Clocks generated using the dedicated PLL requires timing constraints for synthesis, place and route, and timing verification. To generate these timing constraints, select the Timing tab in Constraint Manager, and click Derive Constraints, as shown in the following figure. It also generates the required multi cycle/false path constraints.

Figure 4-14. Constraint Manager-Derive Constraints Tab



When prompted, Click Yes to apply the derived constraints for synthesis, place and route, and timing verification.

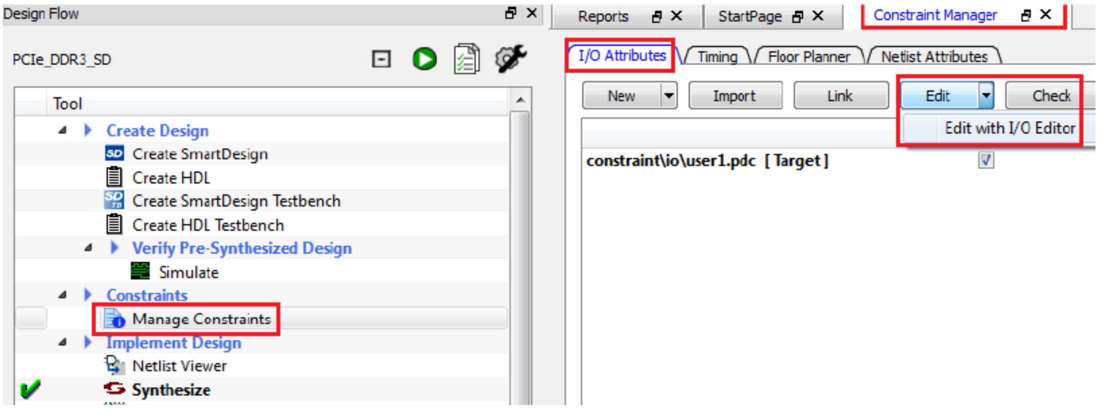
4.5.3.2 Physical Constraints

QDR subsystem can be placed only in the pre-defined locations that are optimized for maximum performance. The maximum width of the QDR subsystem varies based on die/package combination and placement. For information about the QDR placement, see the PolarFire and PolarFire SoC Package Pin Assignment Tables.

The QDR subsystem location is selected using the I/O Editor. Libero assigns I/Os based on the selection of DDR memory location. The following steps explains the selection of the QDR Subsystem Location:

1. Click **Design Flow** > **Constraint Manager** > **I/O Attributes** > **Edit with I/O Editor**.

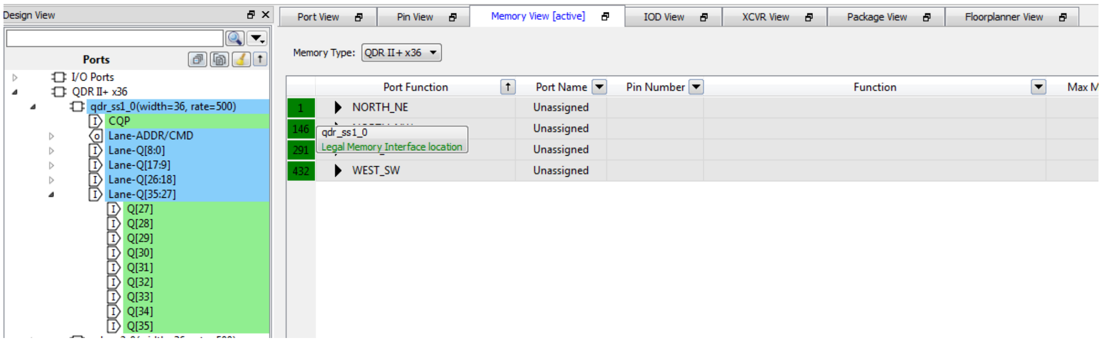
Figure 4-15. Design Flow Window: I/O Editor



constraint\io\user1.pdc [Target] ☑

2. Select the **Memory View** tab in I/O Editor. Select the appropriate **Memory Type** to get the valid locations for the chosen memory.
3. Select the appropriate QDR instance from the **Design View** window, and drag it onto **Port Function**. For the selected QDR instance, I/O Editor shows the valid locations in green.

Figure 4-16. Design View Window: Port Function



Memory Type: QDR II+ x36

	Port Function	Port Name	Pin Number	Function	Max M
1	NORTH_NE	Unassigned			
146		Unassigned			
291		Unassigned			
432	WEST_SW	Unassigned			

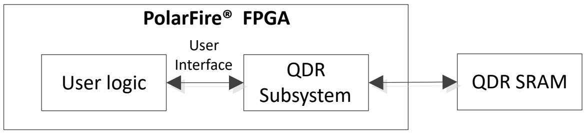
4.6 Functional Example

This section describes how to perform the RTL_FULL and BFM simulations.

4.6.1 RTL_FULL Simulation

The QDR subsystem is used to access QDR SRAMs directly, as shown in the following figure. User logic is connected directly to the QDR subsystem using the user interface.

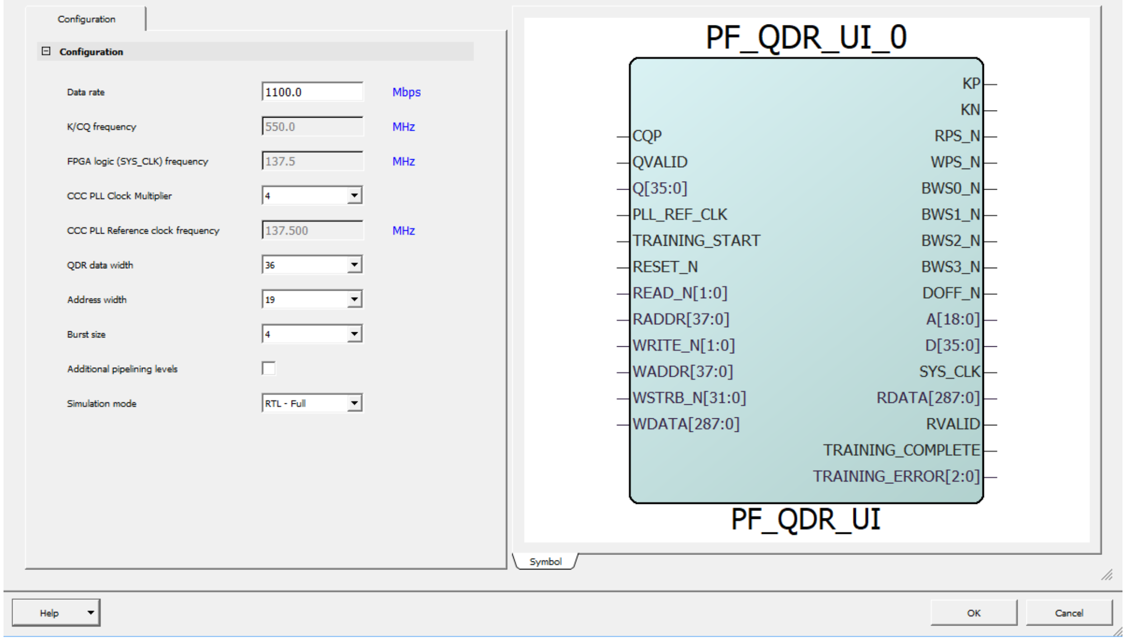
Figure 4-17. QDR Subsystem accessed through User Interface



After successful QDR initialization, the User Interface Master initiates Read or Write to the QDR memory. The following steps describe how to create a design for accessing the QDR memory from the user interface master in the FPGA fabric:

1. Create a SmartDesign, and instantiate the QDR component.
2. Configure the QDR Subsystem as described in 4.5 Implementation. In the following example, a design is created to access the QDR memory with 36-Bit Data Width, 19-Bit Address Width and Burst size of 4.

Figure 4-18. QDR Configuration- RTL_FULL



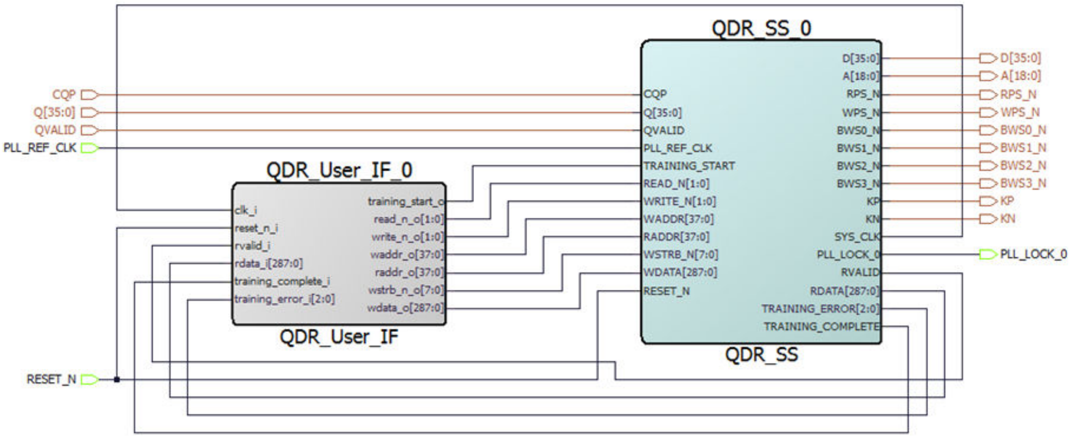
Configuration

⊟ Configuration

Data rate	1100.0	Mbps
K/CQ frequency	550.0	MHz
FPGA logic (SYS_CLK) frequency	137.5	MHz
CCC PLL Clock Multiplier	4	
CCC PLL Reference clock frequency	137.500	MHz
QDR data width	36	
Address width	19	
Burst size	4	
Additional pipelining levels	□	
Simulation mode	RTL - Full	

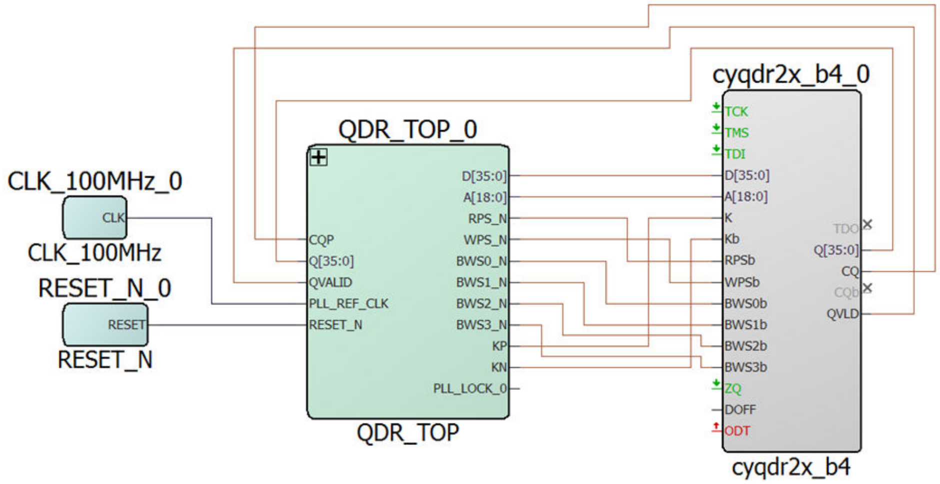
3. Instantiate the User Interface Master logic in the SmartDesign canvas. Ensure that the User Interface Master logic accesses the QDR subsystem only after TRAINING_COMPLETE is high.
4. In the SmartDesign canvas, connect the blocks, as shown in the following figure.

