

## Introduction (Ask a Question)

This document describes the memory controller solutions in the PolarFire<sup>®</sup> device family. The PolarFire family consists of the following FPGA devices.

**PolarFire FPGAs** Microchip's PolarFire<sup>®</sup> 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<sup>®</sup> Gen 2 (EP/RP), and, on select data security (S) devices, an integrated low-power crypto co-processor.

**PolarFire SoC FPGAs** 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<sup>®</sup>. 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.

**RT PolarFire FPGAs** Microchip's RT PolarFire<sup>®</sup> FPGAs combine our 60 years of space flight heritage with the industry's lowest-power PolarFire FPGA family to enable new capabilities for space and mission-critical applications. RT PolarFire FPGA family includes RTPF500T, RTPF500TL, RTPF500TS, RTPF500TLS, RTPF500ZT, RTPF500ZTL, RTPF500ZTS, and RTPF500ZTLS devices.

The following table lists the memory controller IP solutions in the PolarFire family. These memory controller solutions address the high-speed data transfer requirements for a wide range of applications and code execution.

**Table 1.** Memory Controller IP Solutions in the PolarFire<sup>®</sup> Family

Memory Controller	PolarFire <sup>®</sup> FPGA (MPF)	PolarFire SoC FPGA (MPFS)	RT PolarFire <sup>®</sup> FPGA (RTPF)	Description
Embedded MSS DDR Controller (see <a href="#">MSS DDR Memory Controller</a> )	—	✓	—	Supports DDR3, DDR3L, DDR4, LPDDR3, and LPDDR4 memory devices.
PolarFire DDR IP <sup>1</sup> (see <a href="#">Fabric DDR Subsystem</a> )	✓	✓	✓	Supports DDR3, DDR3L, DDR4, LPDDR3 <sup>2</sup> memory devices.
PolarFire QDR IP <sup>1</sup> (see <a href="#">QDR Memory Controller</a> )	✓	✓	✓	Supports QDR II+ and QDR II+ Xtreme memory devices.
PolarFire Octal DDR PHY <sup>1</sup> (see <a href="#">Octal DDR PHY-Only Solution</a> )	✓	✓	✓	Supports xSPI (JESD251), HyperBUS, and ONFI memory devices.



### Important:

1. These IPs belong to the PolarFire family and can be seamlessly used in RT PolarFire and PolarFire SoC designs.
2. RT PolarFire devices do not support LPDDR memories.

## 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 [Standalone MSS Configurator User Guide for PolarFire SoC](#).
- For more information about the clocking resources, see [PolarFire Family Clocking Resources User Guide](#).
- For more information about SmartDebug, see [SmartDebug User Guide](#).
- For more information about PolarFire board design, see [PolarFire FPGA Board Design User Guide](#).
- For more information about PolarFire board design, see [RT PolarFire FPGA Board Design User Guide](#).
- For more information about PolarFire SoC board design, see [PolarFire SoC FPGA Board Design Guidelines User Guide](#)

## Acronyms [\(Ask a Question\)](#)

**Table 2.** Acronyms

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

# Table of Contents

Introduction.....	1
Acronyms.....	2
1. MSS DDR Memory Controller.....	6
1.1. Training Logic for the PolarFire SoC MSS DDR Controller .....	6
1.2. Maximum Density Supported by the PolarFire SoC MSS DDR Controller.....	6
2. Fabric DDR Subsystem.....	9
2.1. Features.....	10
2.2. Performance.....	11
2.3. Resource Utilization.....	11
2.4. Functional Description.....	13
2.4.1. DDR Controller.....	14
2.4.2. Clocking Structure.....	23
2.4.3. Integrated PHY.....	23
2.5. DDR Subsystem Ports.....	25
2.5.1. DDR PHY-Only Solution Ports.....	30
2.5.2. DFI Interface.....	31
2.6. Functional Timing Diagrams.....	34
2.6.1. Native Interface Transactions.....	34
2.6.2. AXI Transactions.....	37
2.7. DDR PHY-Only Solution Integration.....	40
2.7.1. Controller and PHY Integration.....	41
2.7.2. PHY Configurator.....	44
2.7.3. Initialization and Bring-Up of the DDR Subsystem.....	47
2.7.4. Timing.....	59
2.8. Octal DDR PHY-Only Solution.....	59
2.8.1. Configurator.....	60
2.8.2. Timing Constraints.....	67
2.8.3. Place and Route Pin-Out Rules.....	67
2.9. Implementation.....	68
2.9.1. Selecting the Memory Device.....	68
2.9.2. Configuring the DDR3 Subsystem.....	69
2.9.3. Configuring the LPDDR3 Subsystem.....	79
2.9.4. Configuring the DDR4 Subsystem.....	88
2.9.5. Simulating the DDR Subsystem.....	102
2.9.6. Design Constraints.....	102
2.10. Functional Examples.....	106
2.10.1. Accessing DDR Subsystem Through AXI4 Interface.....	106
2.10.2. Accessing DDR Subsystem Through Native Interface.....	109
2.10.3. Accessing DDR Memory Using a Third-Party DDR Controller and the DDR PHY-Only Solution .....	111
3. Initialization Sequence.....	112
3.1. DDR3.....	112
3.2. DDR4.....	113

3.3.	LPDDR3.....	115
3.4.	LPDDR4 (For PolarFire SoC FPGA Only).....	116
4.	Training Logic.....	118
5.	QDR Memory Controller.....	122
5.1.	Features.....	122
5.2.	Performance.....	122
5.2.1.	Read Latency.....	123
5.3.	Functional Description.....	123
5.3.1.	QDR Controller.....	123
5.3.2.	Clocking Structure.....	127
5.3.3.	Port List.....	128
5.4.	Functional Timing Diagrams.....	130
5.4.1.	Write Strobe Mapping.....	130
5.4.2.	QDR SRAM Write.....	134
5.4.3.	QDR SRAM Read.....	135
5.5.	Implementation.....	136
5.5.1.	Configuring QDR Memory Controller IP.....	136
5.5.2.	Simulating QDR Memory Controller.....	138
5.5.3.	Design Constraints.....	139
5.5.4.	QDR Resource Utilization.....	140
5.6.	Functional Example.....	144
5.6.1.	RTL_FULL Simulation.....	144
5.6.2.	BFM Simulation.....	146
5.6.3.	RTL_Fast Simulation.....	147
5.7.	PCB Recommendations.....	148
6.	PolarFire Board Design Recommendations.....	149
6.1.	DDR3.....	149
6.1.1.	DDR3 Layout Guidelines.....	150
6.2.	LPDDR3.....	152
6.2.1.	LPDDR3 Layout Guidelines.....	153
6.3.	DDR4.....	154
6.3.1.	DDR4 Layout Guidelines.....	154
7.	PolarFire SoC Board Design Recommendations.....	157
7.1.	DDR3.....	157
7.1.1.	DDR3 Layout Guidelines.....	158
7.2.	LPDDR3.....	160
7.2.1.	LPDDR3 Layout Guidelines.....	161
7.3.	DDR4.....	162
7.3.1.	DDR4 Layout Guidelines.....	162
7.3.2.	DDR4 Routing Topology.....	163
7.4.	LPDDR4.....	164
7.4.1.	LPDDR4 Layout Guidelines.....	166
7.4.2.	LPDDR4 Routing Topology.....	167
7.4.3.	Designing LPDDR4 Interface with PolarFire SoC FPGA.....	168
8.	Debugging Techniques.....	171

8.1. Debug Tools.....	171
8.1.1. SmartDebug.....	171
8.1.2. Identify RTL Debugger.....	171
8.1.3. Hardware Probes.....	171
8.2. Design Debug During Simulation.....	172
8.3. Hardware Debug.....	172
A. Appendix: Supported Memory Configurations.....	173
B. Appendix: Timing Parameters.....	174
C. Appendix: Fabric DDR Placement Locations.....	175
Revision History.....	182
Microchip FPGA Support.....	188
Microchip Information.....	188
Trademarks.....	188
Legal Notice.....	188
Microchip Devices Code Protection Feature.....	189

## 1. MSS DDR Memory Controller

The PolarFire SoC MSS includes an embedded 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, DDR3L, 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 [Standalone MSS Configurator User Guide for PolarFire SoC](#).

For debugging LPDDR4 and DDR4 training issues, use [Memory Log Analyzer](#) GUI.

### 1.1. Training Logic for the PolarFire SoC MSS DDR Controller [\(Ask a Question\)](#)

The training logic performs the following operations.

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

For more information, see [Training Logic](#).

For information about initialization sequence of DDR3, DDR4, LPDDR3, and LPDDR4, see [Initialization Sequence](#).

### 1.2. Maximum Density Supported by the PolarFire SoC MSS DDR Controller [\(Ask a Question\)](#)

The MSS DDR Controller requires that the minimum Refresh Cycle Interval ( $t_{REFI}$ ) must be greater than 4 times the Refresh Cycle time ( $t_{RFC}$ ). The Refresh Cycle Interval is the period between successive refresh commands, and the Refresh Cycle time is the duration of a refresh operation. DDR manufacturers specify the  $t_{REFI}$  for the DDR memory's operating case temperature. The preceding requirements restrict the maximum supported DDR die density at higher temperatures for DDR4 and LPDDR4. This restriction does not apply to DDR3 and LPDDR3 memories.

The Refresh Cycle time is proportional to the memory die density. The following table lists the maximum supported DDR die density at different case temperatures.

**Table 1.1.** PolarFire SoC FPGA Maximum Die Density

Device	Maximum Die Density			
Memory Case Temperature	$-40 < T_c^1 < 85 \text{ }^\circ\text{C}$	$-40 < T_c < 95 \text{ }^\circ\text{C}$	$-40 < T_c < 105 \text{ }^\circ\text{C}$	$-40 < T_c < 125 \text{ }^\circ\text{C}$
DDR4	16 Gb	16 Gb	8 Gb	2 Gb
LPDDR4	16 Gb	4 Gb	4 Gb	4 Gb

**Note:**

1.  $T_c$  is the case temperature of the DDR Memory.

DDR memory vendors integrate one or more dies in a single component to create memory modules of higher density, and one can use several components to create a data width of x32. The MSS DDR controller is designed to support dual ranks for DDR4.

The maximum supported memory density is based on the following criteria:

- The DDR memory component die size must be within the maximum die density mentioned in [Table 1.1](#) for a given DDR operating case temperature
- More than one component or die can be used to enhance data width to up to x32 for DDR4

- Dual Channel can be used to enhance data width to up to x32 for LPDDR4
- Dual Rank is supported for DDR4


The following table lists the maximum supported memory density at different case temperatures, for the PolarFire SoC MSS DDR controller.

**Table 1.2.** PolarFire SoC FPGA Maximum Supported Memory Density

Device	Maximum Memory Density			
Memory Case Temp	$-40 < T_c^1 < 85\text{ }^\circ\text{C}$	$-40 < T_c < 95\text{ }^\circ\text{C}$	$-40 < T_c < 105\text{ }^\circ\text{C}$	$-40 < T_c < 125\text{ }^\circ\text{C}$
DDR4	128 Gb	128 Gb	32 Gb	8 Gb
LPDDR4	32 Gb	8 Gb	8 Gb	8 Gb

**Note:**

1.  $T_c$  is the case temperature of the DDR Memory.

 **Important:** FPGA temperature grades are denoted by " $T_j$ ", the FPGA junction temperature. Therefore, FPGA " $T_j$ " is different from the DDR memory " $T_c$ ". Typically, " $T_j$ " is higher than " $T_c$ ".

**Memory Selection Guidelines**

The following examples provide the memory selection guidelines for DDR4 and LPDDR4.

• **Example 1: DDR4**

**Requirement:** 4 GB (32 Gb) DDR4 memory at an operating case temperature ( $T_c$ ) of up to 105 °C

**Guideline:** A memory device that uses a 16 Gb die in 2Gx8 configuration must not be selected as the maximum supported die size at  $T_c = 105\text{ }^\circ\text{C}$  is 8 Gb. Instead, you can choose one of the following options:

- **Option A:** Stack 2 components of DDR memory that uses 8 Gb die in 1Gx8 configuration and dual ranks
- **Option B:** Stack 4 components of DDR memory that uses 8 Gb die in 1Gx8 configuration and single rank

• **Example 2: LPDDR4**

**Requirement:** 1 GB (8 Gb) LPDDR4 memory at an operating case temperature ( $T_c$ ) of up to 95 °C

**Guideline:** A memory device that uses 8 Gb die in a 512Mx16 configuration must not be selected as the maximum supported die size at  $T_c = 95\text{ }^\circ\text{C}$  is 4 Gb. Instead, choose a device that uses 4 Gb die in 256Mx32 (two channels of 16-bit data width) to build the 8 Gb LPDDR4 memory.

Some examples of possible memory configurations are provided in the following table.

**Table 1.3.** Memory Configuration Examples

Memory Type	Configuration <sup>1</sup>	Number of Components <sup>2</sup>	Component Density <sup>3</sup>	Total Data Width	Rank/Channel	Total Density
DDR4 (Twin Die)	x8	4	16 Gb	x32	Dual Rank	128 Gb
DDR4	x8	4	16 Gb	x32	Single Rank	64 Gb
DDR4	x8	4	8 Gb	x32	Single Rank	32 Gb
LPDDR4 (Twin Die)	x16	1	16 Gb	x32	Dual Channel	32 Gb
LPDDR4 (Twin Die)	x16	1	4 Gb	x32	Dual Channel	8 Gb

**Notes:**

1. For DDR4, “Configuration” indicates the data bus width of the Chip. For LPDDR4, “Configuration” indicates the width of the channel.
2. “Number of Components” indicates the quantity of memory chips needed.
3. “Component Density” indicates the die size in the memory.

## 2. 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. This solution is supported only for DDR3, DDR3L, DDR4, and LPDDR3 memory devices.
- 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 2.1. Fabric DDR Subsystem

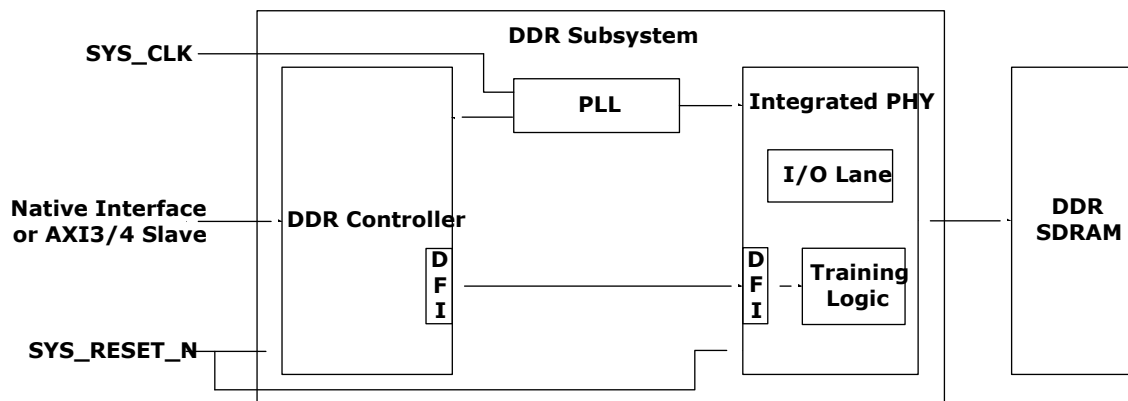


Figure 2.2. DDR PHY-Only Solution

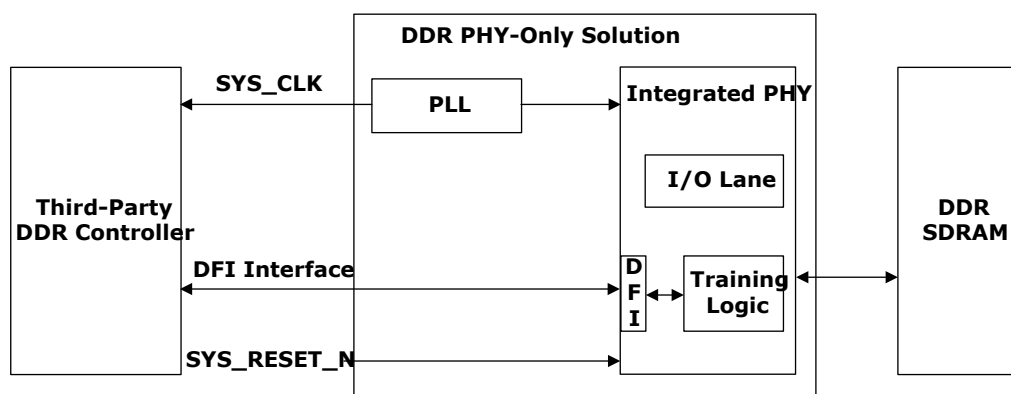
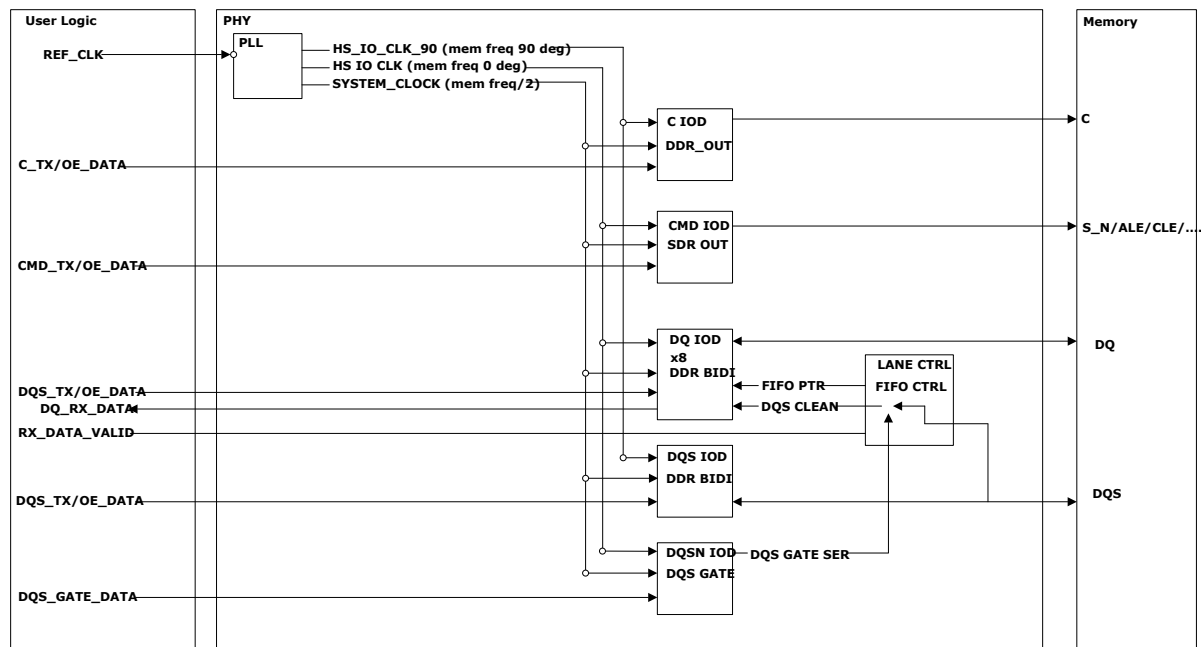


Figure 2.3. Octal DDR PHY-Only Solution



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.

## 2.1. Features [\(Ask a Question\)](#)

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 (SECEDED) Error-Correcting Code (ECC) protection.