Figure 4-19. SmartDesign Connections



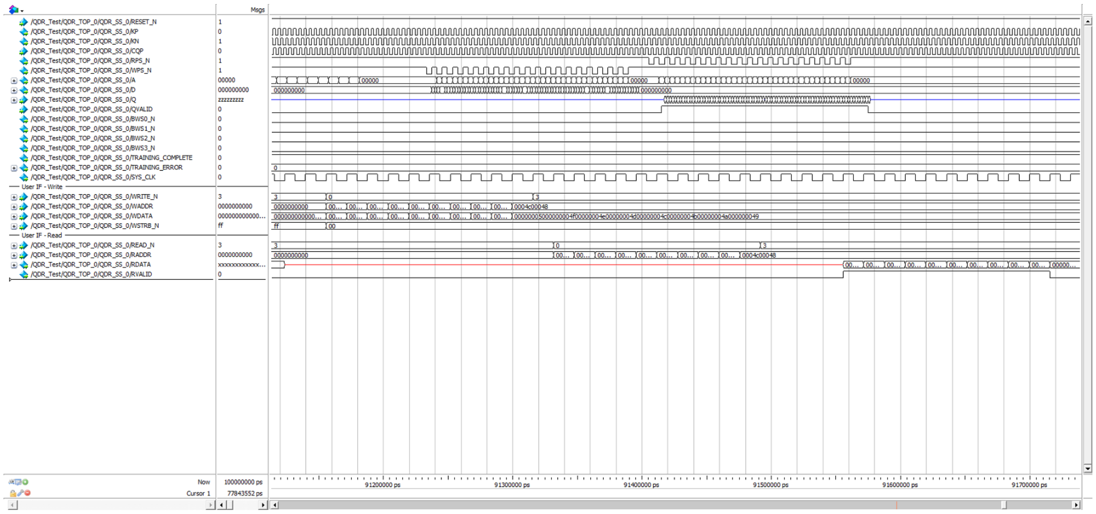
5. Create a new SmartDesign Testbench to simulate the design.
6. Instantiate the top-level design component and the QDR memory simulation model.
7. Configure CLK_GEN to generate the PLL reference clock and connect to PLL_REF_CLK.
8. Connect the blocks in SmartDesign Testbench, as shown in the following figure.

Figure 4-20. QDR SmartDesign Testbench



9. On Libero Design Flow -> Verify Pre-synthesized Design, select Simulate.
10. The QDR subsystem performs the training and asserts the TRAINING_COMPLTETE. The following figure shows the User interface read and writes transactions and the corresponding QDR SRAM transactions.

Figure 4-21. QDR Read and Write Transactions- RTL_FULL


Msgs
/QDR_Test/QDR_TOP_0/QDR_SS_0/RESET_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/KP
/QDR_Test/QDR_TOP_0/QDR_SS_0/KN
/QDR_Test/QDR_TOP_0/QDR_SS_0/CQP
/QDR_Test/QDR_TOP_0/QDR_SS_0/RPS_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/WPS_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/A
/QDR_Test/QDR_TOP_0/QDR_SS_0/D
/QDR_Test/QDR_TOP_0/QDR_SS_0/Q
/QDR_Test/QDR_TOP_0/QDR_SS_0/QVALID
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS0_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS1_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS2_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/BWS3_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/TRAINING_COMPLETE
/QDR_Test/QDR_TOP_0/QDR_SS_0/TRAINING_ERROR
/QDR_Test/QDR_TOP_0/QDR_SS_0/SYS_CLK
User IF - Write
/QDR_Test/QDR_TOP_0/QDR_SS_0/WRITE_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/WADDR
/QDR_Test/QDR_TOP_0/QDR_SS_0/WDATA
/QDR_Test/QDR_TOP_0/QDR_SS_0/WSTRB_N
User IF - Read
/QDR_Test/QDR_TOP_0/QDR_SS_0/READ_N
/QDR_Test/QDR_TOP_0/QDR_SS_0/RADDR
/QDR_Test/QDR_TOP_0/QDR_SS_0/RDATA
/QDR_Test/QDR_TOP_0/QDR_SS_0/RVALID
Now
100000000 ps
Cursor 1
77843552 ps
91200000 ps
91300000 ps
91400000 ps
91500000 ps
91600000 ps
91700000 ps

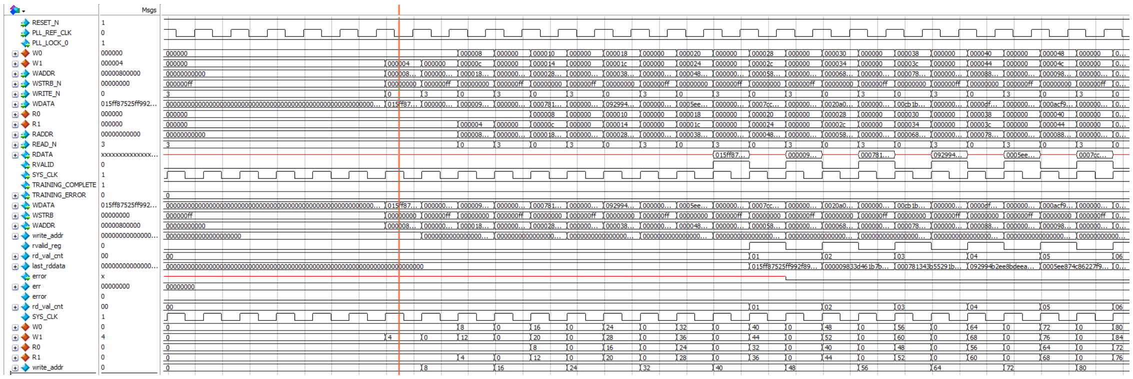
4.6.2 BFM Simulation

In the BFM mode, memory model is a part of the QDR BFM simulation model.

To run the BFM simulation:

1. Follow step 1 to 6 in 4.6.1 RTL_FULL Simulation.
2. Select **Simulate** from Libero **Design Flow** -> **Verify Pre-synthesized Design**.
3. The QDR subsystem skips training and asserts the TRAINING_COMPLTETE signal. The following figure shows the User interface read and writes transactions and the corresponding QDR SRAM transactions.

Figure 4-22. QDR Read and Write Transactions- BFM



4.7 PCB Recommendations

The routing board signals from FPGA to QDR II+ or QDR II+ Extreme devices, adhere to the constraints mentioned in the following tables.

Table 4-8. Maximum Length and Delay Constraints

Signal	Maximum Length (inches)	Routing Delay (ns)
Address and Command	6.0	1

...........continued

Signal	Maximum Length (inches)	Routing Delay (ns)
Data	6.0	1

Table 4-9. Skew Constraints

Signal	Skew Constraints
Data to Clock (D to K and Q to CQP)	±25 ps
Data Bus (D or Q)	±25 ps
Address and Command	±25 ps
Address, Command and to Clock (KP)	±25 ps
KP to KN	±10 ps

5. PolarFire Board Design Recommendations

The GPIO and HSIO of PolarFire devices support various memory interfaces.

- GPIO supports DDR3.
- HSIO supports DDR4, DDR3, and LPDDR3.

The following table lists the various types of memories supported in PolarFire devices.

Table 5-1. DDR3, and DDR4 Parameters

Parameter	DDR3	LPDDR3	DDR3L	DDR4
VDDIO	1.5 V	1.2 V	1.35 V	1.2 V
VTT, VREF	0.75 V	0.6 V	0.675 V	0.6 V
Clock, Address, and Command (CAC) Layout	Daisy Chained (Fly-by)	Point to point	Daisy Chained (Fly-by)	Daisy Chained (Fly-by)
Data Strobe	Differential	Differential	Differential	Differential
ODT	Dynamic	Dynamic	Dynamic	Dynamic
Match Addr/CMD/Ctrl to Clock Tightly	Yes	Yes	Yes	Yes
Match DQ/DM/DQS Tightly	Yes	Yes	Yes	Yes
Match DQS to Clock Loosely	Not Required	Not Required	Not Required	Not Required
I/O Standard	SSTL_15	HSUL-12	SSTL_135	POD-12*
RZQ	240_1%	240_1%	240_1%	240_1%

* HSTL_12 IO standard is used for DDR4 address, command, and control signals.

For more information on DDR memories, see the following documents:

- JESD79-3F JEDEC standard for DDR3 SDRAM specifications
- JESD79-4 JEDEC standard for DDR4 SDRAM specifications
- JESD209-3 JEDEC standard for LPDDR3 SDRAM specifications

5.1 DDR3

The following are the guidelines for connecting the device to the DDR3 memory:

- DDR3 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 60 Ω, and 120 Ω. DQ lines do not need VTT termination. However, VTT termination resistors need to be placed at the end of address and control lines on the PCB.
- Characteristic impedance: Z_O is typically 50 Ω, and Zdiff (differential) is 100 Ω.

The following table lists the DDR3 memory interface features supported in PolarFire devices.

Table 5-2. Interfaces Supported in DDR3 Memory Devices

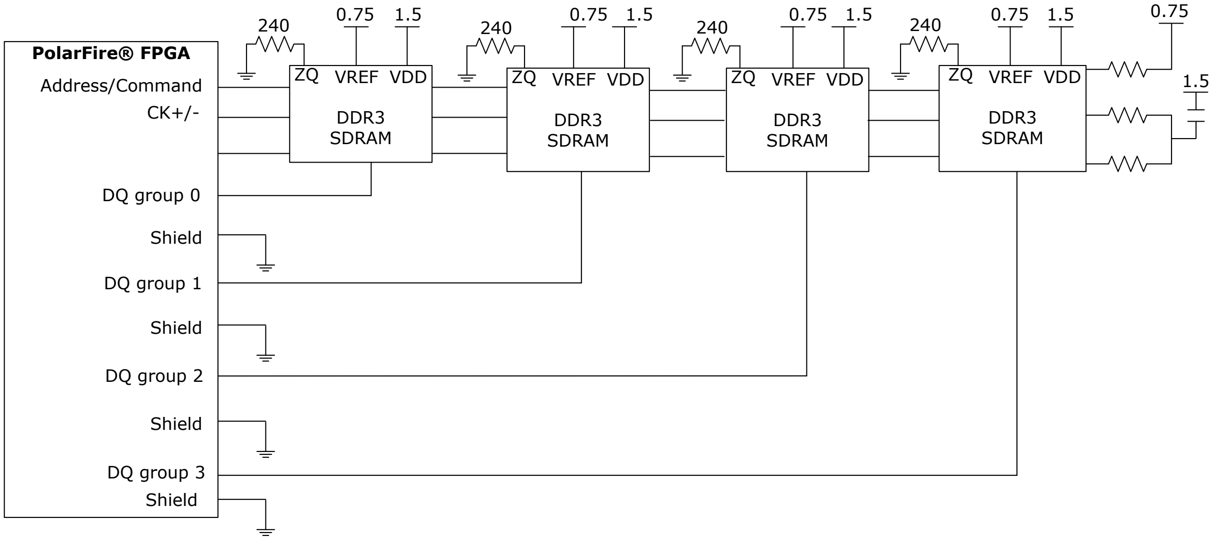
Interface	DDR3
Voltage	1.5 V
I/O standard	SSTL_15
Data rate	1333 MT/S-HSIO and 1066 MT/S-GPIO
Termination	ODT for data group; VTT termination for address, command, and control

...........continued

Interface	DDR3
Routing topology (CK, ADDR/ CMD, and CONTROL)	Fly-by
Data transmission	Point-to-point

The following figure shows the connectivity between DDR3 devices and PolarFire FPGAs.

Figure 5-1. DDR3 Interface Example



5.1.1 DDR3 Layout Guidelines

This section describes the routing guidelines for DDR3 interface for PolarFire Family. The guidelines are with reference to maximum x72 data width from signal integrity perspective. All the guidelines are provided considering maximum data rate supported. It is recommended to evaluate the interface by performing system level signal integrity simulations. The user is assumed to have the knowledge of the memory interface guidelines.

Table 5-3. DDR3 Interface Signals

Clock Signal	Description
CK0:1 P/N	Differential clock signals
Control signals	
CS[1:0]	Chip select
CKE[1:0]	Clock Enable
ODT[1:0]	On die termination enable
Reset_n	Reset signal
Address Signal	
A[15:0]	Address Signals
BA[1:0]	Bank address
Command Signal	
RAS_n	Row address select
CAS_n	Column address select

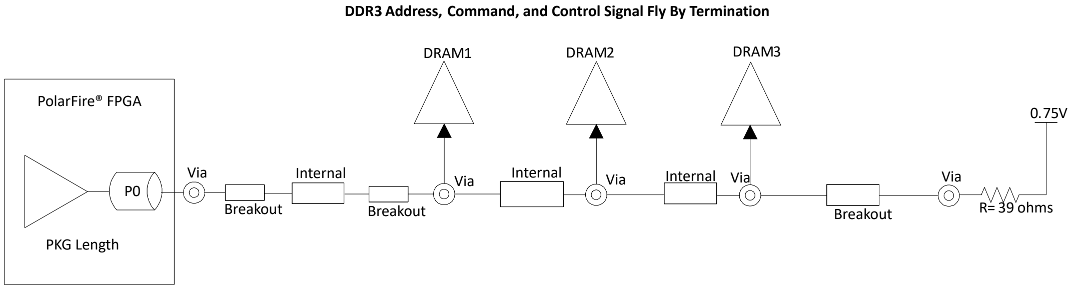
...........continued	
Clock Signal	**Description**
WE_n	Write enable
Data signals	
DQ[71:0]	Data bit
DQM[8:0]	Data mask
DQS_P/N[8:0]	Data strobe

Due to high speed signaling, DDR3 uses fly-by routing topology for Address, Command, Control signals to achieve best performance. As shown in the below figure, the ADDR/CMD are routed as single ended in fly-by topology. It is recommended to have all the signals being routed in one signal layer to control the skew within the signal groups. The DQ/DQS signals being point to point signals, uses on die termination on memory and FPGA side.

5.1.1.1 DDR3 Routing Topology

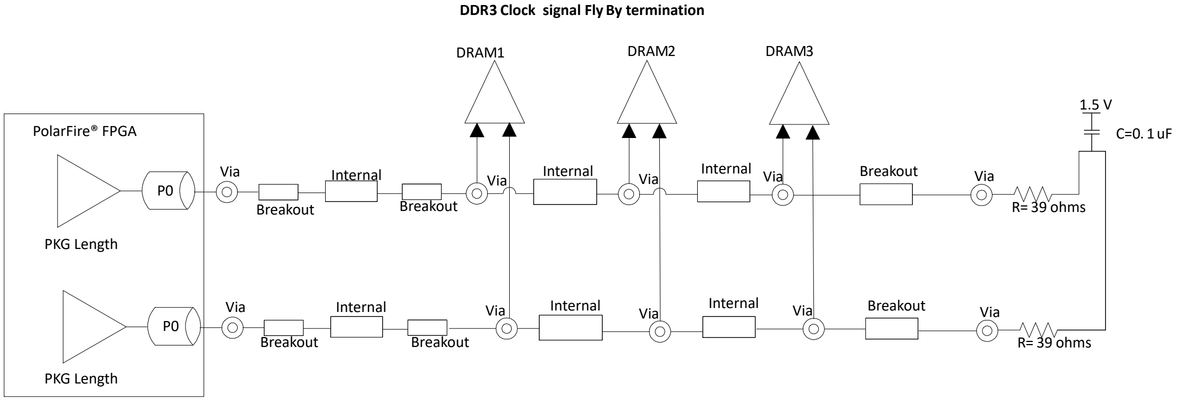
The following figure shows the FlyBy termination of the DDR3 address, command, and control signals (ACC).

Figure 5-2. DDR3 ACC Signals Fly By Termination



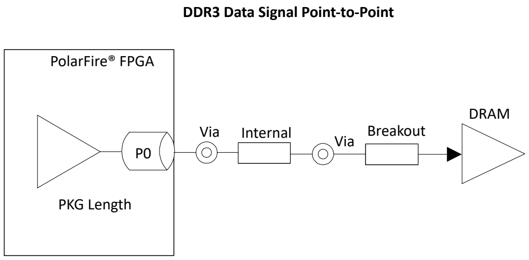
The following figure shows the FlyBy termination of the DDR3 clock signal.

Figure 5-3. DDR3 Clock Signal FlyBy Termination



The following figure shows the Point-to-Point routing of the DDR3 data signal.

Figure 5-4. DDR3 Data Signal Point-to-Point



5.1.1.2 Skew Constraints For Signal Groups

1. The address and command groups must be skew matched with respect to CK signals within ±100 ps.
2. The DQ, DM signals must be skew matched with respect to DQS within ±25 ps.
3. The read DQ/DQS training adjusts the DQS signal to optimal sampling point based on the board skew between DQ and DQS.
4. The write level training ensures that the tDQSS specification for memory is matched for each of the DQ/DQS byte groups and offset the skew introduced due to fly-by routing.

The characteristic impedance of 40 Ω-50 Ω must be used for single ended signals and 90 Ω-100 Ω for differential signals.

5.2 LPDDR3

The following are the guidelines for connecting the device to the LPDDR3 memory:

- LPDDR3 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 120 Ω and 240 Ω.
- DQ lines do not need VTT termination.
- The characteristic impedance, Zo is typically 50 Ω, and Zdiff (differential) is 100 Ω.

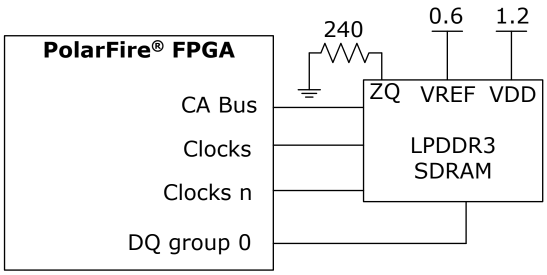
The following table lists the LPDDR3 memory interface features supported in PolarFire devices.

Table 5-4. Interfaces Supported in LPDDR3 Memory Devices

Interface	LPDDR3
Voltage	1.2 V
I/O standard	HSUL-12
Data rate	1333 MT/S-HSIO
Termination	ODT for data group
Routing topology	Point to point

The following figure shows the connectivity between LPDDR3 devices and PolarFire FPGAs.

Figure 5-5. LPDDR3 Interface Example



5.2.1 LPDDR3 Layout Guidelines

The LPDDR3 is a point-to-point interface, where the data group signals and address/command signals are routed from point to point. As LPDDR3 is used in power sensitive designs, there is no termination used for point to point interface. PolarFire supports up to 1333 MT/s for LPDDR3 interface on HSIO only.