## 2.2. Performance [\(Ask a Question\)](#)

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

**Table 2.1.** 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



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

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

**Table 2.2.** 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 <sup>1</sup> )	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 <sup>1</sup> )	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.

## 2.3. Resource Utilization [\(Ask a Question\)](#)

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

**Table 2.3.** DDR3 Resource Utilization

Interface Type	DDR Width	AXI Width	Maximum Number of LUT4	Maximum Number of DFF	Maximum Number of $\mu$ SRAM	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
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 2.4.** DDR4 Resource Utilization

Interface Type	DDR Width	AXI Width	Maximum Number of LUT4	Maximum Number of DFF	Maximum Number of $\mu$ SRAM	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	
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
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

## 2.4. Functional Description [\(Ask a Question\)](#)

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) 4.0 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 Family Clocking Resources User Guide](#).

### 2.4.1. DDR Controller [\(Ask a Question\)](#)

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

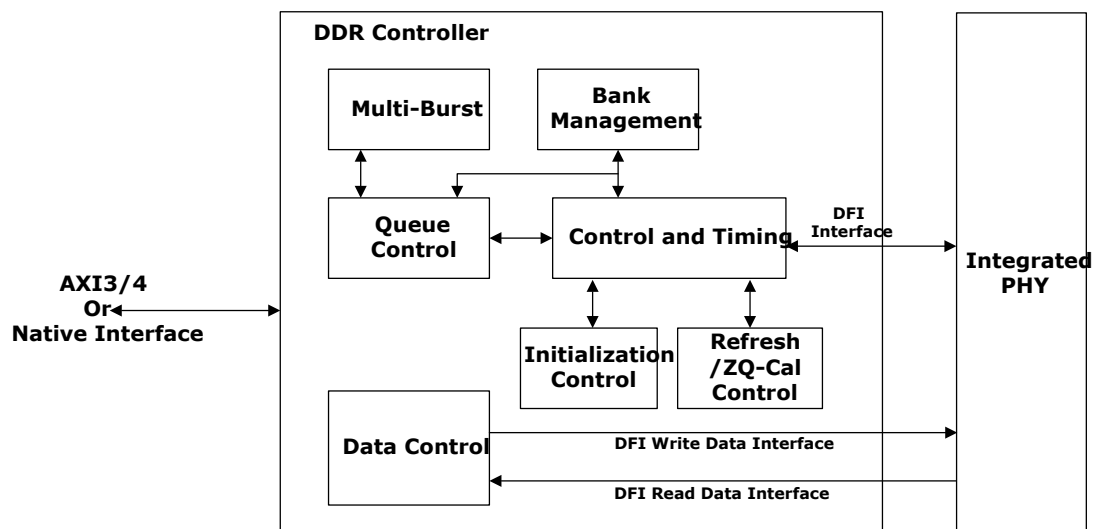
- Control and timing block—contains the main controller logic.
- Initialization control block—performs the initialization sequence after system reset (RESET\_N) is deactivated or when dynamic reinitialization control (CTRLR\_INIT) is pulsed.
- Bank management block—keeps track of the last opened row 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 2.4.** 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](#)

#### 2.4.1.1.1. Periodic Functions [\(Ask a Question\)](#)

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

##### 2.4.1.1.1.1.Refresh [\(Ask a Question\)](#)

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.

##### 2.4.1.1.1.2.ZQ Calibration [\(Ask a Question\)](#)

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 2.5.** 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.

The PolarFire device family supports I/O calibration requests from the Fabric DDR controller, and calibrate only the I/Os that are used with the Fabric DDR controller.

#### 2.4.1.1.2. Look-Ahead Operations (Precharge and Auto-Precharge) [\(Ask a Question\)](#)

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

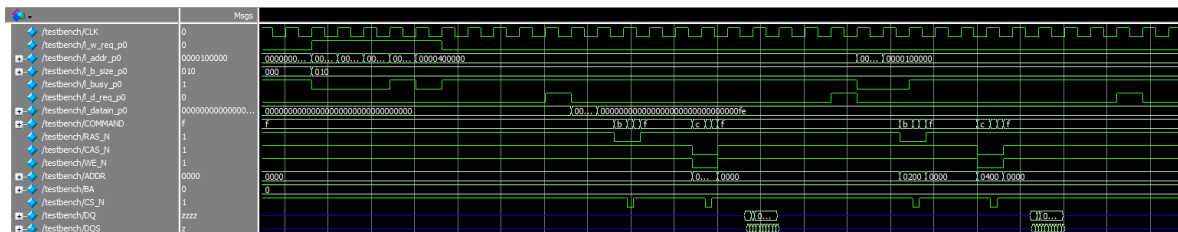
detects that a precharge to that bank is necessary for a read or write request currently pending in the command queue.

### ➔ Important:

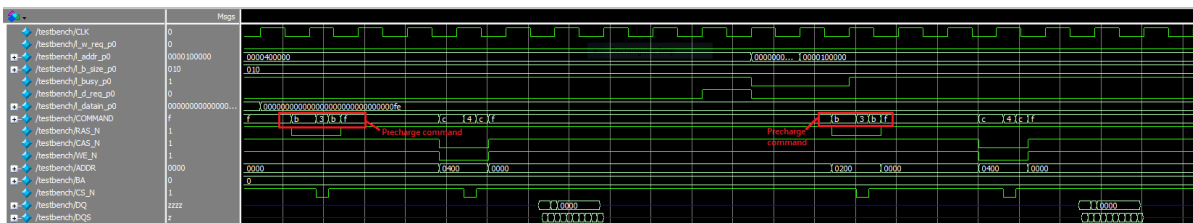
1. By default, Look-Ahead operations are not enabled, they can be enabled using DDR configurator. Look-Ahead operations effect the timing closure of the DDR subsystem.
2. SDRAM commands are issued only for every 4 memory clocks. The timing diagrams shown in this document are only for general understanding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

The following figures show 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.

**Figure 2.5.** Sample Read and Write Sequence With Look Ahead Precharge (Full Wave)



**Figure 2.6.** Sample Read and Write Sequence With Look Ahead Precharge (Zoomed)



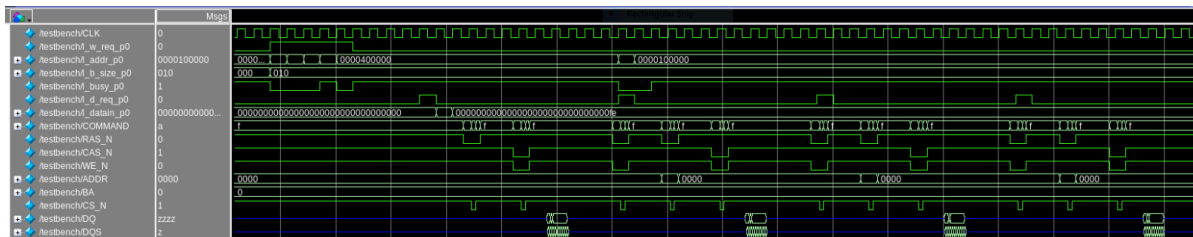
The following points summarize a read-write sequence with Look-Ahead Precharge:

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. Due to 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 2.6](#).
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.

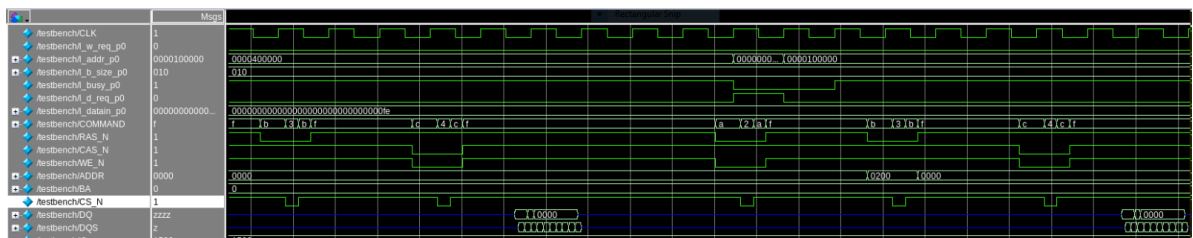
7. The written data begins to appear on the SDRAM bus on the DQ lines.
8. L\_W\_VALID is asserted, and written data appears on the native interface, and then L\_W\_VALID is deasserted.
9. The read operation follows the same sequence.

The following figures show a sample timing diagram without the Look-Ahead Precharge feature.

**Figure 2.7.** Sample Read and Write Sequence Without Look Ahead Precharge (Full Wave)



**Figure 2.8.** 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. Due to 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 2.8](#).
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.

#### 2.4.1.3. Bank Management [\(Ask a Question\)](#)

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

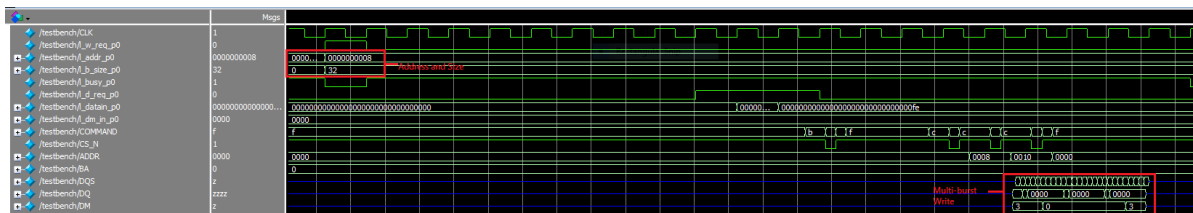
up to 100% memory throughput for sequential accesses (not including refresh and ZQ calibration commands).

#### 2.4.1.4. Multi-Burst Capability (Native-Interface Mode Only) [\(Ask a Question\)](#)

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 [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 2.9.** Sample Write Sequence Demonstrating Multi-Burst Operation

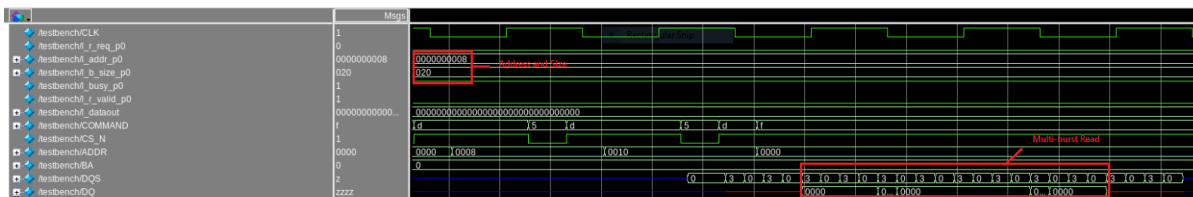


The following points summarize a write sequence with multi-burst capability:

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 2.10.** 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.

#### 2.4.1.5. Data Ordering with Non-Aligned Starting Addresses [\(Ask a Question\)](#)

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.

#### 2.4.1.6. Auto-Precharge [\(Ask a Question\)](#)

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 2.9](#) and [Figure 2.10](#).

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.

#### 2.4.1.7. ECC [\(Ask a Question\)](#)

When ECC is enabled, the DDR controller computes an 8-bit ECC for every 64-bit data to support SECCDED. 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. The following table lists the mapping of the ECC\_ERROR\_POS bits to the error bit position.

**Table 2.6.** Mapping of ECC Position bits to Error Bit Position

ECC_ERROR_POS[6:0]	Error Bit Position
7'h80	127 (check-bit 7)
7'h40	126 (check-bit 6)
7'h20	125 (check-bit 5)
7'h10	124 (check-bit 4)
7'h08	123 (check-bit 3)
7'h04	122 (check-bit 2)
7'h02	121 (check-bit 1)
7'h01	120 (check-bit 0)
7'd127	119
7'd126	118
7'd125	117
7'd124	116
7'd123	115
7'd122	114
7'd121	113
7'd120	112
7'd119	111
7'd118	110
7'd117	109
7'd116	108
7'd115	107
7'd114	106
7'd113	105
7'd112	104
7'd111	103
7'd110	102
7'd109	101
7'd108	100
7'd107	99
7'd106	98
7'd105	97
7'd104	96
7'd103	95
7'd102	94
7'd101	93
7'd100	92
7'd99	91
7'd98	90
7'd97	89
7'd96	88
7'd95	87
7'd94	86
7'd93	85
7'd92	84
7'd91	83

**Table 2.6.** Mapping of ECC Position bits to Error Bit Position (continued)

ECC_ERROR_POS[6:0]	Error Bit Position
7'd90	82
7'd89	81
7'd88	80
7'd87	79
7'd86	78
7'd85	77
7'd84	76
7'd83	75
7'd82	74
7'd81	73
7'd80	72
7'd79	71
7'd78	70
7'd77	69
7'd76	68
7'd75	67
7'd74	66
7'd73	65
7'd72	64
7'd71	63
7'd70	62
7'd69	61
7'd68	60
7'd67	59
7'd66	58
7'd65	57
7'd63	56
7'd62	55
7'd61	54
7'd60	53
7'd59	52
7'd58	51
7'd57	50
7'd56	49
7'd55	48
7'd54	47
7'd53	46
7'd52	45
7'd51	44
7'd50	43
7'd49	42
7'd48	41
7'd47	40
7'd46	39
7'd45	38

**Table 2.6.** Mapping of ECC Position bits to Error Bit Position (continued)

ECC_ERROR_POS[6:0]	Error Bit Position
7'd44	37
7'd43	36
7'd42	35
7'd41	34
7'd40	33
7'd39	32
7'd38	31
7'd37	30
7'd36	29
7'd35	28
7'd34	27
7'd33	26
7'd31	25
7'd30	24
7'd29	23
7'd28	22
7'd27	21
7'd26	20
7'd25	19
7'd24	18
7'd23	17
7'd22	16
7'd21	15
7'd20	14
7'd19	13
7'd18	12
7'd17	11
7'd15	10
7'd14	9
7'd13	8
7'd12	7
7'd11	6
7'd10	5
7'd9	4
7'd7	3
7'd6	2
7'd5	1
7'd3	0

**Notes:**

- The SECDED bits of the fabric DDR Controller for DDR3 and DDR4 will always be 8 for 32 or 64 data width.
- The ECC status signals are supported for DDR3, LPDDR3, and DDR4.

### 2.4.2. Clocking Structure [\(Ask a Question\)](#)

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.

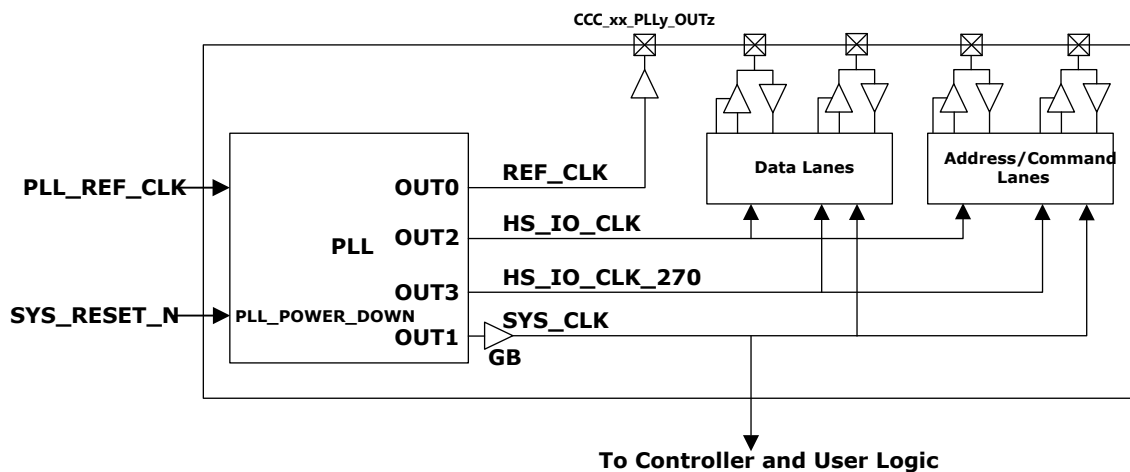
The PLL generates the following required clocks:

- 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 2.11.** DDR Clocking Structure

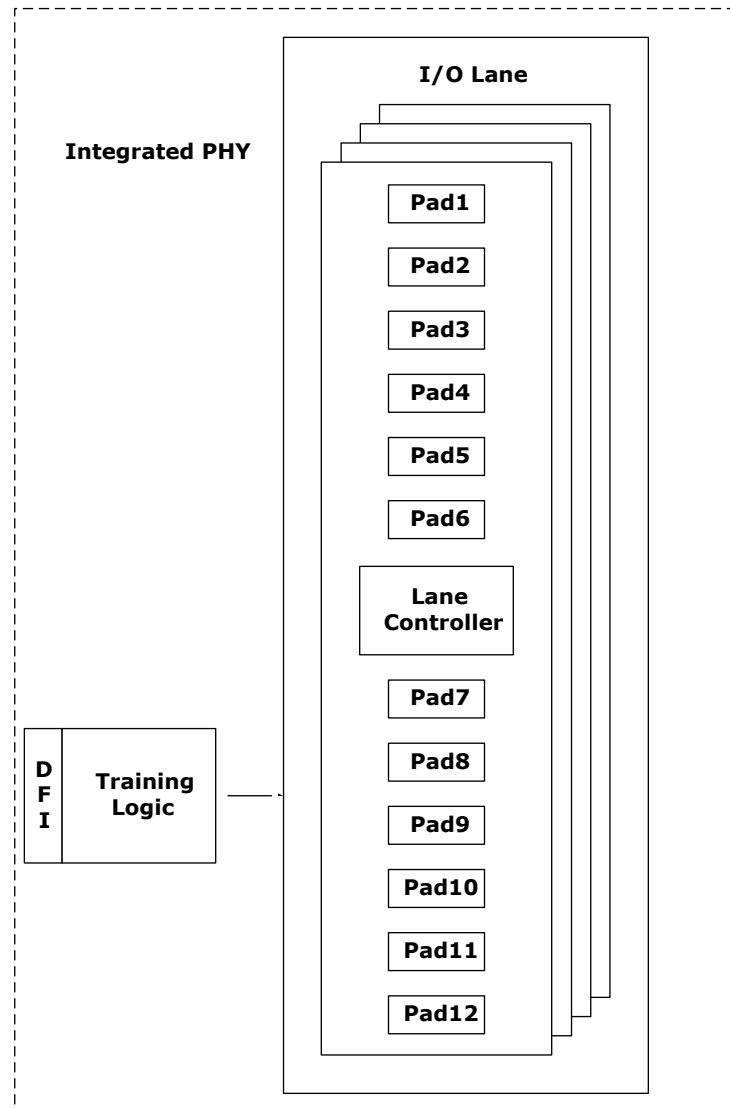


**Note:** The dedicated clock output pad, CCC\_xx\_PLLy\_OUTz, is selected depending on the DDR subsystem placement constraint, where xx = SE/NE/NW/SW, y = 0/1, and z = 0/1.

### 2.4.3. Integrated PHY [\(Ask a Question\)](#)

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 4.0 training requests between the I/O lane and the DDR controller. For more information about the DDR PHY-only integration, see [DDR PHY-Only Solution Integration](#). For more information about Octal DDR PHY-Only Solution, see [Octal DDR PHY-Only Solution](#).