It is recommended to tightly length match the signals CK, ADDR/CMD signals in one group and DQS, DQ, DM signals in one group together to reduce the skew.

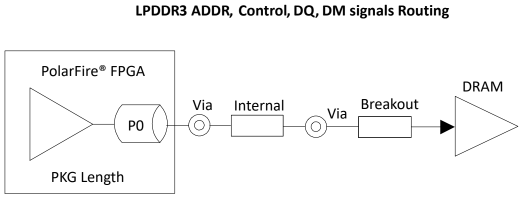
The characteristic impedance of 40-50 Ω should be used for single ended signals and 90-100 Ω for differential signals.

Microchip recommends simulating the interface to tightly match the characteristic impedance of trace with driver impedance to reduce reflections and for any possible electrical and timing violations in meeting JEDEC requirements.

5.2.1.1 LPDDR3 Routing Topology

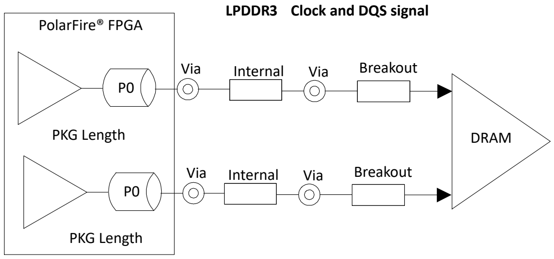
The following figure shows the routing of the LPDDR3 address, control, DQ, and DM signals.

Figure 5-6. LPDDR3 Address/Control/DQ/DM Signal Routing



The following figure shows the routing of LPDDR3 clock and DQS signals.

Figure 5-7. LPDDR3 Clock and DQS signal Routing



5.2.1.2 Skew Matching Constraints

1. Within the data group DQS, DQ, and DM signals must be matched with ±25 ps.
2. Skew between CK, CA, and CS signals within the group must not be more than ±30 ps.
3. Skew between CA and CK group and data lane group must not be more than ±350 ps.

Note: It is recommended to derive the accurate values of skew and termination based on simulations.

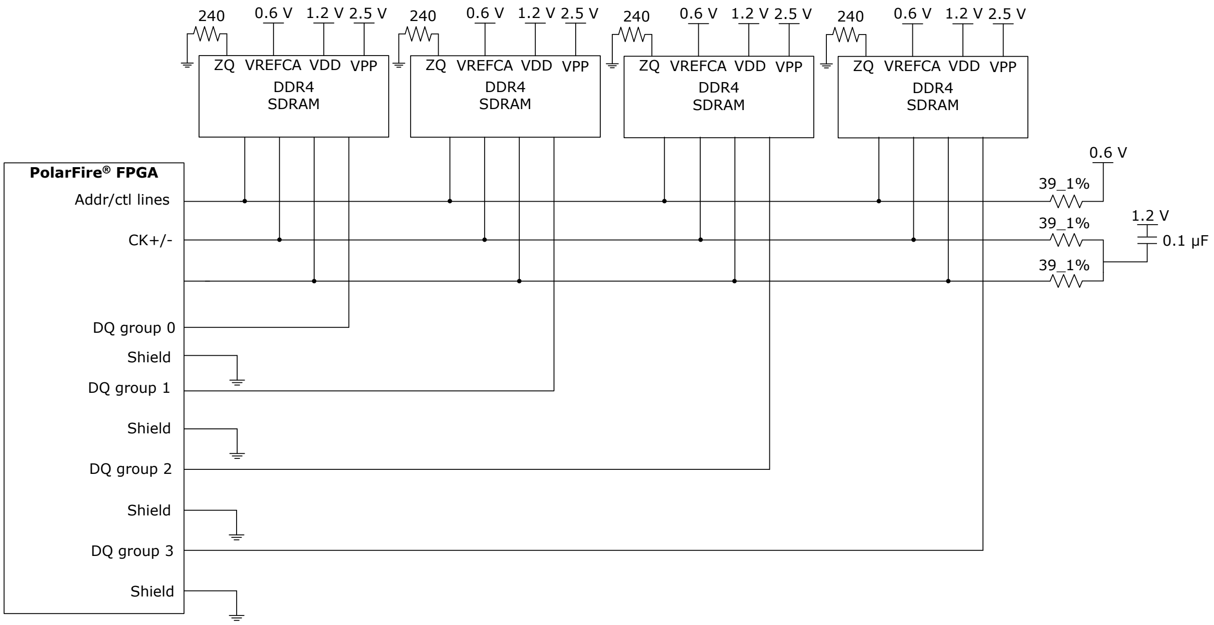
5.3 DDR4

The following are the guidelines for connecting the device to the DDR4 memory:

- DDR4 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 80 Ω, 120 Ω, and 240 Ω. DQ lines do not need VTT termination. However, VTT termination resistors need to be placed at the end of address and control lines on the PCB.
- Characteristic impedance: Z_0 is typically 50 Ω and Zdiff (differential) is 100 Ω.

The following figure shows the features supported by PolarFire FPGA in the DDR4 memory interface.

Figure 5-8. DDR4 Interface Example



5.3.1 DDR4 Layout Guidelines

This section describes the routing guidelines for DDR4 interface for PolarFire Family. The guidelines are with reference to maximum x72 data width from signal integrity perspective. All the guidelines are provided considering maximum data rate supported. It is recommended to evaluate the interface by performing system level signal integrity simulations. The user is assumed to have the knowledge of the memory interface guidelines.

Table 5-5. DDR4 Interface Signals

Clock Signals	Description
CK[1:0]_P\N	Differential Clock signals
Address and Command Signals	
A[15:0]	Address signals
BA[1:0]	Bank Address signals
BG[1:0]	Bank Group signals
WE	Write Enable
RAS_n	Row address strobe
CAS_n	Column address strobe
ACT_n	Activation Command Input

...........continued

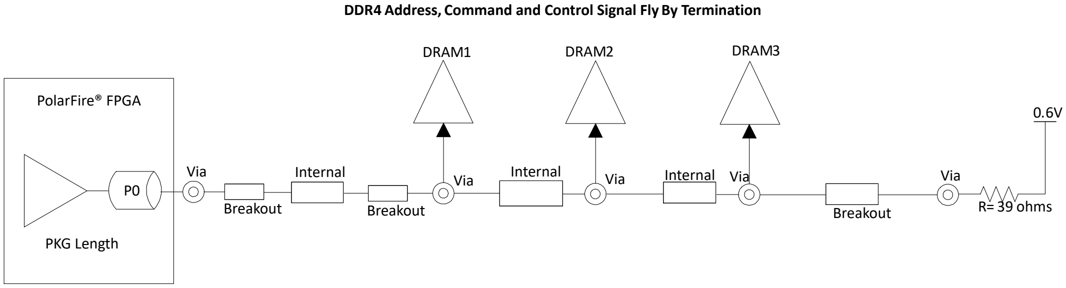
Clock Signals	Description
PAR	Command and address parity
Control Signals	
CKE	Clock Enable
CS_n[1:0]	Chip Select
ODT	On die termination
Reset	Reset
Data Group	
DQ[71:0]	Data signals
DQS[8:0]_P/N	Differential data strobe
M_N[8:0]	Data mask

DDR4 interface on PolarFire supports maximum data rate of 1600 MT/s on HSIO banks. The ADDR/CMD signals are routed in fly-by topology and terminated at the last memory component with respect to VDD/2. CK signals are also routed in fly-by topology and terminated at the last memory component with respect to VDD. The data signals are routed as point to point.

5.3.1.1 DDR4 Routing Topology

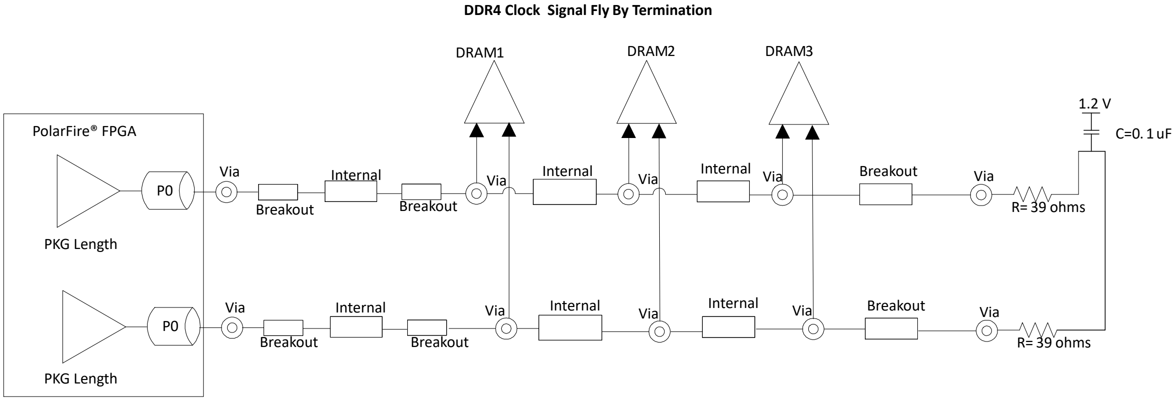
The following figure shows the Flyby termination of DDR4 ADDR/CMD signals.

Figure 5-9. DDR4 ACC Signals FlyBy Termination



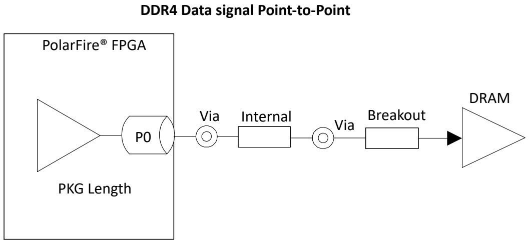
The following figure shows the FlyBy termination of the DDR4 clock signal.

Figure 5-10. DDR4 Clock Signal FlyBy Termination



The following figure shows the point-to-point routing of the DDR4 data signal.

Figure 5-11. DDR4 Data Signal Point-to-Point



5.3.1.2 Skew matching constraints

1. The address and command groups should be skew matched with respect to CK signals within ±100 ps.
2. The DQ, DM signals should be skew matched with respect to DQS within ±25 ps.
3. The read DQ/DQS training will adjust DQS signal to optimal sampling point based on the board skew between DQ and DQS.

6. PolarFire SoC Board Design Recommendations

The GPIO and HSIO of PolarFire SoC devices support various memory interfaces.

- GPIO supports DDR3.
- HSIO supports DDR4, DDR3, LPDDR3, and LPDDR4.

The following table lists the various types of memories supported in PolarFire SoC devices.

Table 6-1. DDR3 and DDR4 Parameters

Parameter	DDR3	LPDDR3	DDR3L	DDR4	LPDDR4
VDDIO	1.5 V	1.2 V	1.35	1.2V	To be updated
VTT, VREF	0.75V	0.6V	0.675V	0.6V	To be updated
Clock, Address, and Command (CAC) Layout	Daisy Chained (Fly-by)	Point to point	Daisy Chained (Fly-by)	Daisy Chained (Fly-by)	To be updated
Data Strobe	Differential	Differential	Differential	Differential	To be updated
ODT	Dynamic	Dynamic	Dynamic	Dynamic	To be updated
Match Addr/CMD/Ctrl to Clock Tightly	Yes	Yes	Yes	Yes	To be updated
Match DQ/DM/DQS Tightly	Yes	Yes	Yes	Yes	To be updated
Match DQS to Clock Loosely	Not Required	Not Required	Not Required	Not Required	To be updated
I/O Standard	SSTL_15	HSUL-12	SSTL_135	POD-12*	To be updated
RZQ	240_1%	240_1%	240_1%	240_1%	To be updated

* HSTL_12 IO standard is used for DDR4 address, command, and control signals.

For more information on DDR memories, see the following documents:

- JESD79-3F JEDEC standard for DDR3 SDRAM specifications
- JESD79-4 JEDEC standard for DDR4 SDRAM specifications
- JESD209-3 JEDEC standard for LPDDR3 SDRAM specifications

6.1 DDR3

The following are the guidelines for connecting the device to the DDR3 memory:

- DDR3 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 60 Ω, and 120 Ω. DQ lines do not need VTT termination. However, VTT termination resistors need to be placed at the end of address and control lines on the PCB.
- Characteristic impedance: Zo is typically 50 Ω, and Zdiff (differential) is 100 Ω.

The following table lists the DDR3 memory interface features supported in PolarFire SoC devices.

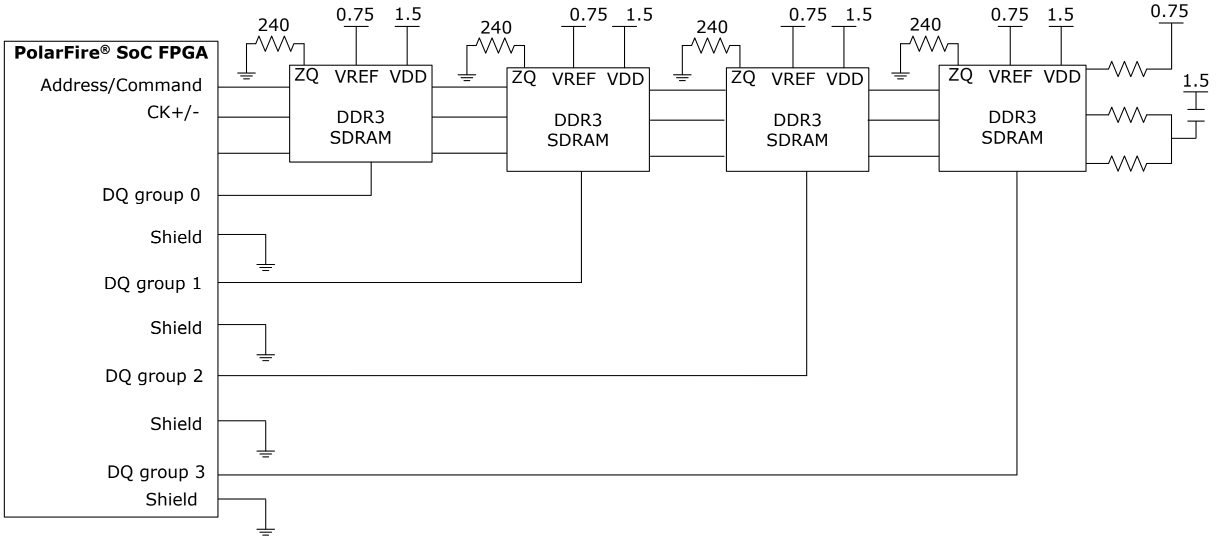
Table 6-2. Interfaces Supported in DDR3 Memory Devices

Interface	DDR3
Voltage	1.5 V
I/O standard	SSTL_15
Data rate	1333 MT/S-HSIO and 1066 MT/S-GPIO

...........continued	
Interface	**DDR3**
Termination	ODT for data group; VTT termination for address, command, and control
Routing topology (CK, ADDR/ CMD, and CONTROL)	Fly-by
Data transmission	Point-to-point

The following figure shows the connectivity between DDR3 devices and PolarFire SoC FPGAs.

Figure 6-1. DDR3 Interface Example



6.1.1 DDR3 Layout Guidelines

This section describes the routing guidelines for DDR3 interface for PolarFire SoC Family. The guidelines are with reference to maximum x72 data width from signal integrity perspective. All the guidelines are provided considering maximum data rate supported. It is recommended to evaluate the interface by performing system level signal integrity simulations. The user is assumed to have the knowledge of the memory interface guidelines.

Table 6-3. DDR3 Interface Signals

Clock Signal	Description
CK[1:0] P/N	Differential clock signals
Control Signals	
CS[1:0]	Chip select
CKE[1:0]	Clock Enable
ODT[1:0]	On die termination enable
Reset_n	Reset signal
Address Signals	
A[12:0]	Address Signals
BA[2:0]	Bank address
Command Signals	
RAS_n	Row address select

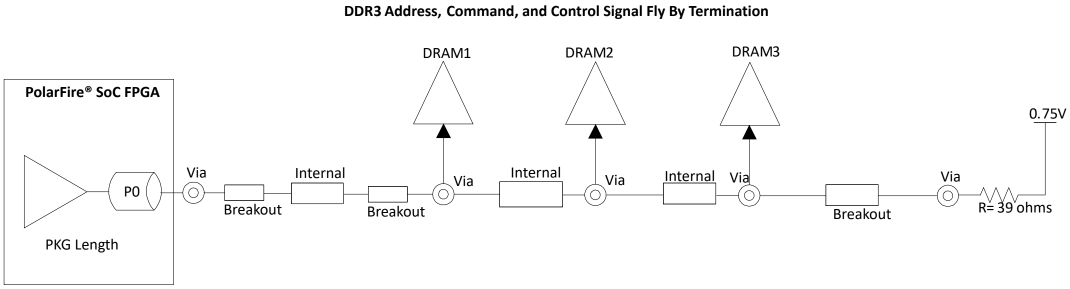
...........continued	
Clock Signal	**Description**
CAS_n	Column address select
WE_n	Write enable
Data Signals	
DQ[31:0]	Data bit
DM[3:0]	Data mask
DQS_P/N[8:0]	Data strobe

Due to high speed signaling, DDR3 uses fly-by routing topology for Address, Command, Control signals to achieve best performance. As shown in the following section, the ADDR/CMD are routed as single ended in fly-by topology. It is recommended to have all the signals being routed in one signal layer to control the skew within the signal groups. The DQ/DQS signals being point to point signals, uses on die termination on memory and FPGA side.

6.1.1.1 DDR3 Routing Topology

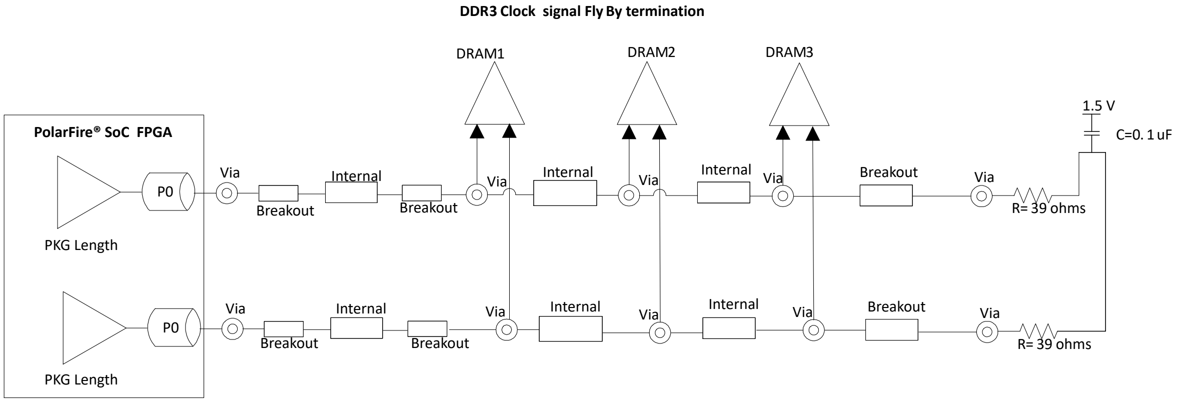
The following figure shows the FlyBy termination of the DDR3 address, command, and control signals (ACC).

Figure 6-2. DDR3 ACC Signals Fly By Termination



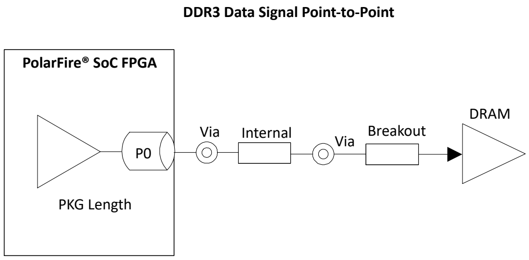
The following figure shows the FlyBy termination of the DDR3 clock signal.