Figure 2.12. DDR PHY



### 2.4.3.1. I/O Lane [\(Ask a Question\)](#)

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 2.7.** DDR Memory Interface Data Lanes

Memory Type	DQ Width	Number of I/O Lanes	
		Address/Command Lanes	Data Lanes
DDR3/4	x16	3	2
DDR3/4	x32	3	4
DDR3/4	x40	3	5

**Table 2.7.** DDR Memory Interface Data Lanes (continued)

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

 **Important:** 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.

#### 2.4.3.2. Training Logic for Fabric DDR Controller [\(Ask a Question\)](#)

The training logic manages the DFI 4.0 training requests between the integrated PHY and DDR controller modules, and performs the following operations.

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

For more information, see [Training Logic](#).

For information about initialization sequence of DDR3, DDR4, and LPDDR3, see [Initialization Sequence](#).

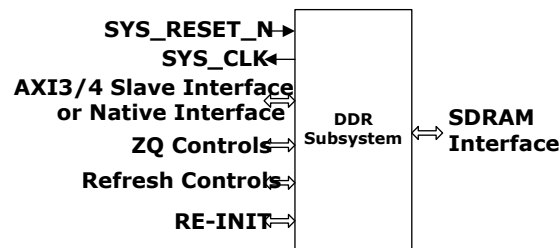
## 2.5. DDR Subsystem Ports [\(Ask a Question\)](#)

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 2.13. 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 2.8. 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 <sup>1</sup> . 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.
ALERT_N (ERR_OUT_N)	Input	This signal is asserted by the DDR memory when it detects a CA parity error or a CRC error. It is an input from DDR3 RDIMM or LRDIMM, and DDR4 components or DIMMs.
stat_ca_parity_error	Output	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 ALERT_N pin. The value is cleared on read.

**Note:**

- For more information about the PF\_INIT\_MONITOR IP, see [PolarFire Family 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. 512-bit AXI interface is supported only for DDR3, DDR3L, DDR4, and LPDDR3 memory devices.

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 2.9.** AXI3/4 Slave Interface Signals

Signal Name	Direction	Description
Write Address Channel		
AXIO_AWID[X-1:0] <sup>2</sup>	Input	Write address ID
AXIO_AWADDR[31:0]	Input	Write address
AXIO_AWLEN[M:0]	Input	Burst length. M = 7 for AXI4 IF and M = 3 for AXI3 IF.
AXIO_AWSIZE[2:0]	Input	Burst size.
AXIO_AWBURST[1:0]	Input	Burst type.
AXIO_AWLOCK[1:0]	Input	Lock type. Not supported.
AXIO_AWCACHE[3:0]	Input	Memory type. Not supported.
AXIO_AWPROT[2:0]	Input	Protection type. Not supported.
AXIO_AWVALID	Input	Write address valid.
AXIO_AWREADY	Output	Write address ready.
Write Data Channel		
AXIO_WID[X-1:0]	Input	Write ID tag. Supported only in AXI3.
AXIO_WDATA[N-1:0] <sup>1</sup>	Input	Write data.
AXIO_WSTRB[N/8 - 1:0] <sup>1</sup>	Input	Write strobes.
AXIO_WLAST	Input	Write last.
AXIO_WVALID	Input	Write valid.
AXIO_WREADY	Output	Write ready.
Write Response Channel		
AXIO_BID[X-1:0]	Output	Response ID tag.
AXIO_BRESP[1:0]	Output	Write response.
AXIO_BVALID	Output	Write response valid.
AXIO_BREADY	Input	Response ready.
Read Address Channel		
AXIO_ARID[X-1:0] <sup>2</sup>	Input	Read address ID.
AXIO_ARADDR[31:0]	Input	Read address.
AXIO_ARLEN[M:0]	Input	Burst length. M = 7 for AXI4 IF and M = 3 for AXI3 IF.
AXIO_ARSIZE[2:0]	Input	Burst size.
AXIO_ARBURST[1:0]	Input	Burst type.
AXIO_ARLOCK[1:0]	Input	Lock type. Not supported.
AXIO_ARCACHE[3:0]	Input	Memory type. Not supported.
AXIO_ARPROT[2:0]	Input	Protection type. Not supported.
AXIO_ARVALID	Input	Read address valid.
AXIO_ARREADY	Output	Read address ready.
Read Data Channel		
AXIO_RID[X-1:0] <sup>2</sup>	Output	Read ID tag.
AXIO_RDATA[N-1:0]	Output	Read data.
AXIO_RRESP[1:0]	Output	Read response.
AXIO_RLAST	Output	Read last.
AXIO_RVALID	Output	Read valid.
AXIO_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.

 **Important:** While performing DDR4 memory transfers using AXI interface, user must ensure not to exceed the 4 KB data limit as per the AXI specification. User can modify burst length (AxLEN) or burst size (AxSIZE) without exceeding the 4 KB limit.

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

**Table 2.10.** 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.
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 2.11.** 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 2.12.** ZQ Calibration Control Signals<sup>1</sup>

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.

**Note:**

1. The user-initiated ZQ calibration request and acknowledge control signals are supported for DDR3, LPDDR3, and DDR4 only.

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

**Table 2.13.** Reinitialization Control Signal<sup>1</sup>

Signal Name	Direction	Description
CTRLR_INIT	Input	Causes the subsystem to reissue the initialization sequence to the SDRAM. The initialization begins when this signal is asserted high for minimum four clock cycles in fabric clock domain. The controller always issues the initialization sequence (including the start-up delay) after a reset, regardless of this signal's state. If run-time reinitialization is not required, the signal can be tied low.

**Note:**

1. The user-initiated reinitialization control signal is supported for DDR3, LPDDR3, and DDR4 only.

The following table lists the ECC status signals.

**Table 2.14.** ECC Status Signals<sup>1</sup>

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.

**Note:**

1. The ECC status signals are supported for DDR3, LPDDR3, and DDR4 only.

The following table lists the SDRAM interface signals.

**Table 2.15.** 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.
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 (ERR_OUT_N)	Input	It is asserted by the DDR memory when it detects a CA parity error or a CRC error. It is an input from DDR3 RDIMM or LRDIMM, and DDR4 components or DIMMs.
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	DM/DM_N is an output mask signal for write data. <ul style="list-style-type: none"> <li>• For DDR3 and LPDDR3, output data is masked when DM is driven HIGH along with the write data during a write access.</li> <li>• For DDR4, output data is masked when DM_N is driven LOW along with the write data during a write access.</li> </ul>
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 the Ground. They are placed between the data lanes for improving signal integrity. <b>Note:</b> SHIELD signals are available only in the fabric DDR subsystem because it does not have dedicated I/O Banks to interface with the DDR memory. SHIELD pins are not applicable to all device and package combinations.

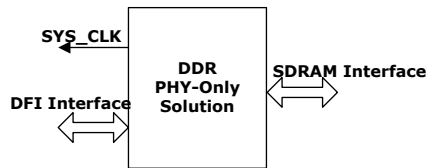
**2.5.1. DDR PHY-Only Solution Ports** [\(Ask a Question\)](#)

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 2.14. DDR PHY-Only Solution Ports



**Note:** The DDR PHY-Only solution is supported for DDR3, LPDDR3, and DDR4 only.

### 2.5.2. DFI Interface [\(Ask a Question\)](#)

The DDR memory controller interface solution leverages the DDR PHY interface (DFI 4.0) 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/LPDDR3/DDR4 PHY configurator to expose the DFI interface. For more information about the interface, see the DFI 4.0 Specification available on the [DFI Group DDR PHY website](#).

**Table 2.16.** DFI Control Signal Interface (synchronous to SYS\_CLK)

Signal Name	Direction	Description
DFI_ADDRESS_P0	Input	DFI Address bus. <ul style="list-style-type: none"> <li>For DDR3, the width is [15:0].</li> <li>For LPDDR3, the width is [19:0].</li> <li>For DDR4, the width is [16:0].</li> </ul>
DFI_ADDRESS_P1	Input	DFI Address bus. <ul style="list-style-type: none"> <li>For DDR3, the width is [15:0].</li> <li>For LPDDR3, the width is [19:0].</li> <li>For DDR4, the width is [16:0].</li> </ul>
DFI_ADDRESS_P2	Input	DFI Address bus. <ul style="list-style-type: none"> <li>For DDR3, the width is [15:0].</li> <li>For LPDDR3, the width is [19:0].</li> <li>For DDR4, the width is [16:0].</li> </ul>
DFI_ADDRESS_P3	Input	DFI Address bus. <ul style="list-style-type: none"> <li>For DDR3, the width is [15:0].</li> <li>For LPDDR3, the width is [19:0].</li> <li>For DDR4, the width is [16:0].</li> </ul>
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.

**Table 2.16.** DFI Control Signal Interface (synchronous to SYS\_CLK) (continued)

Signal Name	Direction	Description
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.
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 2.17.** 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] <sup>1</sup>	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] <sup>1</sup>	Input	DFI write data byte mask
DFI_WRDATA_MASK_P1[15:0] <sup>1</sup>	Input	DFI write data byte mask
DFI_WRDATA_MASK_P2[15:0] <sup>1</sup>	Input	DFI write data byte mask
DFI_WRDATA_MASK_P3[15:0] <sup>1</sup>	Input	DFI write data byte mask
DFI_WRDATA_P0[127:0] <sup>1</sup>	Input	These signals transfer write Data from memory controller to PHY
DFI_WRDATA_P1[127:0] <sup>1</sup>	Input	These signals transfer write Data from memory controller to PHY
DFI_WRDATA_P2[127:0] <sup>1</sup>	Input	These signals transfer write Data from memory controller to PHY
DFI_WRDATA_P3[127:0] <sup>1</sup>	Input	These signals transfer write Data from memory controller to PHY

**Note:**

1. Depends on DQ width (see [Table 2.23](#) and [Table 2.24](#)).

**Table 2.18.** 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] <sup>1</sup>	Input	DFI read data enable
DFI_RDDATA_EN_P1[63:0] <sup>1</sup>	Input	DFI read data enable
DFI_RDDATA_EN_P2[63:0] <sup>1</sup>	Input	DFI read data enable
DFI_RDDATA_EN_P3[63:0] <sup>1</sup>	Input	DFI read data enable
DFI_RDDATA_VALID_W0[7:0] <sup>1</sup>	Output	DFI read data valid
DFI_RDDATA_VALID_W1[7:0] <sup>1</sup>	Output	DFI read data valid
DFI_RDDATA_VALID_W2[7:0] <sup>1</sup>	Output	DFI read data valid
DFI_RDDATA_VALID_W3[7:0] <sup>1</sup>	Output	DFI read data valid
DFI_RDDATA_W0[127:0] <sup>1</sup>	Output	DFI read data
DFI_RDDATA_W1[127:0] <sup>1</sup>	Output	DFI read data
DFI_RDDATA_W2[127:0] <sup>1</sup>	Output	DFI read data
DFI_RDDATA_W3[127:0] <sup>1</sup>	Output	DFI read data

**Note:**

1. Depends on DQ width (see [Table 2.23](#) and [Table 2.24](#)).

**Table 2.19.** 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] <sup>1</sup>	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] <sup>1</sup>	Output	Write calibration data input to the controller Depends on DQ width.
CAL_L_DM_IN[63:0] <sup>1</sup>	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 <sup>2</sup>	Input	Calibration initialization bus available handshake from DDR controller.
CAL_INIT_CS[1:0] <sup>2</sup>	Output	Calibration initialization bus chip select.
CAL_INIT_MR_ADDR[7:0] <sup>2</sup>	Output	Calibration initialization bus mode register write address.
CAL_INIT_MR_DATA[17:0] <sup>2</sup>	Output	Calibration initialization bus mode register write data.
CAL_INIT_MR_MASK[17:0] <sup>2</sup>	Output	Calibration initialization bus mode register write mask.
CAL_INIT_MR_W_REQ <sup>2</sup>	Output	Calibration initialization bus mode register write request.

**Notes:**

1. Depends on DQ width (see [Table 2.23](#) and [Table 2.24](#)).
2. These ports are only for DDR4.
3. The DRAM interface is same as [Table 2.15](#) for DDR PHY only solution.

**Table 2.20.** 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 2.21.** DFI Status Interface

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

**Table 2.22.** 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

## 2.6. Functional Timing Diagrams [\(Ask a Question\)](#)

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

### 2.6.1. Native Interface Transactions [\(Ask a Question\)](#)

This section describes the read/write rules and sequences in the native interface. For information about native interface signals, see [Table 2.10](#).



**Important:** During the simulation, users can find the difference between Core configuration timings (JEDEC<sup>®</sup>) 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.

#### 2.6.1.1. SDRAM Writes [\(Ask a Question\)](#)

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:

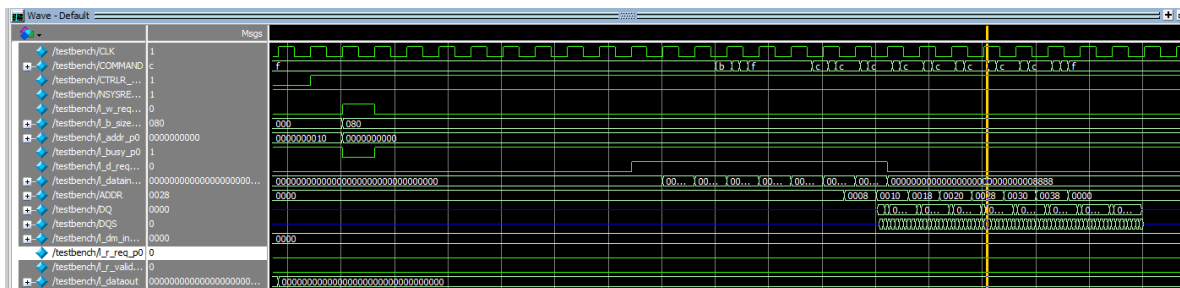
- A write is accepted by the subsystem on any clock cycle where the L\_W\_REQ signal is asserted while the L\_BUSY signal is deasserted. After L\_BUSY is deasserted by the controller, L\_W\_REQ may remain asserted to request a follow-on write transaction. L\_W\_REQ may remain asserted over any number of clock periods to generate any number of cascaded write requests.
- The values of the L\_ADDR\_L\_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.

#### 2.6.1.1.1. Sample Write Sequence [\(Ask a Question\)](#)

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

**Figure 2.15.** 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 128.
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.



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.

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.

### 2.6.2. AXI Transactions [\(Ask a Question\)](#)

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 2.17. AXI Single Write Transaction

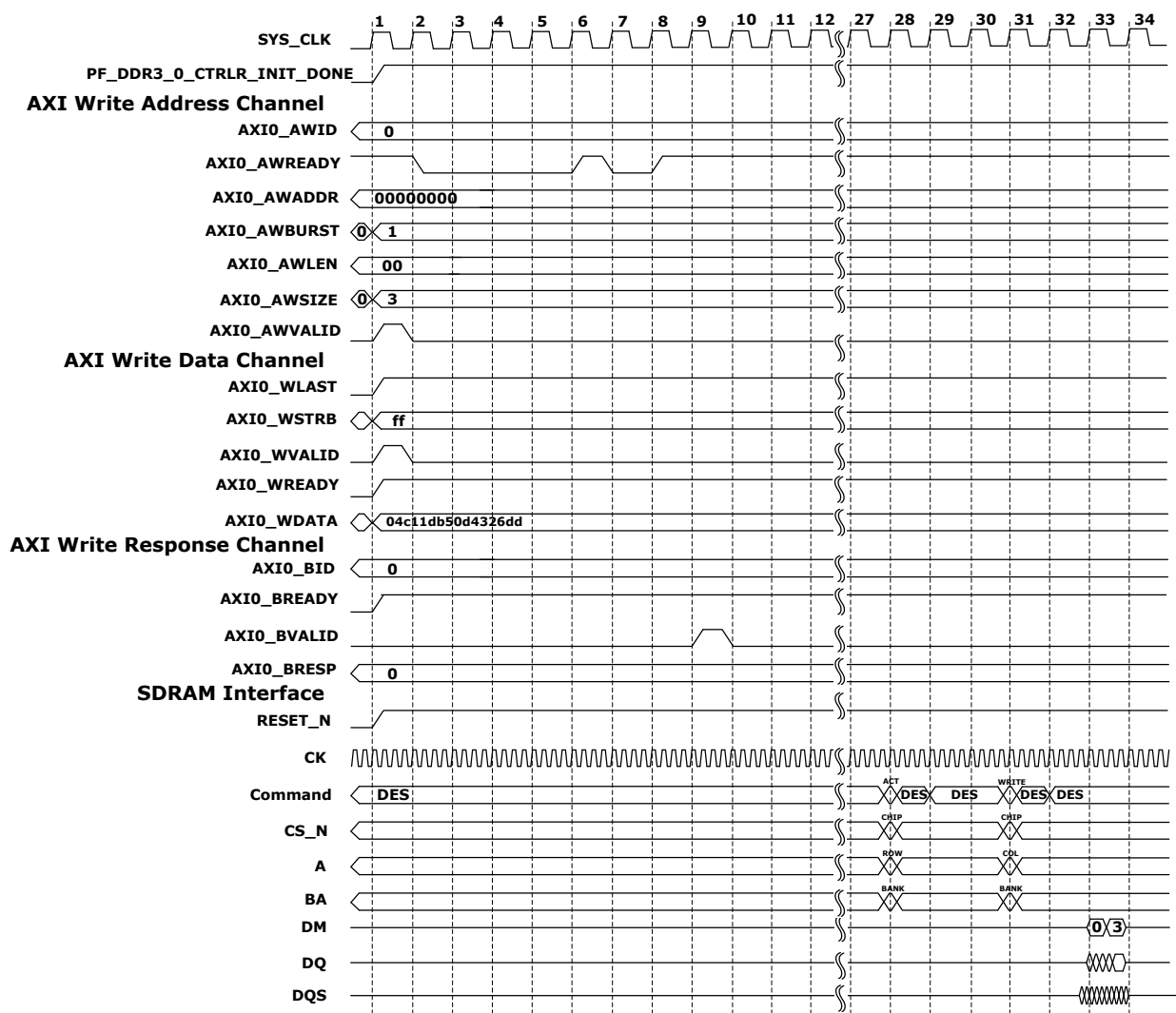


Figure 2.18. AXI Burst Write Transaction (INCR-2)

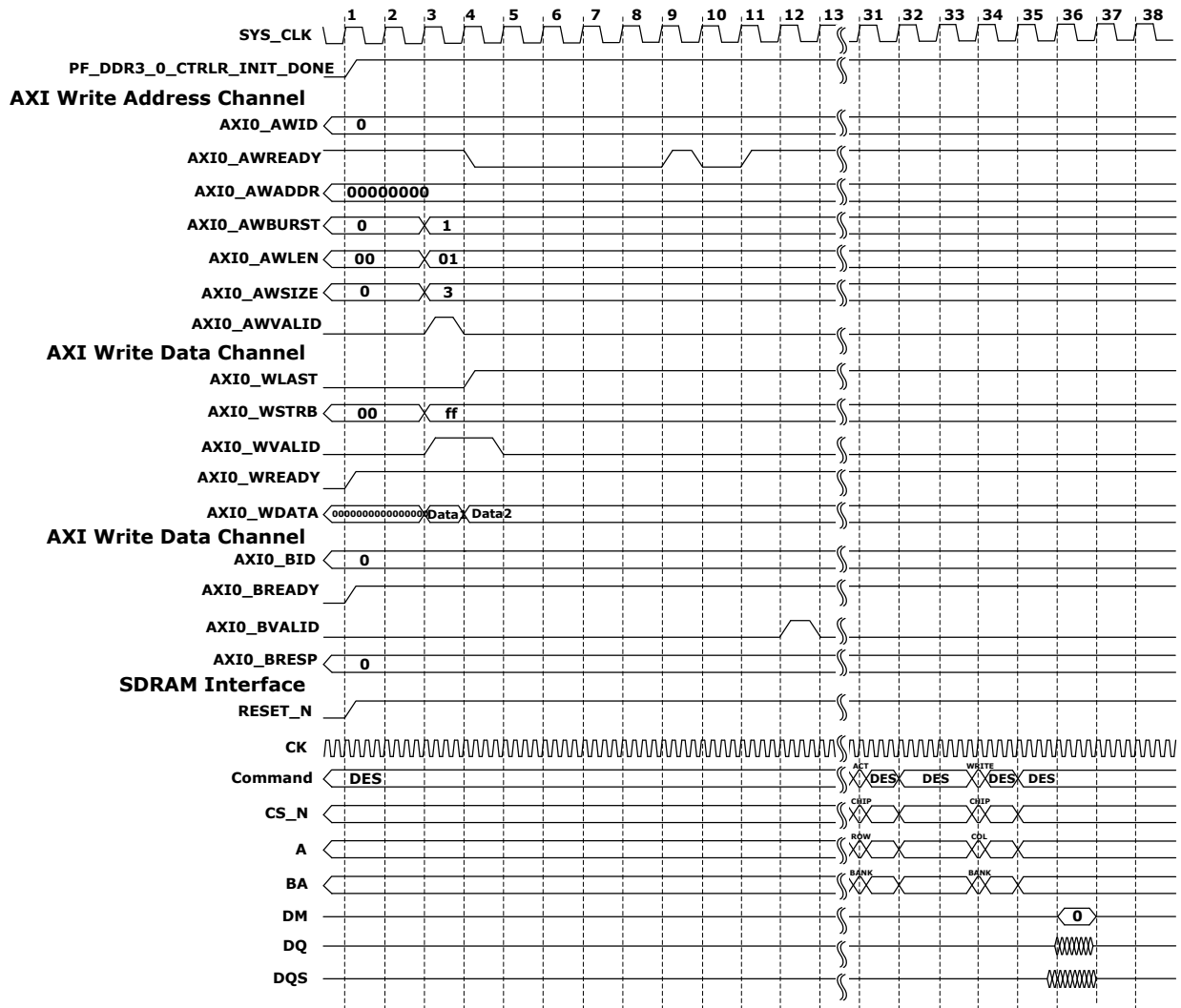


Figure 2.19. AXI Single Read Transaction

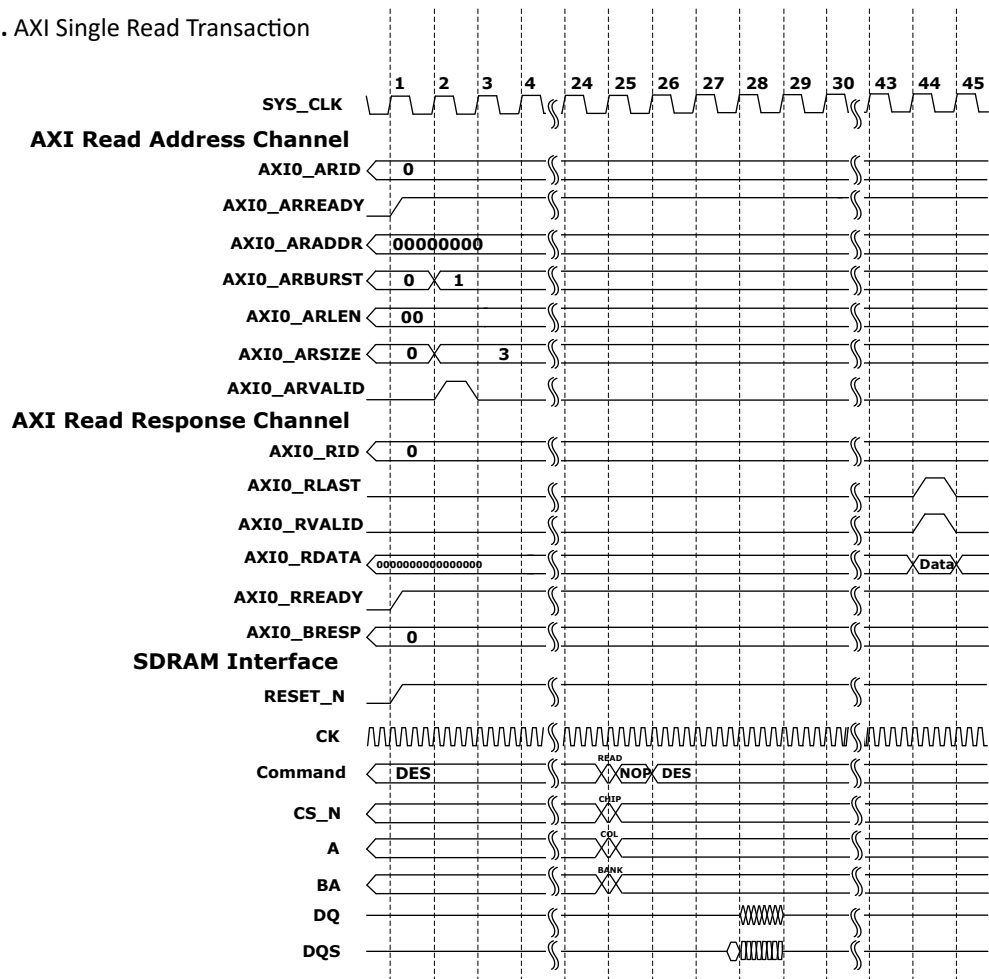
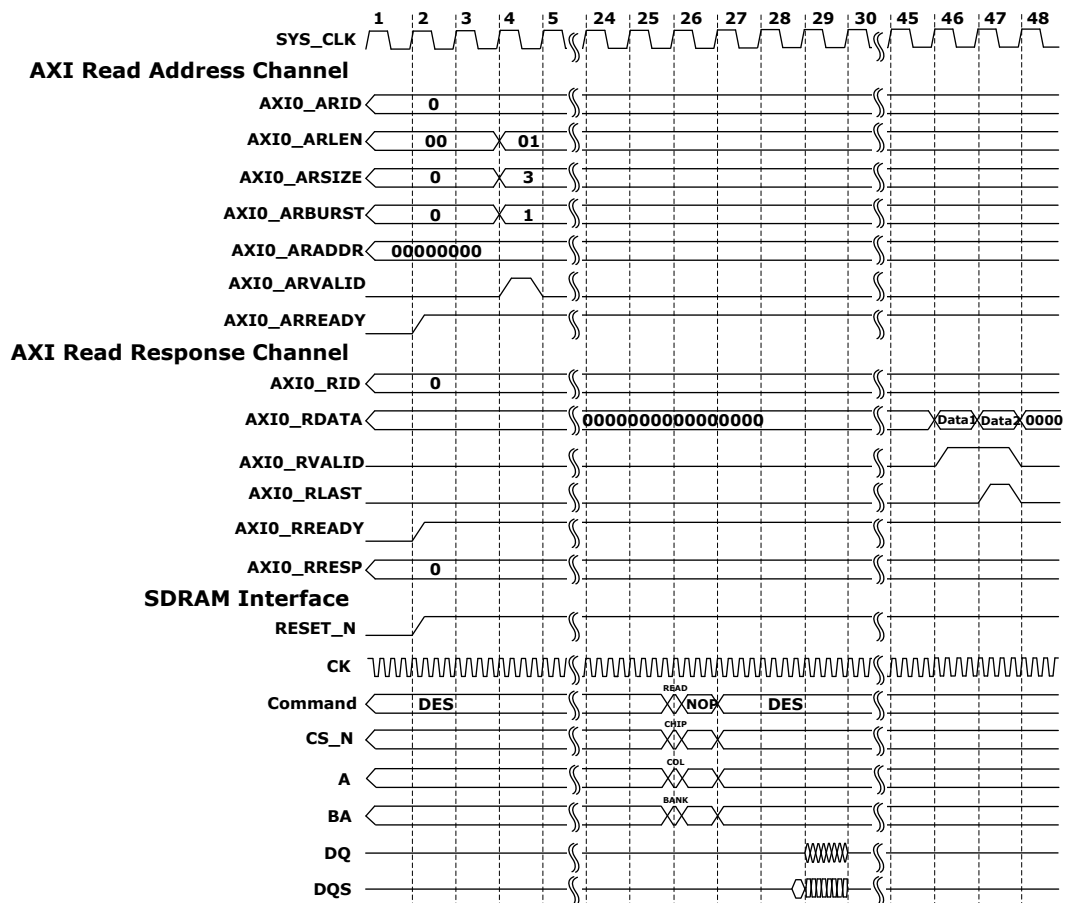


Figure 2.20. AXI Burst Read Transaction (INCR-2)



## 2.7. DDR PHY-Only Solution Integration [\(Ask a Question\)](#)

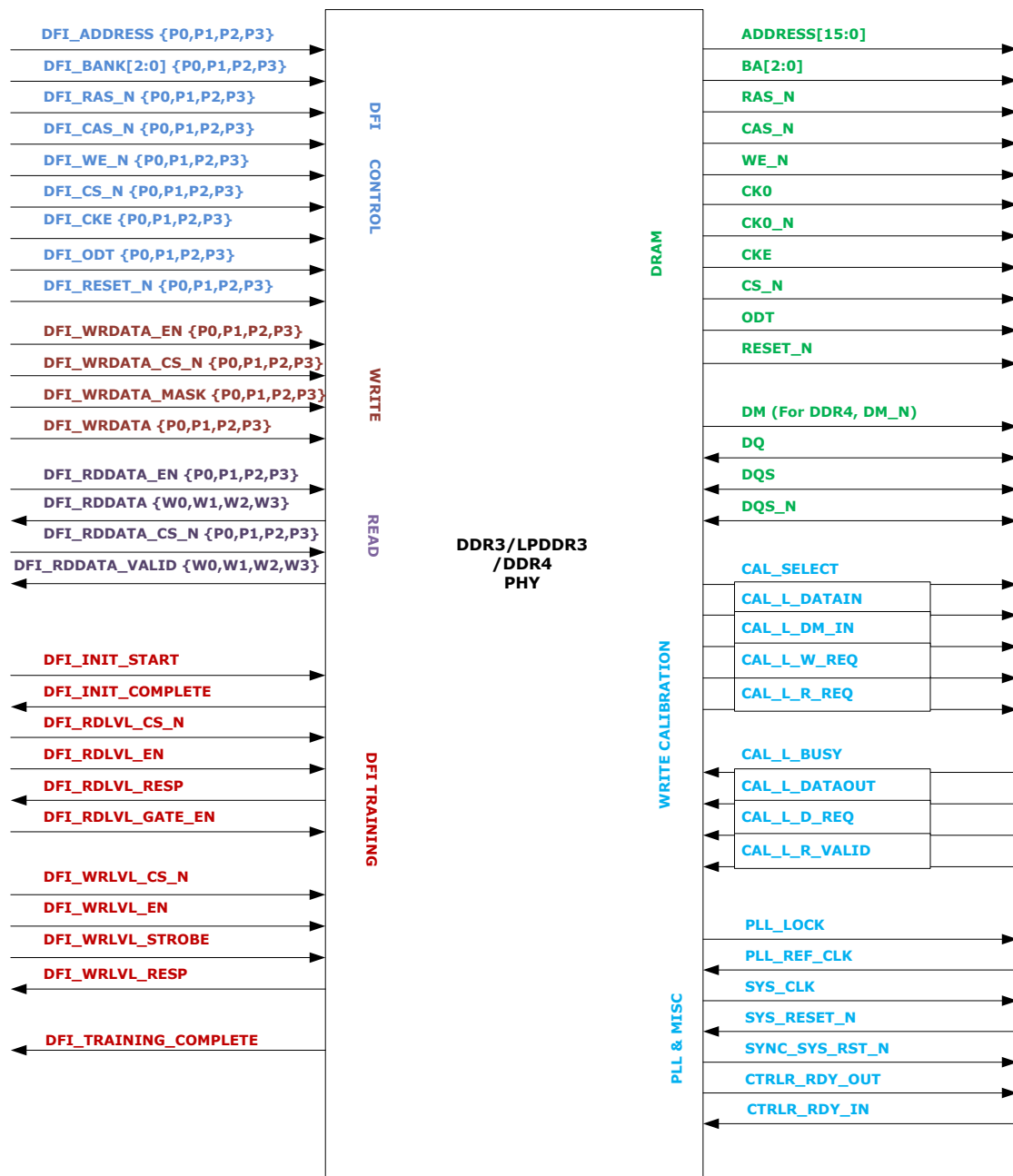
This section describes how to integrate third-party DDR memory controllers with PolarFire DDR PHY. The PHY top-level behavior on both DFI and DRAM interfaces are detailed in this section.

The PolarFire DDR PHY-only solution is DFI 4.0 compliant and supports DDR3, DDR3L, DDR4, and LPDDR3.

The PolarFire DDR PHY-only solution includes a PLL (CCC – Clock Conditioning Circuit) and an integrated PHY as shown in [Figure 2.2](#). 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 2.21. DDR PHY-Only Top-Level I/O



As shown in the preceding figure, DFI signals **DFI\_INIT\_START** and **DFI\_INIT\_COMPLETE** are part of the status interface according to the DFI 4.0 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.

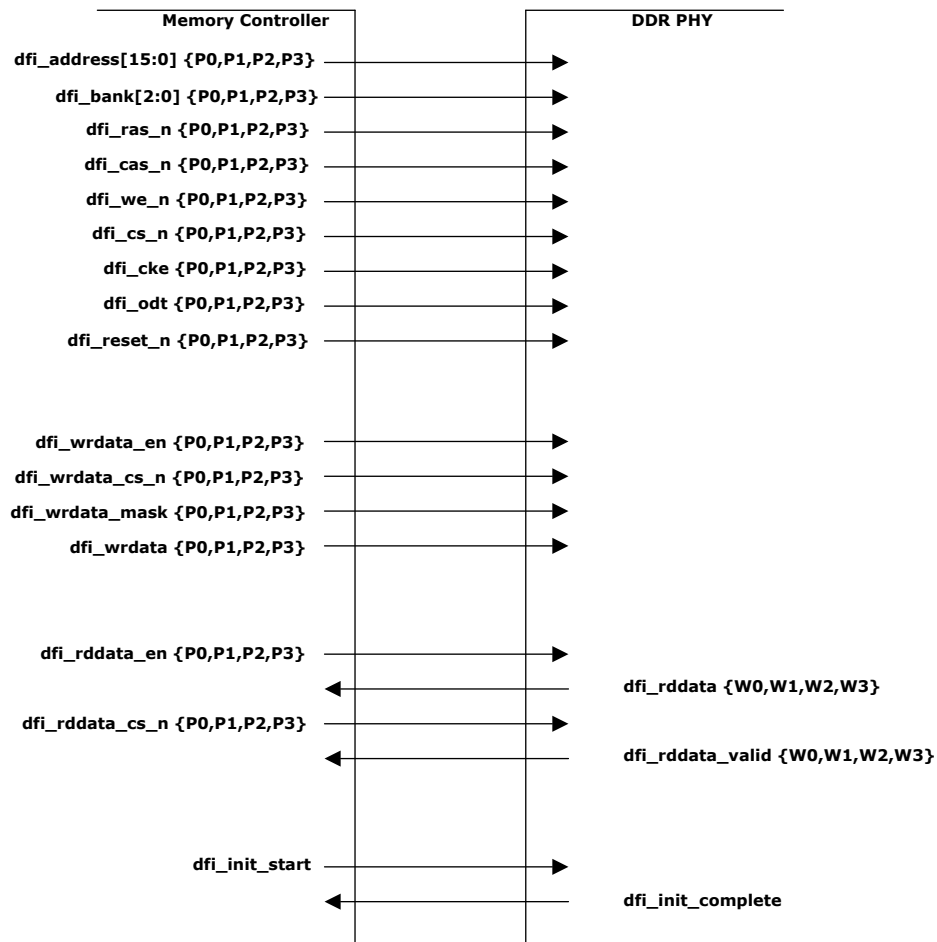
### 2.7.1. Controller and PHY Integration [\(Ask a Question\)](#)

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.

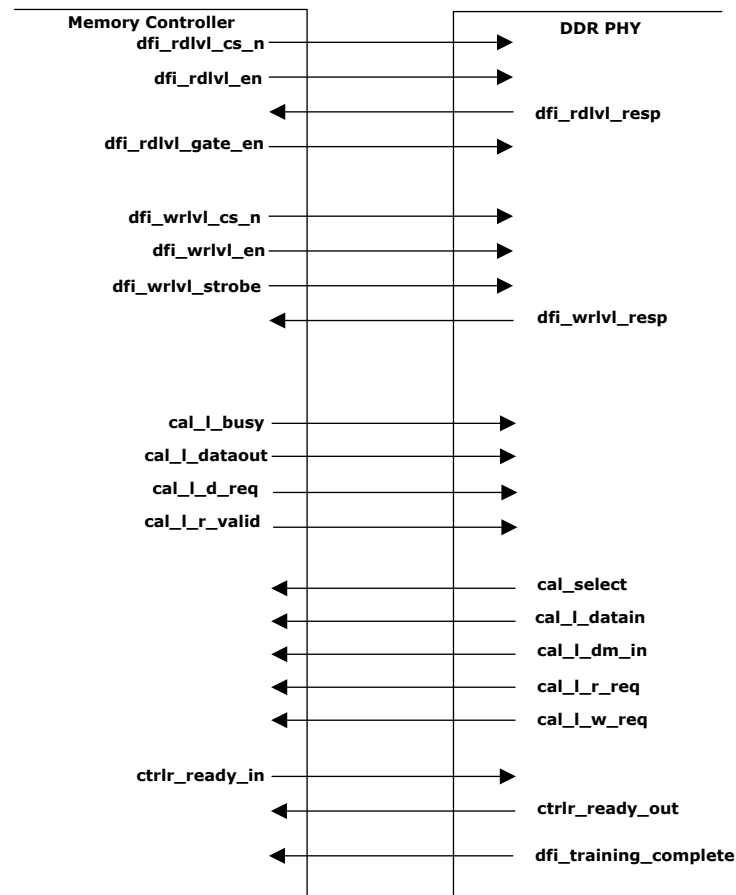
The following figure shows how PHY integrates with a memory controller for control, write and read interfaces.

**Figure 2.22.** DDR PHY DFI Interface



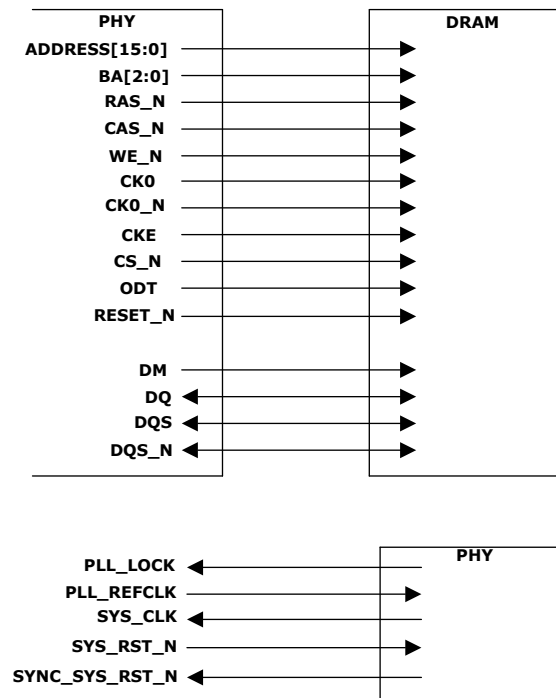
The following figure shows how the training IP inside the DDR PHY integrates with a memory controller. The training IP includes write calibration.