Figure 6-3. DDR3 Clock Signal FlyBy Termination



The following figure shows the Point-to-Point routing of the DDR3 data signal.

Figure 6-4. DDR3 Data Signal Point-to-Point



6.1.1.2 Skew Constraints For Signal Groups

1. The address and command groups must be skew matched with respect to CK signals within ±100 ps.
2. The DQ, DM signals must be skew matched with respect to DQS within ±25 ps.
3. The read DQ/DQS training adjusts the DQS signal to optimal sampling point based on the board skew between DQ and DQS.
4. The write level training ensures that the tDQSS specification for memory is matched for each of the DQ/DQS byte groups and offset the skew introduced due to fly-by routing.

The characteristic impedance of 40 Ω–50 Ω must be used for single ended signals and 90 Ω–100 Ω for differential signals.

6.2 LPDDR3

The following are the guidelines for connecting the device to the LPDDR3 memory:

- LPDDR3 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 120 Ω and 240 Ω.
- DQ lines do not need VTT termination.
- The characteristic impedance, Zo is typically 50 Ω, and Zdiff (differential) is 100 Ω.

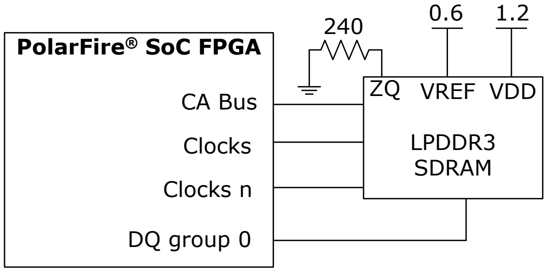
The following table lists the LPDDR3 memory interface features supported in PolarFire SoC devices.

Table 6-4. Interfaces Supported in LPDDR3 Memory Devices

Interface	LPDDR3
Voltage	1.2 V
I/O standard	HSUL-12
Data rate	1333 MT/S-HSIO
Termination	ODT for data group
Routing topology	Point to point

The following figure shows the connectivity between LPDDR3 devices and PolarFire SoC FPGAs.

Figure 6-5. LPDDR3 Interface Example



6.2.1 LPDDR3 Layout Guidelines

The LPDDR3 is a point-to-point interface, where the data group signals and address/command signals are routed from point to point. As LPDDR3 is used in power sensitive designs, there is no termination used for point to point interface. PolarFire SoC supports up to 1333 MT/s for LPDDR3 interface on HSIO only.

It is recommended to tightly length match the signals CK, ADDR/CMD signals in one group and DQS, DQ, DM signals in one group together to reduce the skew.

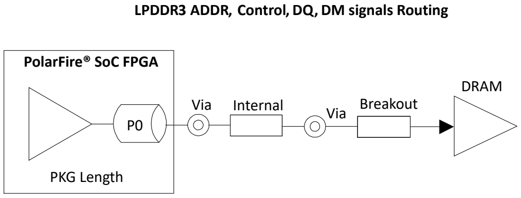
The characteristic impedance of 40 Ω–50 Ω should be used for single ended signals and 90 Ω–100 Ω for differential signals.

Microchip recommends simulating the interface to tightly match the characteristic impedance of trace with driver impedance to reduce reflections and for any possible electrical and timing violations in meeting JEDEC requirements.

6.2.1.1 LPDDR3 Routing Topology

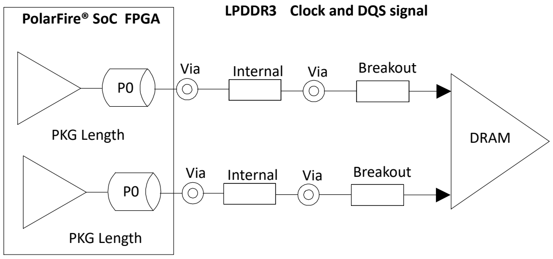
The following figure shows the routing of the LPDDR3 address, control, DQ, and DM signals.

Figure 6-6. LPDDR3 Address/Control/DQ/DM Signal Routing



The following figure shows the routing of LPDDR3 clock and DQS signals.

Figure 6-7. LPDDR3 Clock and DQS signal Routing



6.2.1.2 Skew Matching Constraints

1. Within the data group DQS, DQ, and DM signals must be matched with ±25 ps.
2. Skew between CK, CA, and CS signals within the group must not be more than ±30 ps.
3. Skew between CA and CK group and data lane group must not be more than ±350 ps.

Note: It is recommended to derive the accurate values of skew and termination based on simulations.

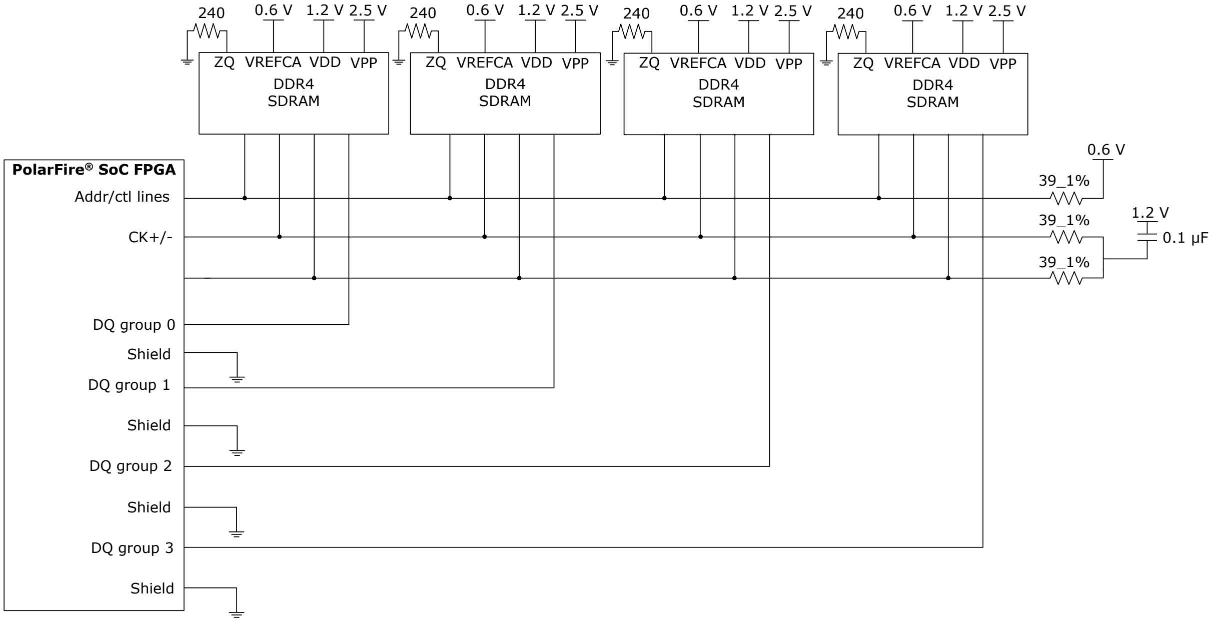
6.3 DDR4

The following are the guidelines for connecting the device to the DDR4 memory:

- DDR4 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 80 Ω, 120 Ω, and 240 Ω. DQ lines do not need VTT termination. However, VTT termination resistors need to be placed at the end of address and control lines on the PCB.
- Characteristic impedance: Zo is typically 50 Ω and Z_{diff} (differential) is 100 Ω.

The following figure shows the features supported by PolarFire SoC FPGA in the DDR4 memory interface.

Figure 6-8. DDR4 Interface Example



6.3.1 DDR4 Layout Guidelines

This section describes the routing guidelines for DDR4 interface for PolarFire SoC Family. The guidelines are with reference to maximum x72 data width from signal integrity perspective. All the guidelines are provided considering maximum data rate supported. It is recommended to evaluate the interface by performing system level signal integrity simulations. The user is assumed to have the knowledge of the memory interface guidelines.

Table 6-5. DDR4 Interface Signals

Clock Signals	Description
CK[1:0]_P\N	Differential Clock signals
Address and Command Signals	
A[12:0]	Address signals
BA[1:0]	Bank Address signals
BG[1:0]	Bank Group signals
WE	Write Enable
RAS_n	Row address strobe
CAS_n	Column address strobe
ACT_n	Activation Command Input

...........continued

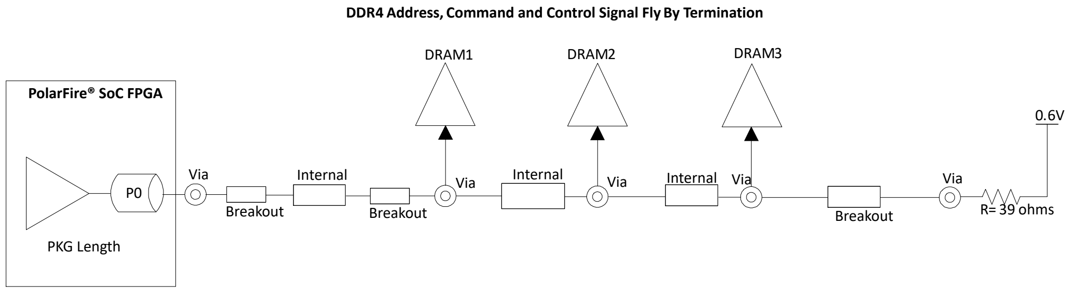
Clock Signals	Description
PAR	Command and address parity
Control Signals	
CKE	Clock Enable
CS_n[1:0]	Chip Select
ODT	On die termination
Reset	Reset
Data Group	
DQ[31:0]	Data signals
DQS[3:0]_P/N	Differential data strobe
DM_N[3:0]	Data mask

DDR4 interface on PolarFire SoC supports maximum data rate of 1600 MT/s on HSIO banks. The ADDR/CMD and CK signals are routed in fly-by topology and they are terminated at the last memory component with respect to VDDI/2. The data signals are routed as point to point.