Figure 2.23. DDR PHY Training IP Interface with Write Calibration



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

Figure 2.24. DDR PHY Interface with DRAM, PLL, and Miscellaneous



### 2.7.2. PHY Configurator [\(Ask a Question\)](#)

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 2.23.** 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_w1[1:0]			
dfi_rddata_valid_w2[1:0]			
dfi_rddata_valid_w3[1:0]			
dfi_rddata_cs_n_p0**			
dfi_rddata_cs_n_p1**			
dfi_rddata_cs_n_p2**			
dfi_rddata_cs_n_p3**			
dfi_rddata_en_p0[1:0]*			
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]			
dfi_wrdata_mask_p1[3:0]			
dfi_wrdata_mask_p2[3:0]			
dfi_wrdata_mask_p3[3:0]			
dfi_wrdata_cs_n_p0**			
dfi_wrdata_cs_n_p1**			
dfi_wrdata_cs_n_p2**			
dfi_wrdata_cs_n_p3**			
dfi_wrdata_en_p0[1:0]*			
dfi_wrdata_en_p1[1:0]*			
dfi_wrdata_en_p2[1:0]*			
dfi_wrdata_en_p3[1:0]*			

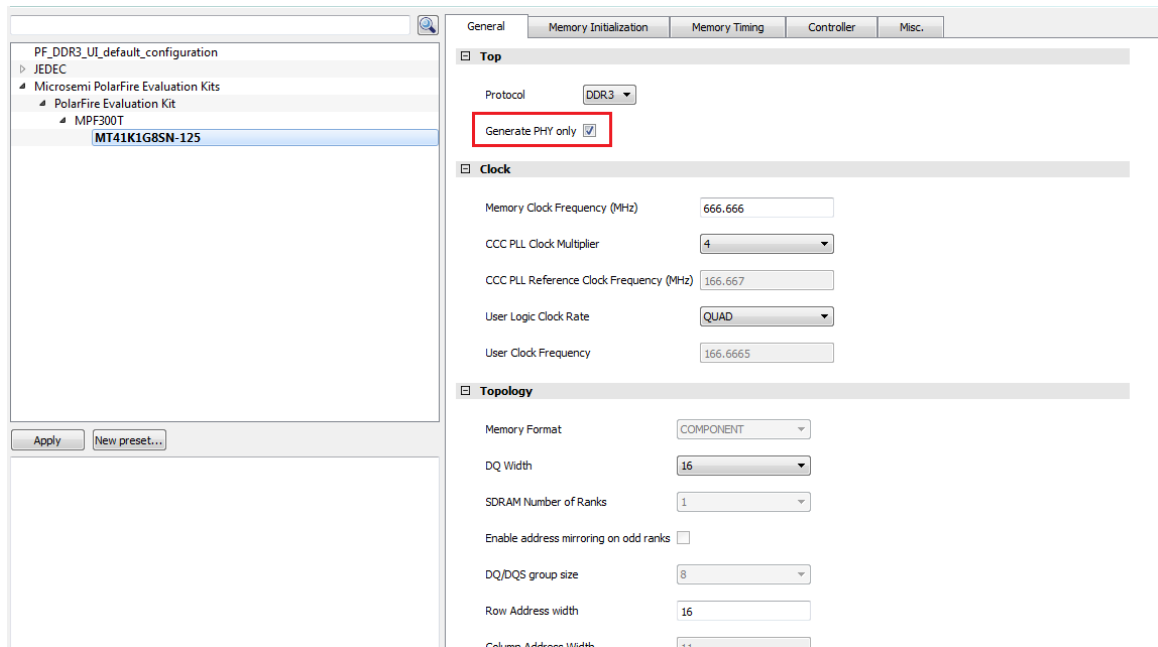
**Table 2.24. DFI Port Width Vs DRAM DQ Bus Width – 32-Bit Interface**

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**		WRITE	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 2.25. DDR3 PHY-Only Configurator



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

### 2.7.3. Initialization and Bring-Up of the DDR Subsystem [\(Ask a Question\)](#)

The DDR subsystem initializes in the following sequence:

1. DFI Initialization (DFI\_INIT\_START and 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



**Important:** Any DFI compliant controller must follow this sequence.

### 2.7.3.1. DFI Initialization [\(Ask a Question\)](#)

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<sub>REF</sub> 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  $\mu$ s, and then drive DFI\_RESET\_N\_P\* high. After that, the controller must wait for another 500  $\mu$ s and assert DFI\_CKE\_P\* high. This is required as per the JEDEC specification.

### 2.7.3.2. DRAM Initialization [\(Ask a Question\)](#)

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.

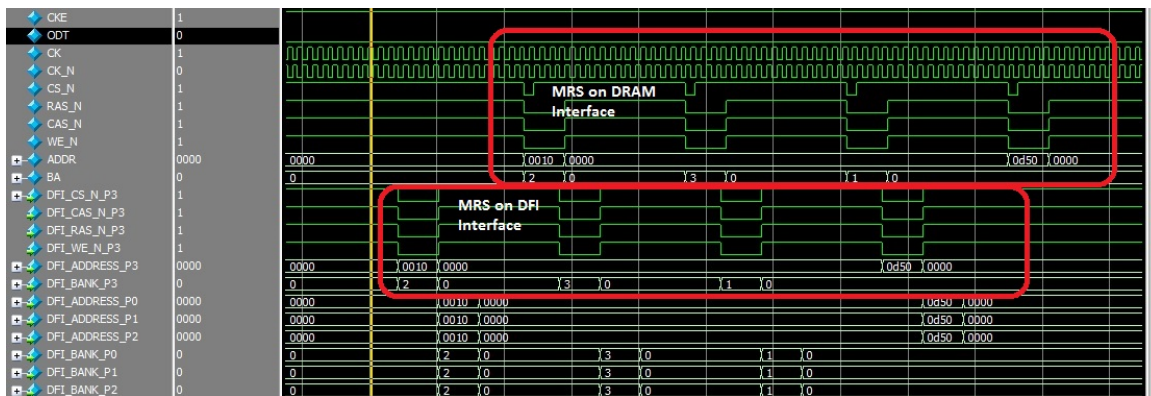
[Figure 2.26](#) shows the DFI to DRAM latency and is an example of how the memory controller drives the DFI interface to obtain the desired functionality and timing on the DRAM interface.

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 2.26](#) 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 2.26](#).

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

Figure 2.26. DFI To DRAM Control Interface Timing Relationship

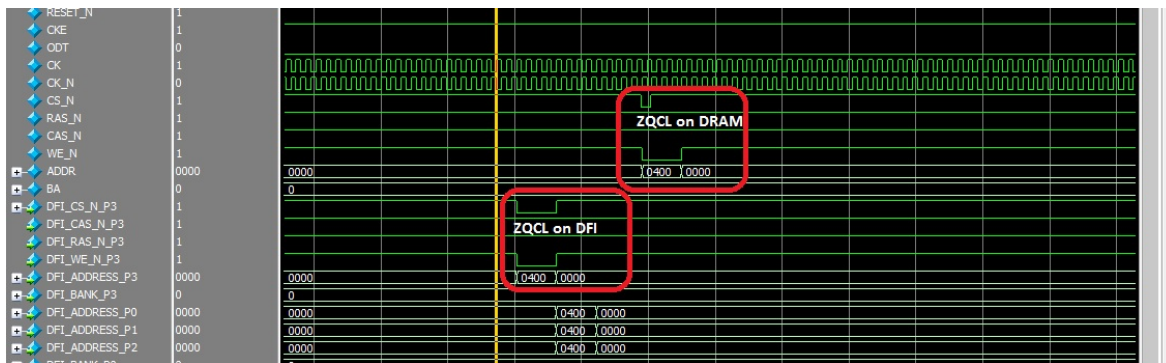


### 2.7.3.3. ZQ Calibration [\(Ask a Question\)](#)

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 2.27 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 2.27 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 2.27.

Figure 2.27. ZQCL On DFI Followed By ZQCL On DRAM INTERFACE



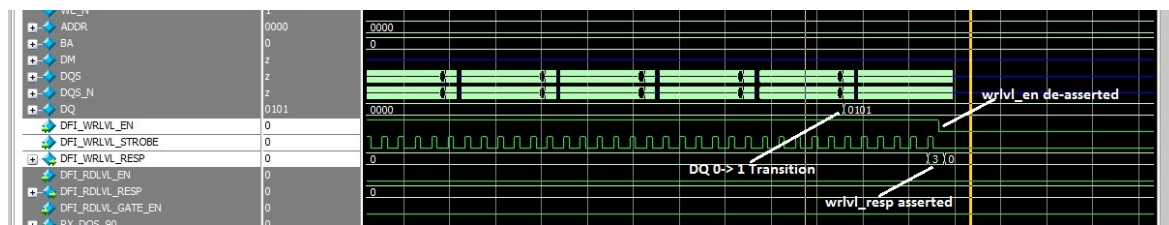
### 2.7.3.4. Write Leveling [\(Ask a Question\)](#)

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 2.28.** DFI Write Leveling Complete

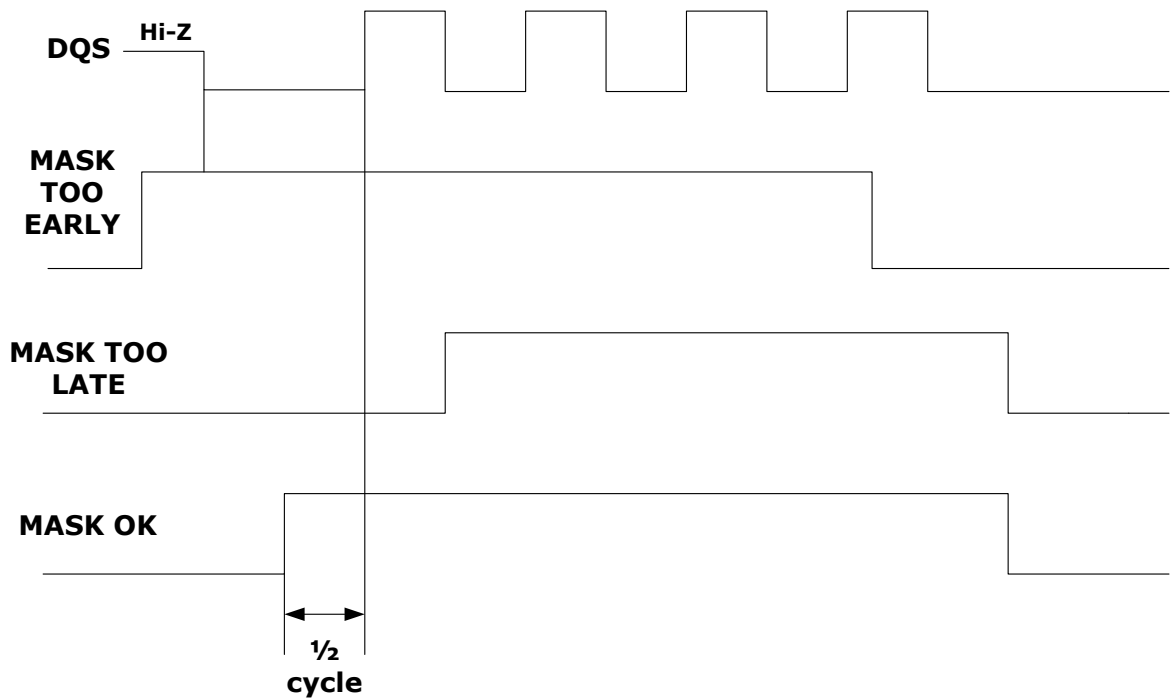


### 2.7.3.5. DQS Gate Training [\(Ask a Question\)](#)

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 2.29. 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.

### 2.7.3.6. Read Data Eye Training [\(Ask a Question\)](#)

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.

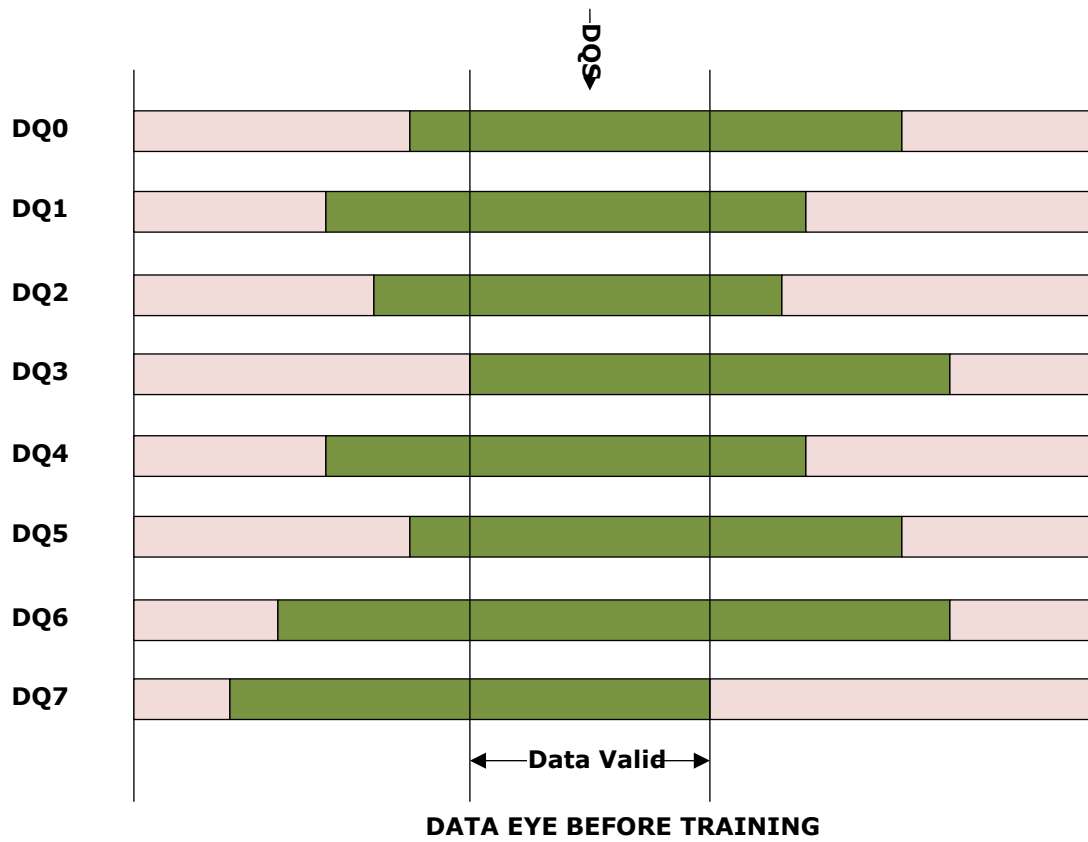


**Important:** For DDR3/4, the controller must issue the regular 'READ' commands to read data from the MPR register. For LPDDR3, the controller must issue mode register read to MR32 or MR40.

---

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

Figure 2.30. Read Data Eye Before And After Training



### 2.7.3.7. Write Calibration [\(Ask a Question\)](#)

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 family 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' and '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 2.31.** Write And Read Of Known Patterns During Write Calibration

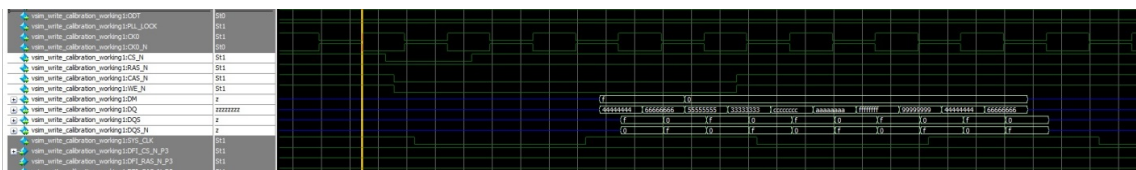


Figure 2.32. Write And Read Sequence During Write Calibration

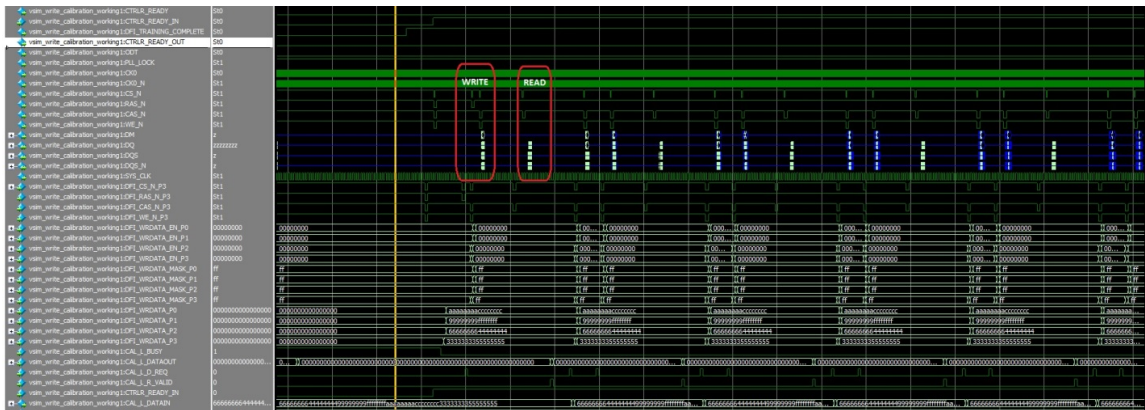


Figure 2.33. Write Calibration Sequence

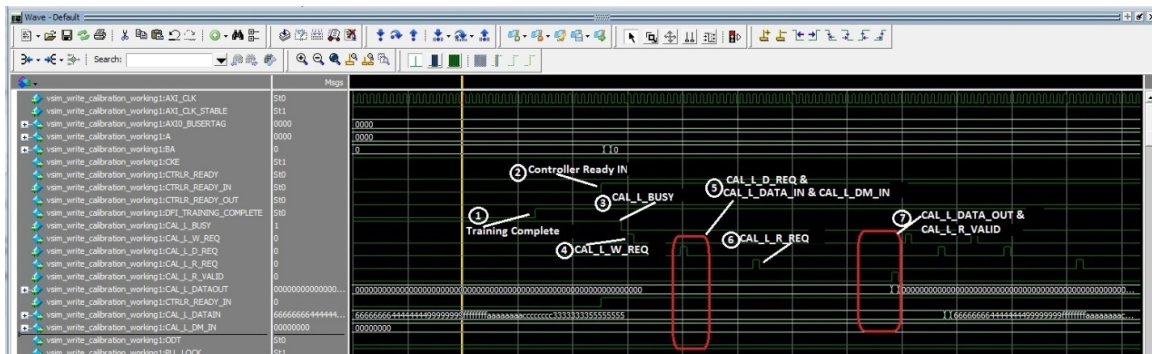
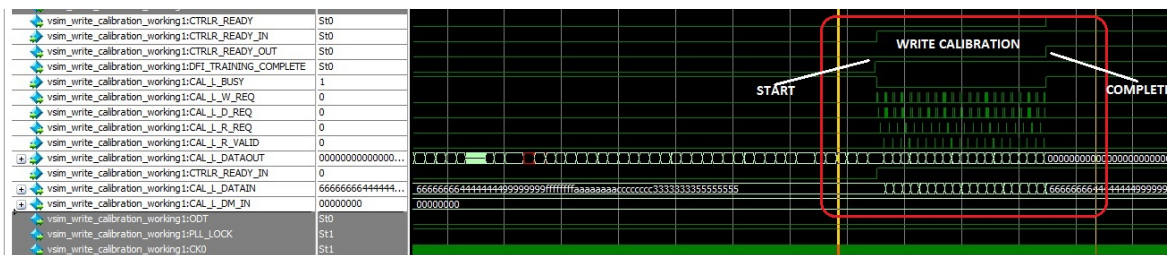


Figure 2.34. Write Calibration Complete



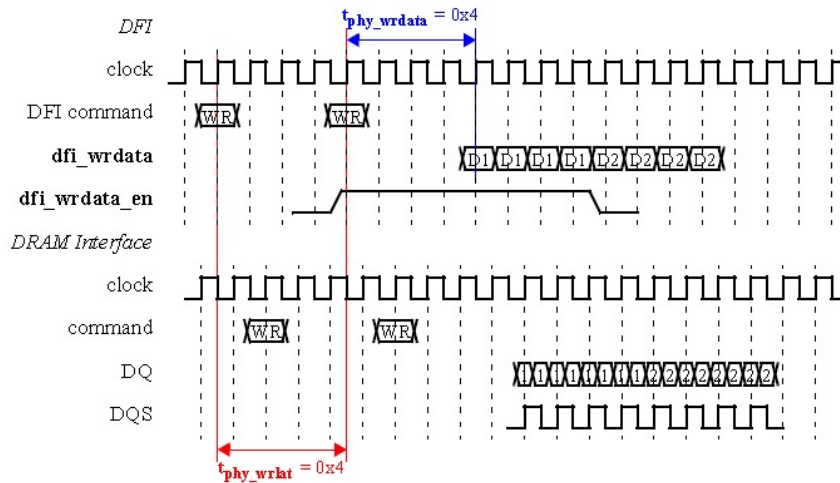
### 2.7.3.8. Simple Burst Write (Ask a Question)

After Write Calibration, the DDR subsystem is ready for a normal operation. Typically, the controller asserts a signal like CTRL\_R\_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 2.35 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 2.35.** 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 and 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 and WRDATA\_MASK are issued in the next phase after the issuing WRDATA\_EN.

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 and WRDATA\_MASK are issued in the next phase after issuance of WRDATA\_EN.

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

**Figure 2.36.** Activate And Write Commands On DFI and DRAM (16-Bit Interface)

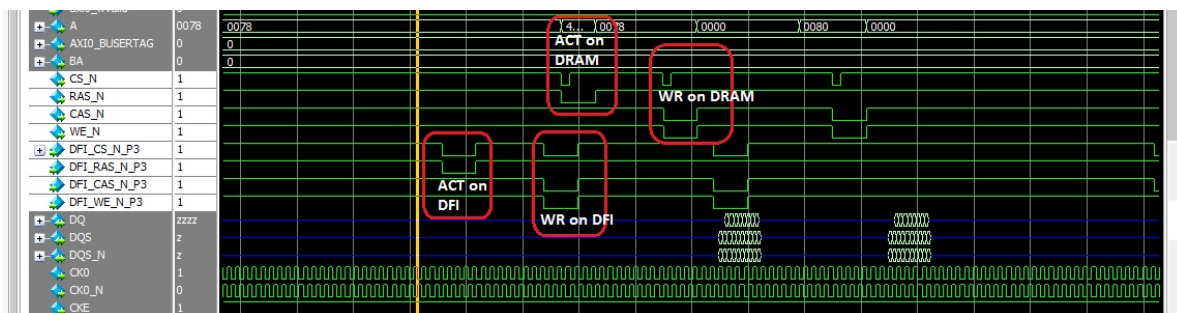


Figure 2.37. WRDATA On DFI and DRAM - DFI 128 Bits (4 X 32) Translate To 16-Bit BL8 On DRAM

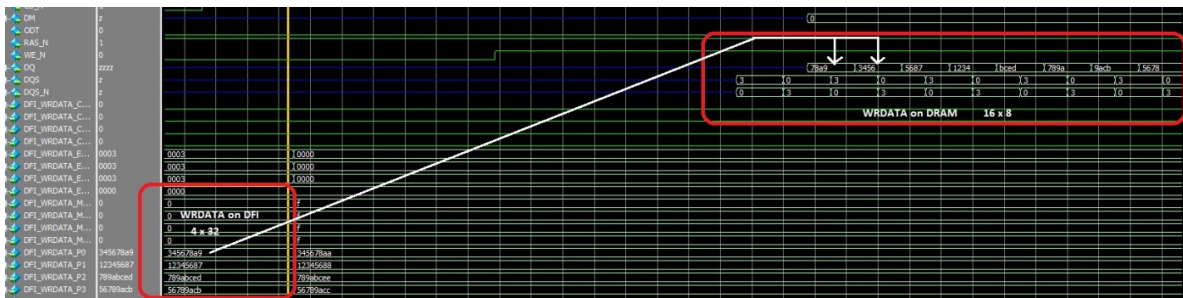
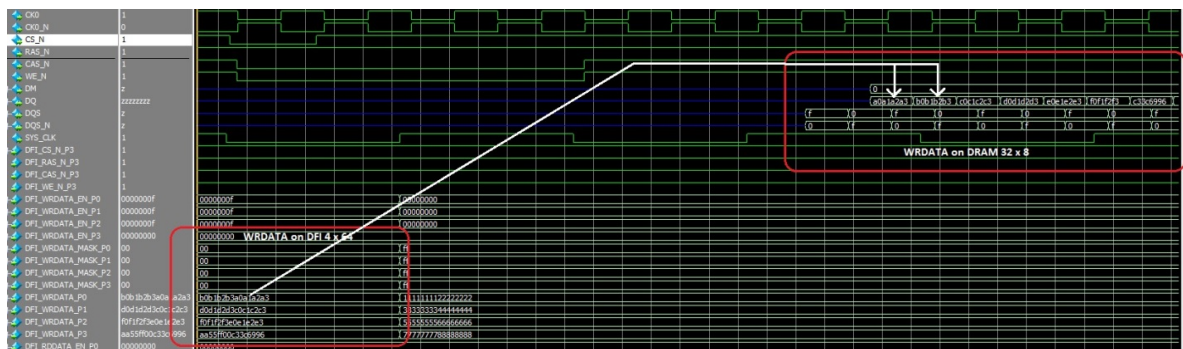


Figure 2.38. WRDATA On DFI and DRAM - DFI 256 BITS (4 X 64) Translate To 32-Bit BL8 On DRAM



### 2.7.3.9. Simple Burst Read (Ask a Question)

This step describes simple burst read operation. Similar to WRITE, there are two Timing parameters the controller must be aware of:  $t_{rddata\_en}$  and  $t_{phy\_rdlat}$ . After issuing DFI READ command on control interface, Controller must assert  $DFI\_RDDATA\_EN$  after  $t_{rddata\_en}$  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 2.39. DFI Read Burst Example

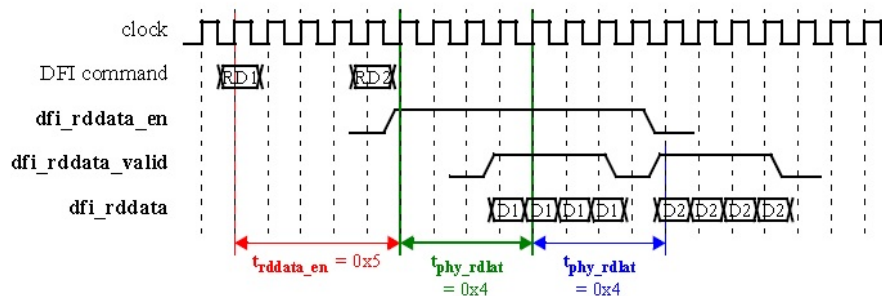


Figure 2.40. RDDDATA On DFI - 16-Bit BL8 On DRAM Translate To DFI 128 BITS (4 X 32)

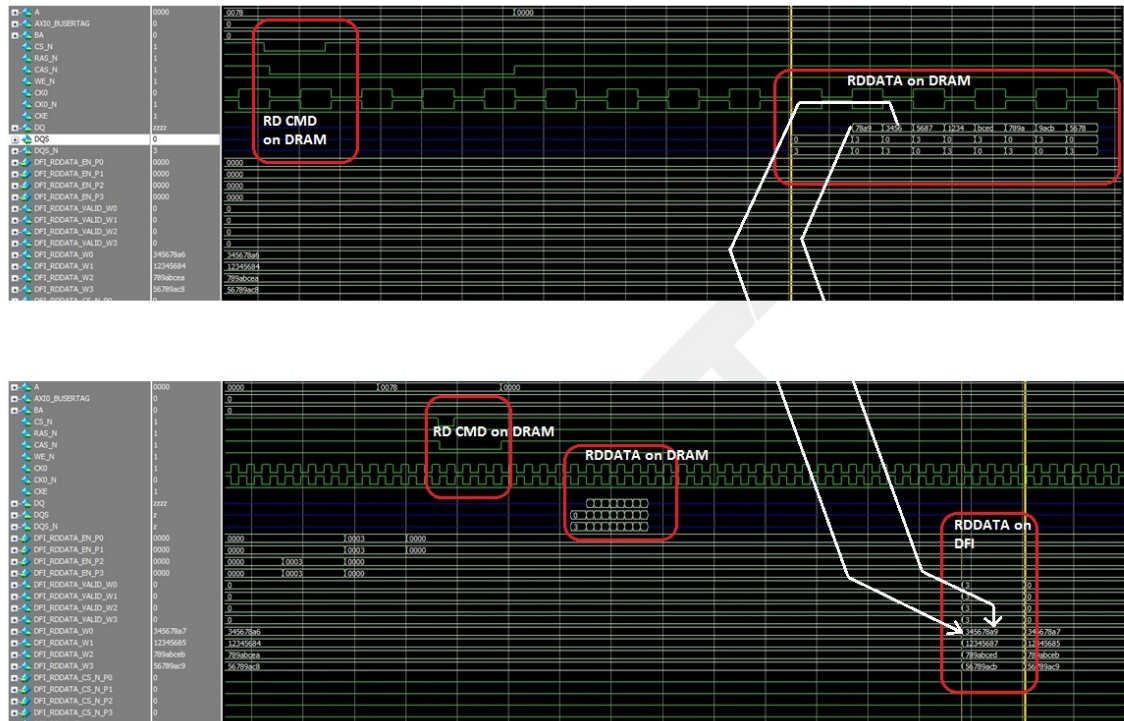
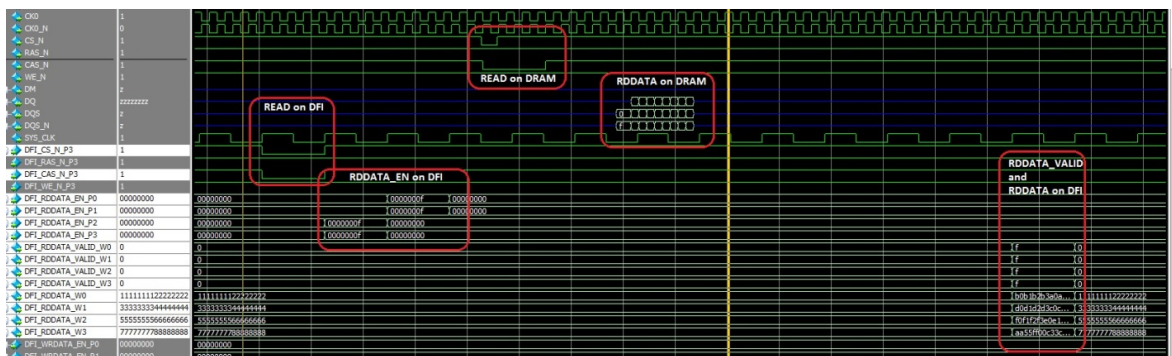


Figure 2.41. RDDDATA On DFI - 32-Bit BL8 On DRAM Translate To DFI 256 BITS (4 X 64)



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

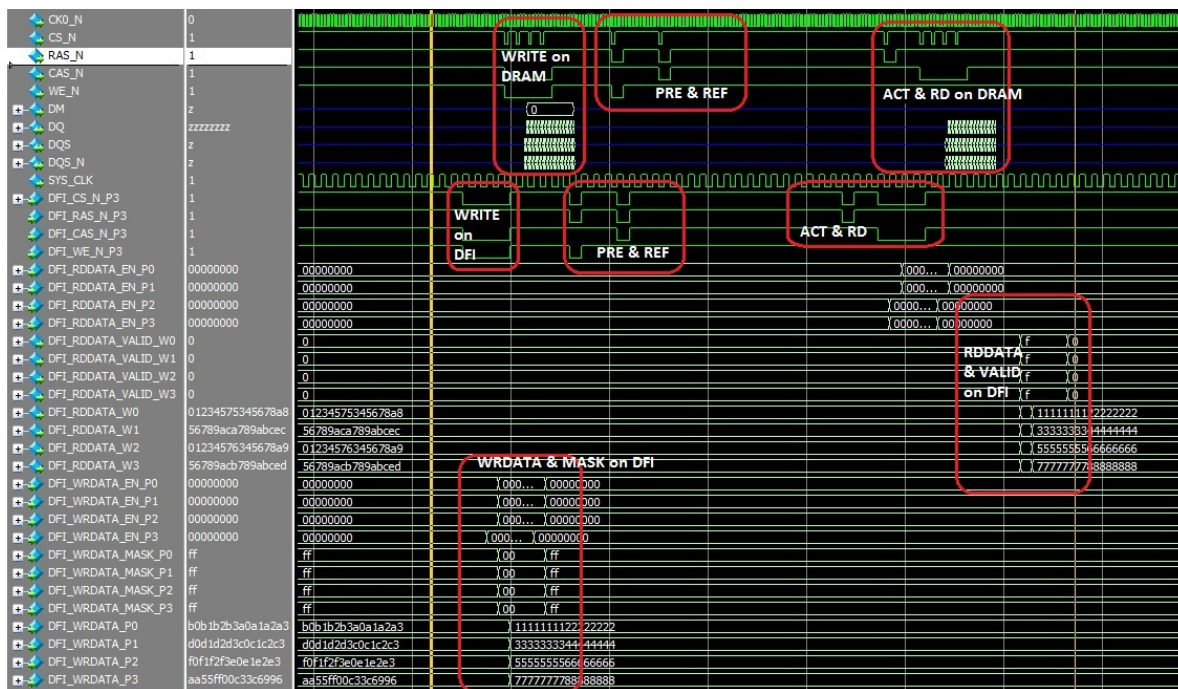
any time prior to max delay. DFI\_RDDATA\_VALID is trained and DFI\_RDDATA is valid only when DFI\_RDDATA\_VALID is asserted.

#### 2.7.4. Timing [\(Ask a Question\)](#)

The following figure shows a typical DDR3 protocol sequence and behavior of DFI and DRAM interfaces. All 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 includes commands like PRECHARGE and REFRESH.

**Figure 2.42.** Continuous Multi-Burst Write And Read On DFI/DRAM



#### 2.8. Octal DDR PHY-Only Solution [\(Ask a Question\)](#)

Libero SoC Catalog includes the PolarFire Octal DDR PHY IP, that 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 the xSPI (JESD251), HyperBUS, and ONFI memories. 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 250 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 four 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.

### 2.8.1. Configurator [\(Ask a Question\)](#)

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

**Figure 2.43.** Octal DDR PHY Configurator Main GUI

The screenshot shows the Octal DDR PHY Configurator GUI with the following settings:

- Configuration Tab:**
  - Protocol: xSPI
  - Memory clock frequency: 200 MHz
  - Clock type: Single ended
  - Enable Differential DQS:
- Topology Section:**
  - Number of interface: 1
  - x number of rank per interface: 1
  - Number of DQ: 8
- Clocking Section:**
  - 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 Section:**
  - DQS/RWDS delay tap adjustment: 70
  - Expose DQ raw data:

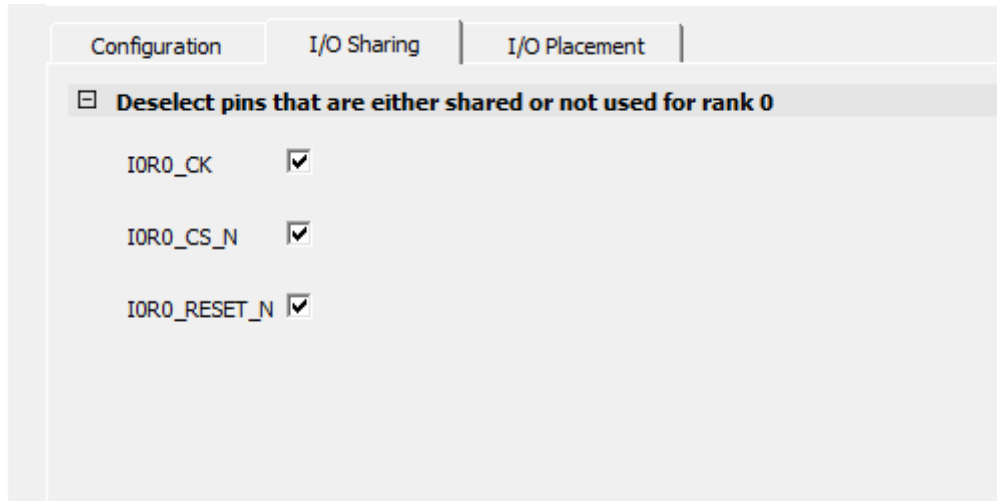
The following table lists the Octal DDR PHY Configurator options and parameters.

**Table 2.25.** Configurator Options and Parameters

Parameter	Range	Description
PROTOCOL	XSPI	Protocol used to build the PHY
	ONFI	
	HYPERBUS	
	APMEMORY_PSRAM	
CLOCK_FREQ	1-250 MHz	Memory clock frequency
CLOCK_TYPE	SINGLE_ENDED	1 clock
	DIFFERENTIAL	True differential clock pair
	PSEUDO_DIFFERENTIAL	Two single ended clocks in phase opposition
Enable Differential DQS	—	This option is available for ONFI, HyperBus, and APMemory PSRAM protocols. When this option is selected, differential DQS bidirectional ports are exposed for DQS IOD. This option is required for interfacing to an external flash or NVDDR RAM at more than 200 MHz.
NUMBER_OF_DQ	4 or 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	1-39	Multiplication factor from the external reference clock to the memory frequency.
CCC_PLL_REF_FREQ	—	Resulting reference clock frequency
DQS_DELAY_TAP	1 - 255	Delay tap set on DQS for read. Libero default value of 70 is optimized for 200 MHz performance if PCB routing is closely matched. Value below 20 fails in simulation.
INTERFACE_NB	1-8	Number of shared interface
RANK_NB	1-4	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 HyperRAM), or for multi-rank deselect ports that are shared between ranks.