6.3.2 DDR4 Routing Topology

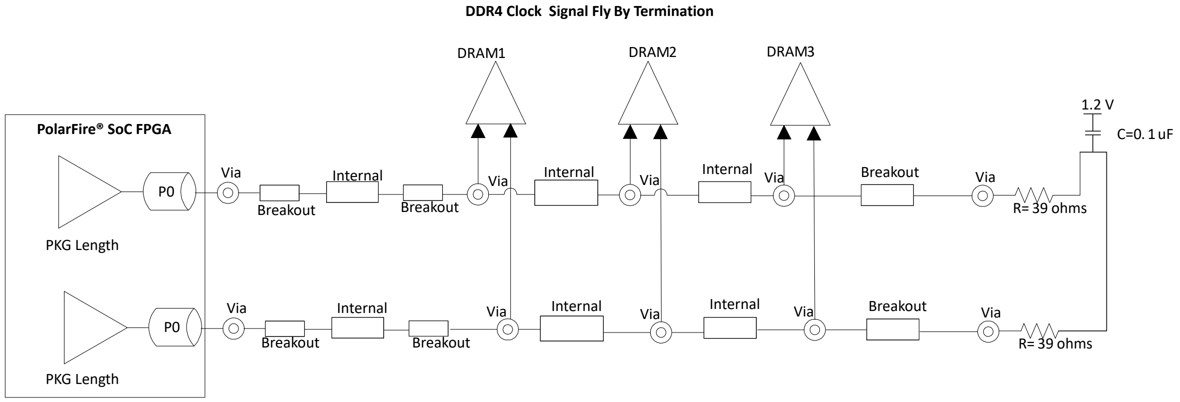
The following figure shows the Flyby termination of DDR4 ADDR/CMD signals.

Figure 6-9. DDR4 ACC Signals FlyBy Termination



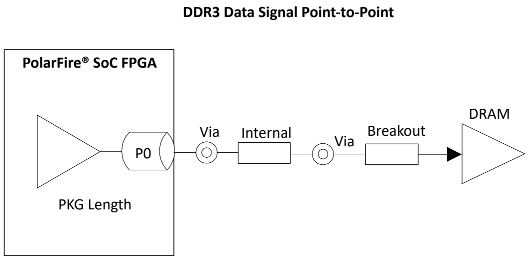
The following figure shows the FlyBy termination of the DDR4 clock signal.

Figure 6-10. DDR4 Clock Signal FlyBy Termination



The following figure shows the point-to-point routing of the DDR4 data signal.

Figure 6-11. DDR4 Data Signal Point-to-Point



6.3.2.1 Skew Matching Constraints

1. The address and command groups should be skew matched with respect to CK signals within ±100 ps.
2. The DQ, DM signals should be skew matched with respect to DQS within ±25 ps.

The read DQ/DQS training will adjust DQS signal to optimal sampling point based on the board skew between DQ and DQS.

7. Debugging Techniques

Although the DDR subsystem eliminates the complexity in DDR SDRAM interface design, debugging memory failures is a challenge. Tracking down functional issues (at the FPGA or system level), functional system interaction problems, system timing issues, and signal fidelity issues between FPGA and memory devices (such as noise, crosstalk, or reflections) becomes much more complex. Care must be taken right from the DDR subsystem configuration phase to the PCB layout phase to achieve the expected performance. This section describes the tools and processes used to debug the DDR subsystem interface.

7.1 Debug Tools

This section provides an overview of the tools available for debugging DDR subsystem.

7.1.1 SmartDebug

Microchip's SmartDebug supports the capability of analyzing and displaying the DDR Training data including DDR IO margining data for all DDR instances used in a design. For more information, see UG0773: PolarFire SmartDebug User Guide.

7.1.2 Identify RTL Debugger

Identify is an RTL debugger that allows you to probe the DDR subsystem (native interface, AXI3/4 slave interface, and DFI interface) interface signals and view the signals in the original RTL code or waveform viewer.

Identify consists of two tools—the Instrumentor and the Debugger. The Instrumentor compiles and inserts the necessary logic into the original RTL to create an instrumented design that allows for the probing of internal signals. The Debugger communicates with the FlashPro device via a JTAG interface to first retrieve the values for probed signals from the programmed FPGA and then display the retrieved signals.

7.1.3 Hardware Probes

PolarFire and PolarFire SoC devices have built-in probe points that enhance the ability to debug design logic within a device through the live and active probe features. These enhanced debug features embedded into the device hardware provide access to logic elements and enable real-time monitoring of inputs and outputs without the need to recompile the design.

- Live probes: Two dedicated probes can be configured to observe a probe point, which is the output of a register. After configuring, the probe data can be sent to two dedicated pins (PROBE_A and PROBE_B), and then to an oscilloscope.
- Active probes: Active probes allow dynamic asynchronous read and write to/from a flip-flop or probe point. This capability helps you to observe the output of the logic and perform quick experiments to determine how the logic is affected if a probe point is written to.

In many cases, the hardware probe capabilities such as live probe can be used along with the Identify RTL debugger and external test techniques to debug issues in a subsystem.

7.2 Design Debug During Simulation

Before a design can be debugged in the hardware, functional simulation must be performed to rule out any RTL issues. Functional issues are independent of process, voltage, temperature, speed grade, and PCB design.

7.3 Hardware Debug

After the design works correctly in simulation, hardware debugging can be started. The first step is to ensure that the guidelines specified in *PolarFire FPGA Board Design User Guide and PolarFire SoC FPGA Board Design User Guide* are followed, see . References. This section lists general checks that can be performed during hardware debug:

- Verify that the parameters (timing, signal width, memory parts, and so on) entered in the configurator are correct.
- Create a simple design that replicates the issue.
- Ensure that all voltages on the board are correctly set and the noise is within the specified limits.
- Ensure that all terminations are adequate.
- Ensure that clock and reset signals are clean and correct.
- Perform a signal integrity (SI) analysis. Key items to be observed during the analysis are:
  - Setup and hold time between data signals (DQ) and their respective DQS
  - Setup and hold time between control/command/address signals and the clock
  - Overshoot and undershoot of all signals with respect to JEDEC specifications
  - DC threshold multi-crossing because of excessive ringing
- Cool and heat the device to understand whether temperature variations are affecting the functionality. Temperature fluctuations may uncover a timing-related issue.
- Run the device at a lower speed. Ensure that the DDR clock frequency is set within the minimal operating frequency of the device.
- Verify that the Libero SoC version and the DDR subsystem IP version are compatible.
- Try debugging the design on a different board of the same revision.

8. Appendix: Timing Parameters

The following table lists the timing parameters used in this document.

Table 8-1. Timing Parameters

Parameter	Description
tXPR	Exit reset from CKE HIGH to a valid command
tINIT1	Minimum CKE LOW time after completion of voltage ramp
tINIT3	Minimum idle time after the first CKE assertion
tINIT5	Maximum duration of device auto initialization
tZQINIT	ZQ initial calibration
tRP	PRECHARGE command period
tRFC	REFRESH-to-ACTIVATE or REFRESH command period

9. Appendix: Fabric DDR Placement Locations

This section lists the DDR placement locations available on PolarFire devices.

Table 9-1. NORTH_NE ANCHOR (HSIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	NORTH_NE	138	64	YES	YES	—	—	✓	✓
	FC784	NORTH_NE	126	64	NO	YES	—	✓	✓	✓
	FC484	NORTH_NE	88	40	YES	NO	✓	✓	✓	—
	FCV484	NORTH_NE	88	40	YES	NO	✓	✓	✓	—
	FCS536	NORTH_NE	88	40	YES	NO	—	✓	✓	—
	FCS325	NORTH_NE	76	32	NO	NO	✓	✓	—	—
DDR4	FC1152	NORTH_NE	139	64	YES	YES	—	—	✓	✓
	FC784	NORTH_NE	127	64	NO	YES	—	✓	✓	✓
	FC484	NORTH_NE	91	40	YES	YES	✓	✓	✓	—
	FCV484	NORTH_NE	91	40	YES	YES	✓	✓	✓	—
	FCS536	NORTH_NE	91	40	YES	YES	—	✓	✓	—
	FCS325	NORTH_NE	79	32	NO	YES	✓	✓	—	—
LPDDR3	FC1152	NORTH_NE	63	32	na	na	—	—	✓	✓
	FC784	NORTH_NE	63	32	na	na	—	✓	✓	✓
	FC484	NORTH_NE	63	32	na	na	✓	✓	✓	—
	FCV484	NORTH_NE	63	32	na	na	✓	✓	✓	—
	FCS536	NORTH_NE	63	32	na	na	—	✓	✓	—
	FCS325	NORTH_NE	63	32	na	na	✓	✓	—	—
QDR X9	FC1152	NORTH_NE	48	9	na	na	—	—	✓	✓
	FC784	NORTH_NE	48	9	na	na	—	✓	✓	✓
	FC484	NORTH_NE	48	9	na	na	✓	✓	✓	—
	FCV484	NORTH_NE	48	9	na	na	✓	✓	✓	—
	FCS536	NORTH_NE	48	9	na	na	—	✓	✓	—
	FCS325	NORTH_NE	48	9	na	na	✓	✓	—	—
QDR X18	FC1152	NORTH_NE	67	18	na	na	—	—	✓	✓
	FC784	NORTH_NE	67	18	na	na	—	✓	✓	✓
	FC484	NORTH_NE	67	18	na	na	✓	✓	✓	—
	FCV484	NORTH_NE	67	18	na	na	✓	✓	✓	—
	FCS536	NORTH_NE	67	18	na	na	—	✓	✓	—
	FCS325	NORTH_NE	67	18	na	na	✓	✓	—	—

...........continued

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
QDR X36	FC1152	NORTH_NE	105	36	na	na	—	—	✓	✓
	FC784	NORTH_NE	105	36	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

(1) If a third party DDR Controller is used that supports data in place of ECC bits, the Max Width will include an additional 8 data bits. For example, a Max Width of 32 with an entry of “Yes” for ECC will become a Max Width of 40 data bits.