Figure 2.44. I/O Sharing Configurator Tab



The  $I\{\text{INTERFACE}\}R\{\text{RANK}\}_{\text{IO}}\_IN\_DQ\_LANE$  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 2.45. I/O Placement Configurator Tab

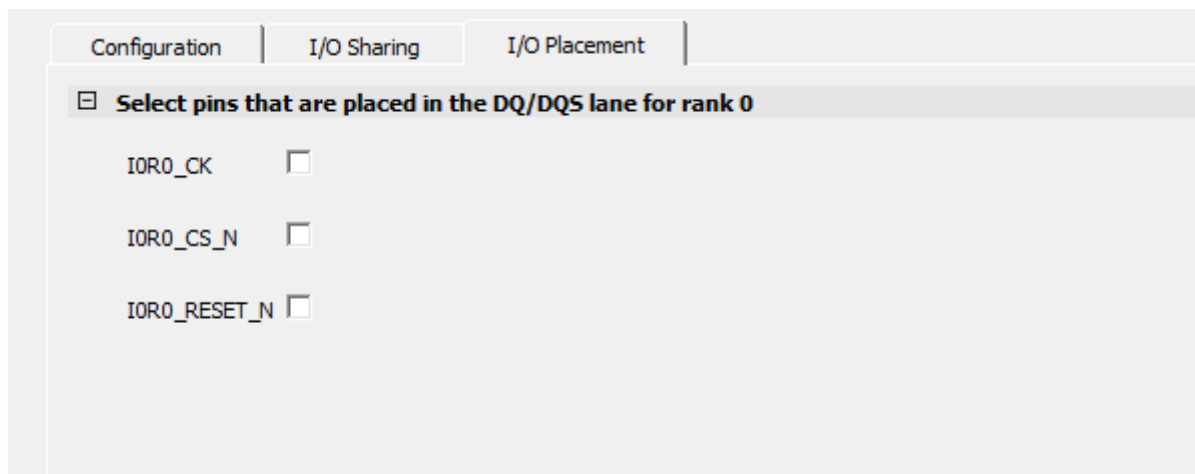


Figure 2.46. Generated Octal DDR PHY SmartDesign Component

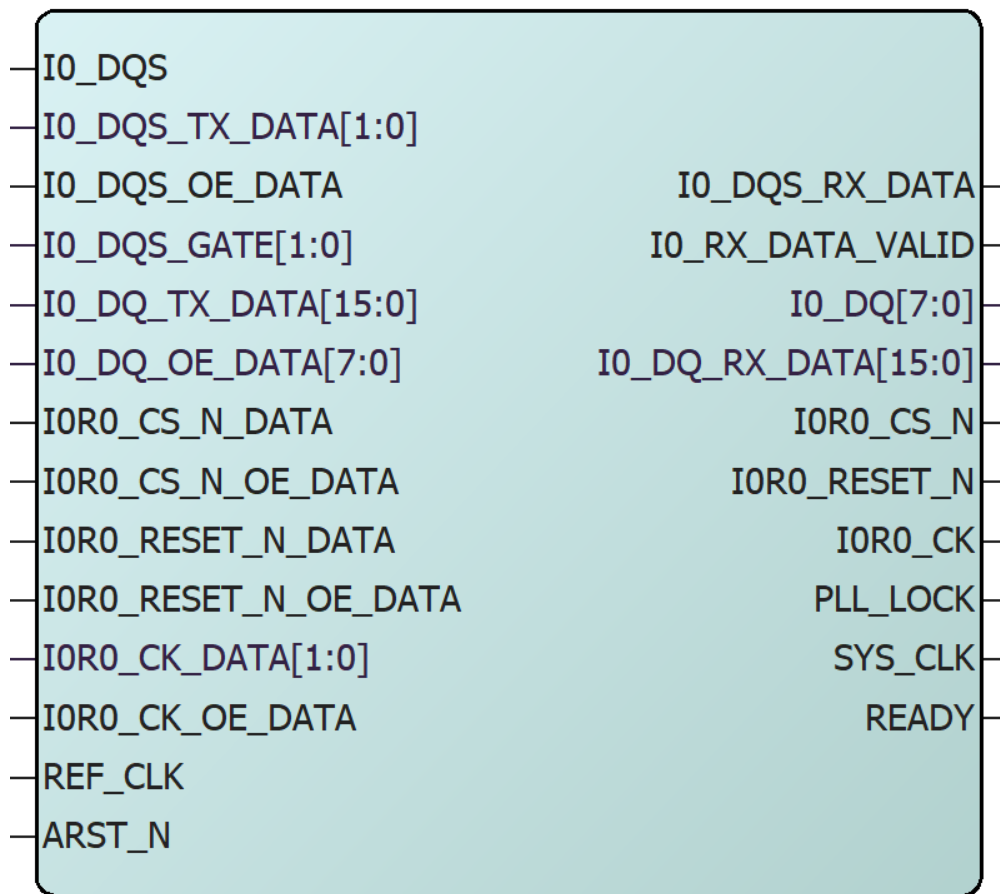


Table 2.26. Port List

Port	Width	Direction	Visibility	Description	Notes
<b>General</b>					
REF_CLK	1	IN	—	Reference clock to the CCC. It must be at CCC_PLL_REF_FREQ frequency.	—
ARST_N <sup>1</sup>	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.	—

Table 2.26. Port List (continued)

Port	Width	Direction	Visibility	Description	Notes
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.	—
<b>Output</b>				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 I/O Sharing tab in the Configurator.	Output pad N side to the memory, DDR.	CCC OUT1 through HS_IO_CLK, 90deg phase shift
I{INTERFACE#}R{RANK#}_{CLOCK}_DATA	2	IN	I{INTERFACE#}R{RANK#}_{CLOCK}_ENABLED== 1. This parameter is set using the related I/O Sharing tab in the Configurator.	2-bit port. <ul style="list-style-type: none"> <li>• 00: Drives 0 out</li> <li>• 01: Drives clock pattern out</li> <li>• 10: Drives inverted clock pattern out</li> <li>• 11: Drives 1 out</li> </ul>	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 I/O 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
<b>Command</b>				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

Table 2.26. Port List (continued)

Port	Width	Direction	Visibility	Description	Notes
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 I/O 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
<b>Asynchronous Inputs</b>					
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
<b>DQS</b>					
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

Table 2.26. Port List (continued)

Port	Width	Direction	Visibility	Description	Notes
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
<b>DQ</b>					
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	1	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

**Note:**

1. The ARST\_N signal is not internally synchronized to the REF\_CLK signal in the PF\_IOD\_OCTAL\_DDR IP. Thus, as good design practice, synchronize the ARST\_N signal in the fabric, and then, define a false path constraint for any paths traversing through the reset synchronizer.

Figure 2.47. Command Timing

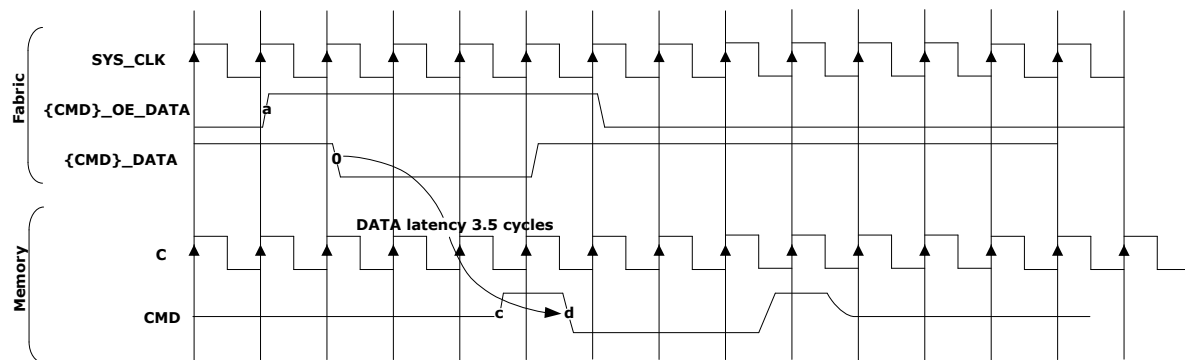
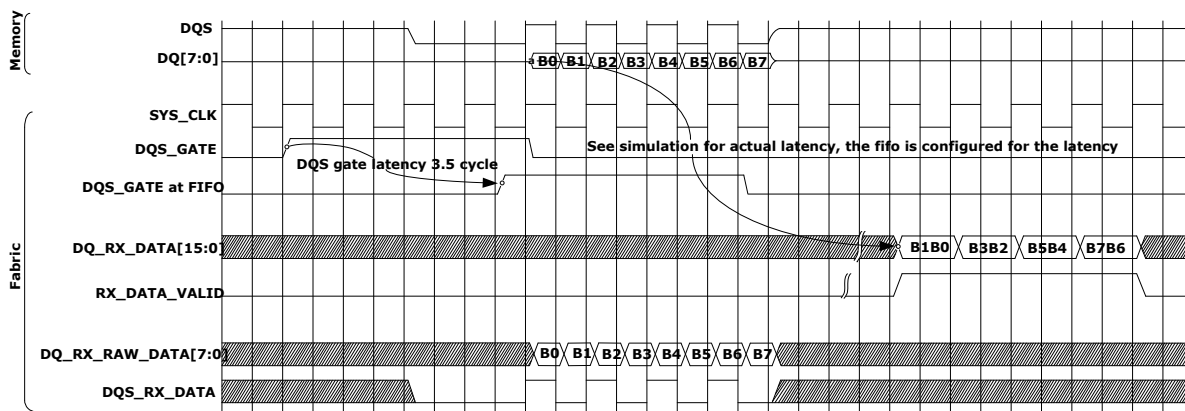


Figure 2.48. DQ/DQS Receive



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

### 2.8.2. Timing Constraints [\(Ask a Question\)](#)

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**.

### 2.8.3. Place and Route Pin-Out Rules [\(Ask a Question\)](#)

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.



**Important:**

In the Octal PHY solution the number of DQS must be a multiple of 2. If you do not follow this rule, then RX\_DATA\_VALID will stay stuck at 1. Another way to formulate it is that the burst length of transaction issue to Octal PHY must be a multiple of 2.

## 2.9. Implementation [\(Ask a Question\)](#)

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.

### 2.9.1. Selecting the Memory Device [\(Ask a Question\)](#)

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, DDR3L, DDR4, and LPDDR3 memory devices. Based on the system requirements (bandwidth, speed, bus width, density, and cost), the appropriate memory device is selected.

**Table 2.27.** 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.

#### 2.9.1.1. DDR4 Optimization Guidelines [\(Ask a Question\)](#)

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  $\geq 5$  ns.
    - $4 \text{ clock cycles} * (1/800 \text{ MHz}) = 5 \text{ ns}$ , hence, an 800 MHz memory clock can be used to achieve 1600 Mbps data rate.
  - For DDR4-2133 memory components, tCCD\_L must be  $\geq 5.355$  ns
    - $4 \text{ clock cycles} * (1/746 \text{ MHz}) = 5.36 \text{ ns}$ , hence, a 746 MHz memory clock can be used to achieve 1492 Mbps data rate.
  - For DDR4-1866 memory components, tCCD\_L must be  $\geq 5.355$  ns
    - $4 \text{ clock cycles} * (1/746 \text{ MHz}) = 5.36 \text{ ns}$ , 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 preceding 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.

### 2.9.2. Configuring the DDR3 Subsystem [\(Ask a Question\)](#)

The PF\_DDR3 macro located in the Libero IP catalog must be instantiated in SmartDesign to access the DDR3 memory from the FPGA fabric. The DDR3 configurator, shown in [Figure 2.51](#), configures the DDR3 subsystem. It supports the following modes:

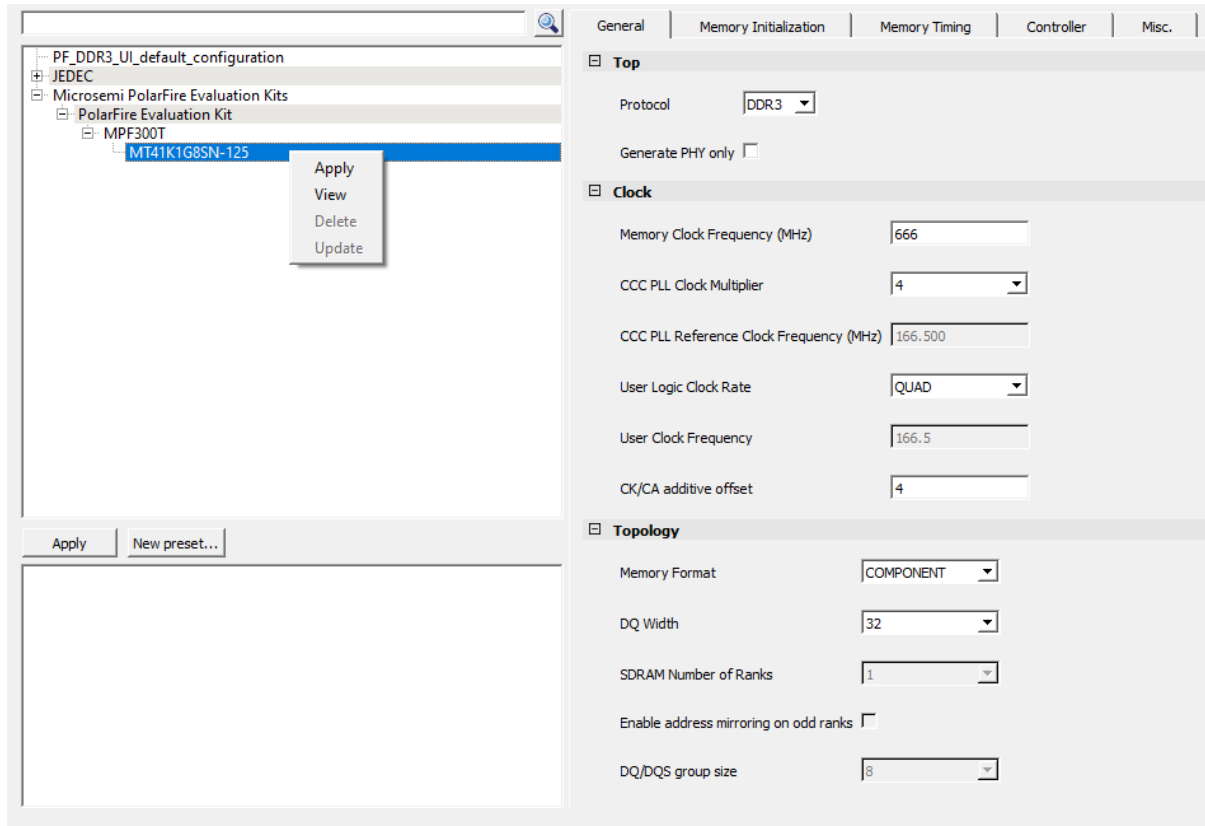
- Preset configuration—allows selection from a list of memory vendors and devices to preset all of the memory initialization and timing parameters into the DDR3 Configurator, as shown in [Figure 2.49](#).

---

 **Important:** For DDR3, preset configuration is supported for 1333, 1600, 1866, and 2133 at 666.67 MHz (minimum tCK  $\geq 1.5$  ns). For DDR4, preset configuration is supported for 1600, 1866, 2133, 2400, 2666, 2933, and 3200 at 800 MHz (minimum tCK  $\geq 1.25$  ns).

---

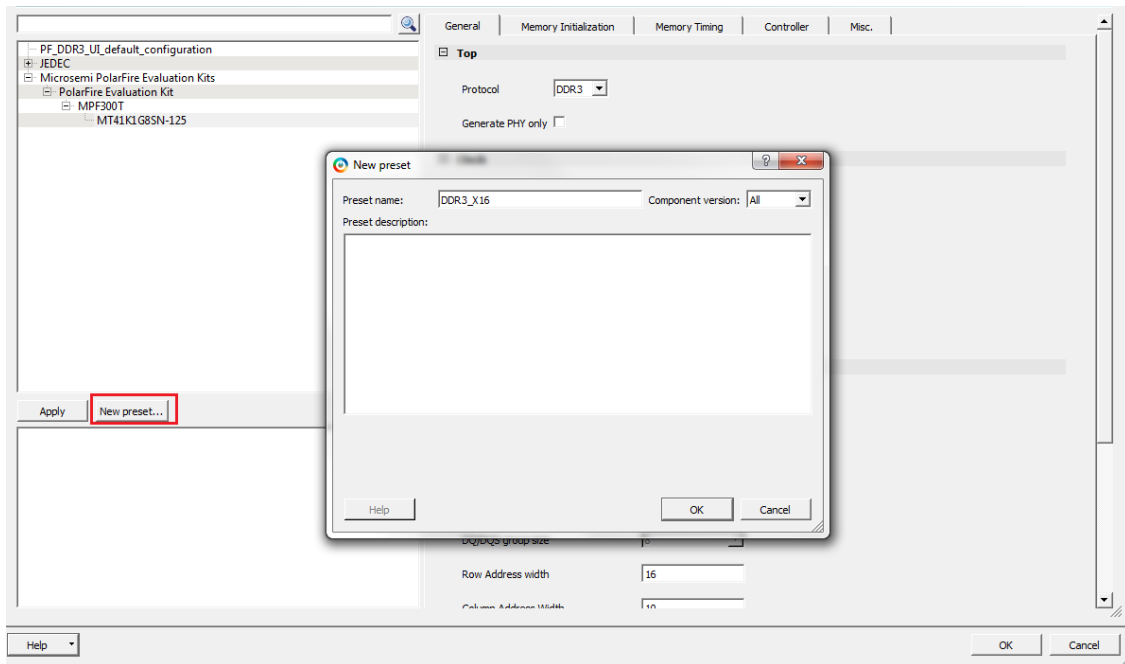
Figure 2.49. DDR3 Configurator—Preset Configuration



- User configuration—allows manual configuration of all memory initialization and timing parameters. This can be saved as a preset configuration, as shown in [Figure 2.50](#).

**➔ Important:** See the DDR3 vendor datasheets before configuring DDR3 parameters.

Figure 2.50. User-Defined Configuration



The DDR3 Configurator provides the following configuration options.

- [General](#)
- [Memory Initialization](#)
- [Memory Timing](#)
- [Controller](#)
- [Miscellaneous](#)

### General

The following figure shows the **General** tab.

Figure 2.51. PF\_DDR3 Configurator—General Tab

The screenshot displays the 'General' tab of the PF\_DDR3 Configurator. It is divided into three main sections: 'Top', 'Clock', and 'Topology'. Each section contains various configuration parameters, some as dropdown menus and others as text input fields. The 'Top' section has 'Protocol' set to 'DDR3' and 'Generate PHY only' unchecked. The 'Clock' section has 'Memory Clock Frequency (MHz)' at 666, 'CCC PLL Clock Multiplier' at 4, 'CCC PLL Reference Clock Frequency (MHz)' at 166.500, 'User Logic Clock Rate' at QUAD, 'User Clock Frequency' at 166.5, and 'CK/CA additive offset' at 4. The 'Topology' section has 'Memory Format' at COMPONENT, 'DQ Width' at 32, 'SDRAM Number of Ranks' at 1, 'Enable address mirroring on odd ranks' unchecked, and 'DQ/DQS group size' at 8.

As shown in the preceding figure, the **General** tab provides the following options.