Table 9-2. NORTH_NE_OPT ANCHOR (HSIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	NORTH_NE_OPT	138	64	YES	YES	—	—	—	✓
	FC784	NORTH_NE_OPT	138	64	YES	YES	—	—	✓	✓
DDR4	FC1152	NORTH_NE_OPT	139	64	YES	YES	—	—	—	✓
	FC784	NORTH_NE_OPT	139	64	YES	YES	—	—	✓	✓
LPDDR3	FC784	NORTH_NE_OPT	63	32	na	na	—	—	✓	✓
QDR X18	FC784	NORTH_NE_OPT	67	18	na	na	—	—	✓	✓
QDR X36	FC1152	NORTH_NE_OPT	105	36	na	na	—	—	—	✓
	FC784	NORTH_NE_OPT	105	36	na	na	—	—	✓	✓

Table 9-3. NORTH_NW ANCHOR (HSIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	NORTH_NW	138	64	YES	YES	—	—	✓	✓
	FC784	NORTH_NW	126	64	NO	YES	—	✓	✓	✓
	FC484	NORTH_NW	88	40	NO	NO	✓	✓	✓	—
	FCV484	NORTH_NW	88	40	NO	NO	✓	✓	✓	—
	FCS536	NORTH_NW	88	40	NO	NO	—	✓	✓	—
	FCS325	NORTH_NW	76	32	NO	NO	✓	✓	—	—
DDR4	FC1152	NORTH_NW	139	64	YES	YES	—	—	✓	✓
	FC784	NORTH_NW	127	64	NO	YES	—	✓	✓	✓
	FC484	NORTH_NW	91	40	NO	YES	✓	✓	✓	—
	FCV484	NORTH_NW	91	40	NO	YES	✓	✓	✓	—
	FCS536	NORTH_NW	91	40	NO	YES	—	✓	✓	—
	FCS325	NORTH_NW	79	32	NO	YES	✓	✓	—	—

...........continued

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
LPDDR3	FC1152	NORTH_NW	63	32	na	na	—	—	✓	✓
	FC784	NORTH_NW	63	32	na	na	—	✓	✓	✓
	FC484	NORTH_NW	63	32	na	na	✓	✓	✓	—
	FCV484	NORTH_NW	63	32	na	na	✓	✓	✓	—
	FCS536	NORTH_NW	63	32	na	na	—	✓	✓	—
	FCS325	NORTH_NW	63	32	na	na	✓	✓	—	—
QDR X9	FC1152	NORTH_NW	48	9	na	na	—	—	✓	✓
	FC784	NORTH_NW	48	9	na	na	—	✓	✓	✓
	FC484	NORTH_NW	48	9	na	na	✓	✓	✓	—
	FCV484	NORTH_NW	48	9	na	na	✓	✓	✓	—
	FCS536	NORTH_NW	48	9	na	na	—	✓	✓	—
	FCS325	NORTH_NW	48	9	na	na	✓	✓	—	—
QDR X18	FC1152	NORTH_NW	67	18	na	na	—	—	✓	✓
	FC784	NORTH_NW	67	18	na	na	—	✓	✓	✓
	FC484	NORTH_NW	67	18	na	na	✓	✓	✓	—
	FCV484	NORTH_NW	67	18	na	na	✓	✓	✓	—
	FCS536	NORTH_NW	67	18	na	na	—	✓	✓	—
	FCS325	NORTH_NW	67	18	na	na	✓	✓	—	—
QDR X36	FC1152	NORTH_NW	105	36	na	na	—	—	✓	✓
	FC784	NORTH_NW	105	36	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

Table 9-4. NORTH_NW_OPT ANCHOR (HSIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	NORTH_NW_OPT	138	64	YES	YES	—	—	—	✓
	FC784	NORTH_NW_OPT	138	64	YES	YES	—	—	✓	✓
DDR4	FC1152	NORTH_NW_OPT	139	64	YES	YES	—	—	—	✓
	FC784	NORTH_NW_OPT	139	64	YES	YES	—	—	✓	✓
QDR X18	FC784	NORTH_NW_OPT	67	18	na	na	—	—	✓	✓
QDR X36	FC1152	NORTH_NW_OPT	105	36	na	na	—	—		✓
	FC784	NORTH_NW_OPT	105	36	na	na	—	—	✓	✓

Table 9-5. SOUTH_SE ANCHOR (HSIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	SOUTH_SE	52	16	NO	NO	—	—	✓	✓
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
DDR4	FC1152	SOUTH_SE	55	16	NO	YES	—	—	✓	✓
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
LPDDR3	FC1152	SOUTH_SE	63	32	na	na	—	—	✓	✓
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X9	FC1152	SOUTH_SE	48	9	na	na	—	—	✓	✓
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X18	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X36	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

Table 9-6. SOUTH_SE_OPT ANCHOR (HSIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	SOUTH_SE_OPT	88	40	NO	NO	—	—	—	✓
DDR4	FC1152	SOUTH_SE_OPT	91	40	NO	YES	—	—	—	✓
QDR X18	FC1152	SOUTH_SE_OPT	67	18	na	na	—	—	—	✓

Table 9-7. SOUTH_SW ANCHOR (GPIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	SOUTH_SW	88	40	NO	NO	—	—	✓	✓
	FC784	SOUTH_SW	88	40	NO	NO	—	✓	✓	✓
	FC484	SOUTH_SW	76	32	NO	NO	✓	✓	✓	—
	FCV484	SOUTH_SW	88	40	NO	NO	✓	✓	✓	—
	FCS536	SOUTH_SW	88	40	NO	NO	—	✓	✓	—
	FCS325	SOUTH_SW	40	8	NO	NO	✓	✓	—	—
DDR4	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
LPDDR3	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X9	FC1152	SOUTH_SW	48	9	na	na	—	—	✓	✓
	FC784	SOUTH_SW	48	9	na	na	—	✓	✓	✓
	FC484	SOUTH_SW	48	9	na	na	✓	✓	✓	—
	FCV484	SOUTH_SW	48	9	na	na	✓	✓	✓	—
	FCS536	SOUTH_SW	48	9	na	na	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X18	FC1152	SOUTH_SW	67	18	na	na	—	—	✓	✓
	FC784	SOUTH_SW	67	18	na	na	—	✓	✓	✓
	FC484	SOUTH_SW	67	18	na	na	✓	✓	✓	—
	FCV484	SOUTH_SW	67	18	na	na	✓	✓	✓	—
	FCS536	SOUTH_SW	67	18	na	na	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—

...........continued

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
QDR X36	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

Table 9-8. WEST_NW ANCHOR (GPIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	WEST_NW	136	64	YES	NO	—	—	✓	✓
	FC784	WEST_NW	124	64	NO	NO	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	WEST_NW	52	16	NO	NO	✓	✓	✓	—
	FCS536	WEST_NW	76	32	NO	NO	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—
DDR4	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
LPDDR3	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X9	FC1152	WEST_NW	48	9	na	na	—	—	✓	✓
	FC784	WEST_NW	48	9	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	WEST_NW	48	9	na	na	✓	✓	✓	—
	FCS536	WEST_NW	48	9	na	na	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—

...........continued

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
QDR X18	FC1152	WEST_NW	67	18	na	na	—	—	✓	✓
	FC784	WEST_NW	67	18	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	WEST_NW	67	18	na	na	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X36	FC1152	WEST_NW	105	36	na	na	—	—	✓	✓
	FC784	WEST_NW	105	36	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

Table 9-9. WEST_NW_OPT ANCHOR (GPIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	WEST_NW_OPT	136	64	YES	NO	—	—	—	✓
QDR X36	FC1152	WEST_NW_OPT	105	36	na	na	—	—	—	✓

Table 9-10. WEST_SW ANCHOR (GPIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	WEST_SW	126	64	NO	YES	—	—	✓	✓
	FC784	WEST_SW	88	40	NO	NO	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	WEST_SW	52	16	NO	NO	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—
DDR4	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

...........continued

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
LPDDR3	FC1152	na	na	na	na	na	—	—	—	—
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X9	FC1152	WEST_SW	48	9	na	na	—	—	✓	✓
	FC784	WEST_SW	48	9	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	WEST_SW	48	9	na	na	—	✓	✓	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X18	FC1152	WEST_SW	67	18	na	na	—	—	✓	✓
	FC784	WEST_SW	67	18	na	na	—	✓	✓	✓
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—
QDR X36	FC1152	WEST_SW	105	36	na	na	—	—	✓	✓
	FC784	na	na	na	na	na	—	—	—	—
	FC484	na	na	na	na	na	—	—	—	—
	FCV484	na	na	na	na	na	—	—	—	—
	FCS536	na	na	na	na	na	—	—	—	—
	FCS325	na	na	na	na	na	—	—	—	—

Table 9-11. WEST_SW_OPT ANCHOR (GPIO)

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
DDR3	FC1152	WEST_SW_OPT	138	64	YES	YES	—	—	—	✓
	FC784	WEST_SW_OPT	88	40	NO	NO	—	—	✓	✓
	FC484	WEST_SW_OPT	52	16	NO	NO	—	✓	✓	—
	FCV484	WEST_SW_OPT	52	16	NO	NO	—	✓	—	—
QDR X9	FC1152	WEST_SW_OPT	48	9	na	na	—	—	—	✓
	FC784	WEST_SW_OPT	48	9	na	na	—	—	✓	✓
	FC484	WEST_SW_OPT	48	9	na	na	—	✓	✓	—
	FCV484	WEST_SW_OPT	48	9	na	na	—	✓	✓	—

...........continued

Standard	Package	Edge_Anchor	Max Pin Count	Max Width	ECC AT Max Width	Add Cmd Parity Check	MPF100T/TS	MPF200T/TS	MPF300T/TS	MPF500T/TS
QDR X18	FC1152	WEST_SW_OPT	67	18	na	na	—	—	—	✓
	FC784	WEST_SW_OPT	67	18	na	na	—	—	✓	✓
QDR X36	FC1152	WEST_SW_OPT	105	36	na	na	—	—	—	✓
	FC784	WEST_SW_OPT	105	36	na	na	—	—	✓	✓

10. Revision History

The revision history table describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
A	08/2021	The first publication of this document. This user guide was created by merging the following documents: • UG0676: PolarFire FPGA Memory Controller User Guide • UG0906: PolarFire SoC FPGA Memory Controller User Guide

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-8761-6

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-72884388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820