- **Top**
  - **Protocol:** By default, this is set to DDR3.
  - **Generate PHY only:** Enable this option to create a PHY-only DDR3 subsystem.
- **Clock:** To set the DDR3 memory clock frequency and user logic clock rate. The user clock frequency is automatically populated.
  - **Memory Clock Frequency (MHz):** Valid range 0 to 667 MHz.
  - **CCC PLL Clock Multiplier:** Valid range 1 to 49.
  - **CCC PLL Reference Clock Frequency (MHz):** Depends upon the setting of **CCC PLL Clock Multiplier**.
  - **User Logic Clock Rate:** QUAD
  - **User Clock Frequency:** 166.5 MHz
  - **CK/CA additive offset:** The value must be within 0–7. The default value is 4.
- **Topology-** To set the topology of the memory system, including the address bits, bank configurations, DM modes, and ECC.
  - **Memory Format:** COMPONENT or type of DIMM (UDIMM, RDIMM, LRDIMM, or SODIMM)

- **DQ Width:** 16-bit, 32-bit, or 64-bit



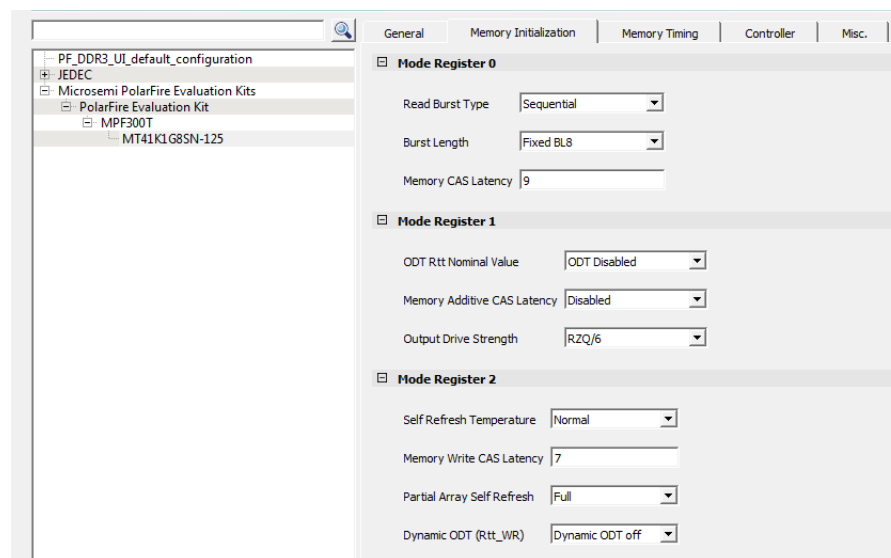
**Important:** AXI width must be larger than the data width and clock rate.

- **SDRAM Number of Ranks:** 1
- **Enable address mirroring on odd ranks:** Disabled
- **DQ/DQS group size:** 8 (fixed)
- **Row Address width:** Value must be within 12-16
- **Column Address Width:** Value must be within 5-12
- **Bank Address width:** 3 (fixed)
- **Enable DM:** OFF or DM
- **Enable Parity/Alert:** When the memory format is RDIMM or LRDIMM, this option can be selected to support the Alert signal from DDR memory when it detects a CA parity error.
- **Enable ECC:** Disabled
- **Number of clock outputs:** 1 (fixed)

### Memory Initialization

The **Memory Initialization** tab sets the DDR3 mode register configuration according to the JEDEC specification. In the Fabric DDR3 subsystem architecture, these parameters are passed to the DDR3 subsystem IP, which then performs the DDR3 initialization sequence and configures the mode registers. The following figure shows the **Memory Initialization** tab.

Figure 2.52. Memory Initialization Tab



As shown in the preceding figure, the **Memory Initialization** options provides the following options.

- **Mode Register 0**
  - **Read Burst Type:** Sequential or Interleaved

- **Burst length:** Fixed BL8



**Important:** On-the-fly BC4 or BL8 option for burst length mode is not supported for QUAD clock rate.

---

- **Memory CAS Latency:** Value must be within 5-14
- **Mode Register 1**
  - **ODT Rtt Nominal Value:** Any one of the following options can be selected:
    - ODT Disabled
    - RZQ/4
    - RZQ/2
    - RZQ/6
    - RZQ/12 for non-write
    - RZQ/8 for non-write
  - **Memory Additive CAS Latency:** Any one of the following options can be selected:
    - Disabled
    - CL-1
    - CL-2
  - **Output Drive Strength:** Any one of the following options can be selected:
    - RZQ/6
    - RZQ/7
- **Mode Register 2**
  - **Self Refresh Temperature :** Any one of the following options can be selected:
    - Normal
    - Extended
  - **Memory Write CAS Latency:** Value must be within 5–12
  - **Partial Array Self Refresh:** Any one of the following options can be selected:
    - Full
    - Half
    - Quarter
    - 1/8
    - Half (reversed)
    - Quarter (reversed)
    - 1/8 (reversed)
  - **Dynamic ODT (Rtt\_WR):** Any one of the following options can be selected:
    - Dynamic ODT off
    - RZQ/4
    - RZQ/2

## Memory Timing

The following figure shows the **Memory Timing** tab.

Figure 2.53. Memory Timing Tab

Parameter	Value
tRAS (ns)	36
tRCD (ns)	13.5
tRP (ns)	13.5
tRC (ns)	49.5
tWR (ns)	15.0
tFAW (ns)	45
tWTR (cycles)	5
tRRD (ns)	6
tRTP (ns)	7.5
tREFI (us)	7.8
tRFC (ns)	110.0
tZQinit (cycles)	512
ZQ Calibration Type	Short
tZQCS (cycles)	64
tZQoper (cycles)	256
Enable User ZQ Calibration Controls	<input type="checkbox"/>
Automatic ZQ Calibration Period (us)	200

As shown in the preceding figure, the **Controller** options provides the following options.

- **Timing parameters dependents on speed bin**

- **tRAS (ns)**: By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **tRCD (ns)**: By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **tRP (ns)**: By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **tRC (ns)**: By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.

- **tWR (ns):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **tFAW (ns):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **Timing parameters depends on speed bin and clock frequency**
  - **tWTR (cycles):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
  - **tRRD (ns):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
  - **tRTP (ns):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **Timing parameters depends on operating condition**
  - **tREFI (us):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **Timing parameters depends on speed bin and page size**
  - **tRFC (ns):** By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor data sheet.
- **Others Timing parameters**
  - **tZQinit (cycles):** By default, 512 is selected. The value must be within [512, 4095] range.
  - **ZQ Calibration Type:** Select Short or Long. By default, Short is selected.
  - **tZQCS (cycles):** By default, 64 is selected. The value must be within [64, 2047].
  - **tZQoper (cycles):** This value is fixed to 256.
  - **Enable User ZQ Calibration Controls:** Enable or Disable. By default, this option is disabled.
  - **Automatic ZQ Calibration Period (us):** By default, 200  $\mu$ s is selected. For every 200  $\mu$ s, ZQ calibration is triggered based on the ZQCL or ZQCS cycles selected.



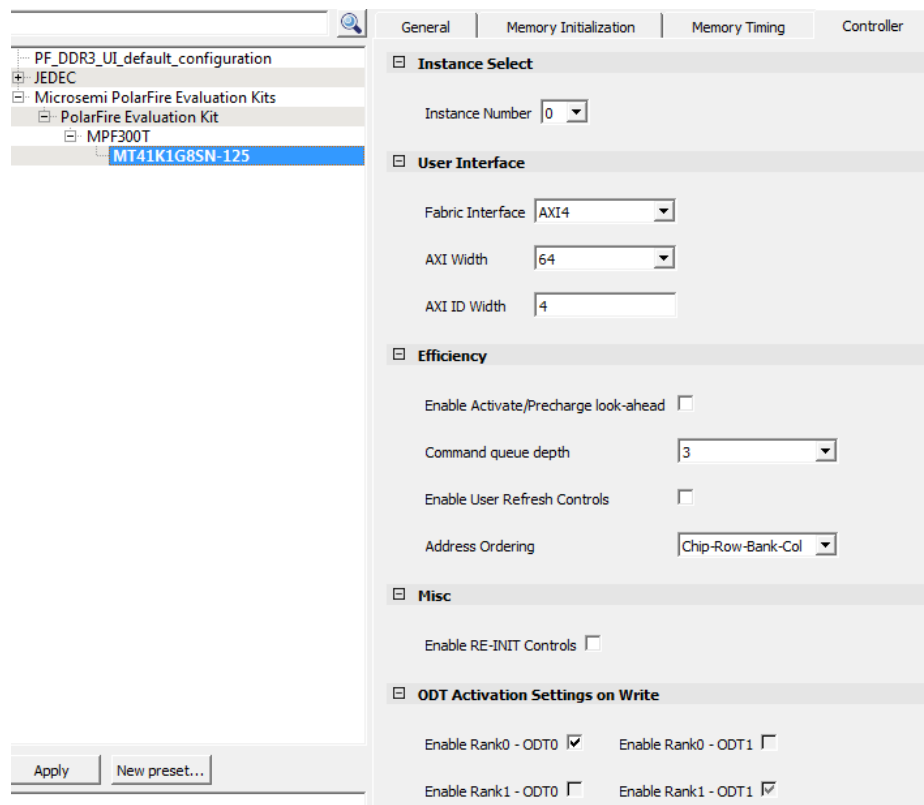
#### Important:

- By default, preset configuration for the MT41K1G8SN-125 part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- 

## Controller

The following figure shows the **Controller** tab.

Figure 2.54. Controller Options



As shown in the preceding figure, the **Controller** options provides the following options.

- **Instance Select:** Enables selection of the DDR3 subsystem instance number that is used to unify the controller instance. The PolarFire family supports a maximum of six DDR3 subsystem instances.
  - **Instance Number:** Up to eight instances can be generated.
- **User Interface:**
  - **Fabric 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-2.28. Supported AXI Bus Data Width

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

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-2.29.** AXI Address Mapping

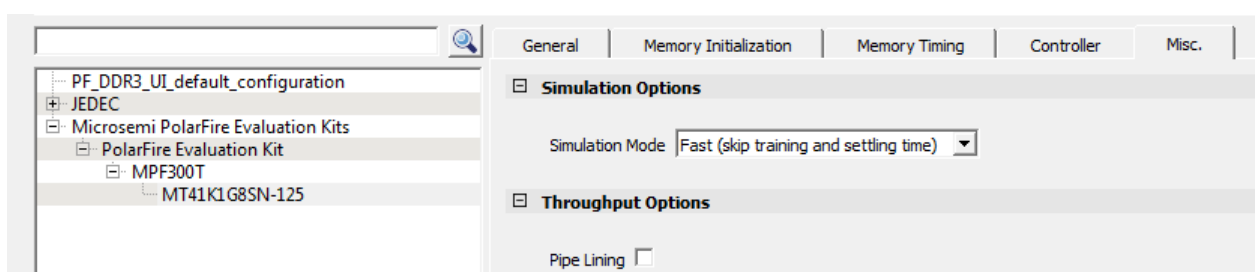
AXI Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Column Address																					C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	
Bank Address																			BA2	BA1	BA0											
Row Address				R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0																

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

- **AXI Width:** 64, 128, 256, or 512
- **AXI ID Width:** Valid range is 1-8
- **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 DDR3 Configurator > Controller tab.
  - **Enable Activate/Precharge look-head:** Check box toggle option
  - **Command queue depth:** 3 or 4
  - **Enable User Refresh Controls:** Check box toggle option
  - **Address Ordering:** Any one of the following options is supported:
    - Chip-Bank-Row-Col
    - Chip-Row-Bank-Col
- **Misc**
  - **Enable RE-INIT Controls:** When selected, the configurator exposes the CTRLR\_INIT signal. The initialization begins when this signal is asserted high for minimum 4 clock cycles in the fabric clock domain.
- **ODT Activation Setting on Write:** Provides options to enable ODT on write operations.
  - **Enable Rank0 - ODT0:** By default, this option is enabled and can be disabled.
  - **Enable Rank0 - ODT1:** Not supported
  - **Enable Rank1 - ODT0:** Not supported
  - **Enable Rank1 - ODT1:** By default, this option is enabled and cannot be disabled.

### Miscellaneous

The following figure shows the **Misc.** tab.

**Figure 2.55.** Misc. Tab

The **Misc.** tab is used to select the simulation mode and expose the debug interface. As shown in the preceding figure, the **Misc.** options provides the following options.

- **Simulation Options**
  - **Simulation mode:**
    - **Fast (skip training and settling time):** In this mode, the DDR3 subsystem skips the DDR3 training sequence and asserts CTRLR\_READY without waiting for the settling time. CTRLR\_READY is asserted within 8  $\mu$ s.
    - **Training (skip settling time):** In this mode, the DDR3 subsystem performs the DDR3 training sequence but asserts CTRLR\_READY without waiting for settling time. CTRLR\_READY is asserted within 240  $\mu$ s.
    - **Full (training and settling time):** In this mode, the DDR3 subsystem performs the DDR3 training sequence and also waits for settling time before asserting CTRLR\_READY. In this mode, CTRLR\_READY is asserted within 10 ms.
- **Throughput Options**
  - **Pipe Lining:** Adds pipeline registers to Training IP (TIP) write and read data path to improve static timing closure.

### 2.9.3. Configuring the LPDDR3 Subsystem [\(Ask a Question\)](#)

The PF\_LPDDR3 macro located in the Libero IP catalog must be instantiated in SmartDesign to access the LPDDR3 memory from the FPGA fabric. The following figure shows the LPDDR3 configurator that configures the LPDDR3 subsystem. It supports the following modes:

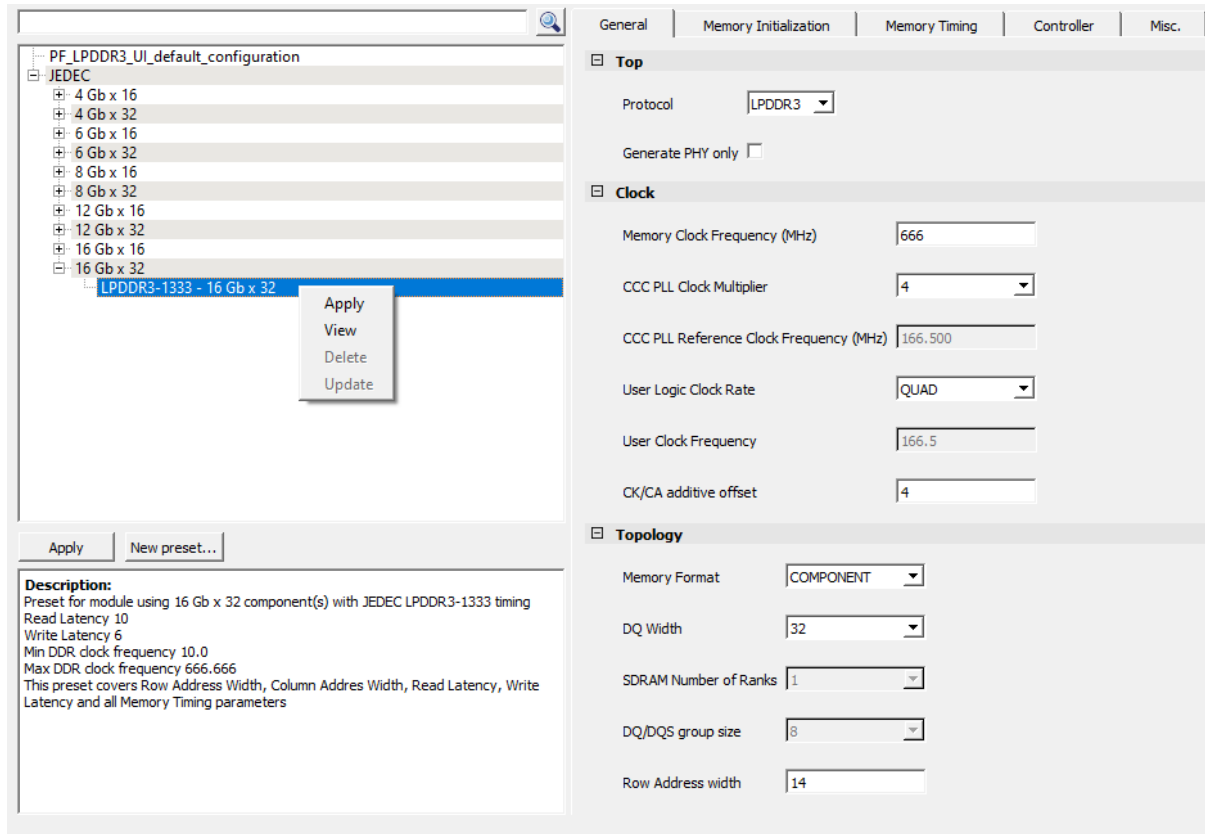
- Preset configuration—allows selection from a list of memory vendors and devices to preset all of the memory initialization and timing parameters into the DDR Configurator, as shown in [Figure 2.56](#).



**Important:** For LPDDR3, preset configuration is supported for 1333 at 666.666 MHz (minimum tCK  $\geq$  1.5 ns).

---

Figure 2.56. LPDDR3 Configurator—Preset Configuration



- User configuration—allows manual configuration of all memory initialization and timing parameters. This can be saved as a preset configuration, as shown in Figure 2.56. See the DDR vendor datasheets before configuring DDR parameters.

The LPDDR3 Configurator provides the following options.

- [General](#)
- [Memory Initialization](#)
- [Memory Timing](#)
- [Controller](#)
- [Miscellaneous](#)

### General

The following figure shows the **General** tab.

Figure 2.57. LPDDR3 General Tab

Section	Parameter	Value
Top	Protocol	LPDDR3
	Generate PHY only	<input type="checkbox"/>
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
	CK/CA additive offset	4
Topology	Memory Format	COMPONENT
	DQ Width	32
	SDRAM Number of Ranks	1
	DQ/DQS group size	8
	Row Address width	14

As shown in the preceding figure, the **General** tab provides the following options.

- **Top**
  - **Protocol:** By default, this is set to LPDDR3.
  - **Generate PHY only:** Enable this option to create a PHY-only LPDDR3 subsystem.
- **Clock:** To set the LPDDR3 memory clock frequency and user logic clock rate. The user clock frequency is automatically populated.
  - **Memory Clock Frequency (MHz):** Valid range is 400–667 MHz
  - **CCC PLL Clock Multiplier:** Valid range 1 to 49
  - **CCC PLL Reference Clock Frequency (MHz):** Depends on the setting of **CCC PLL Clock Multiplier**.
  - **User Logic Clock Rate:** QUAD
  - **User Clock Frequency:** 166.5 MHz
  - **CK/CA additive offset:** The value must be within 0–7. The default value is 4.
- **Topology:** To set the topology of the memory system, including the address bits, bank configurations, DM modes, and ECC.
  - **Memory Format:** COMPONENT

- **DQ Width:** Select any one of the following options:
  - 16
  - 32
- **SDRAM Number of Ranks:** 1
- **DQ/DQS group size:** 8
- **Row Address width:** The value must be within [14, 15] range
- **Column Address Width:** The value must be within [10, 12] range
- **Bank Address width:** 3
- **Enable DM:** Select any one of the following options:
  - OFF
  - DM
- **Number of clock outputs:** 1

### Memory Initialization

The following figure shows the **Memory Initialization** tab.

Figure 2.58. LPDDR3 Memory Initialization Tab

The screenshot shows a configuration window with five tabs: General, Memory Initialization (selected), Memory Timing, Controller, and Misc. The Memory Initialization tab is expanded to show three sections:

- Mode Register 2:** Data Latency (#RL, #WL) is set to (10, 6).
- Mode Register 3:** Output Drive Strength is set to 40 typical pull-down/pull-up.
- Mode Register 11:** DQ ODT is set to Disable, and Power down ODT is set to Disabled.

As shown in the preceding figure, the **Memory Initialization** tab provides the following options.

- **Mode Register 2**
  - **Data Latency (#RL, #WL):** Select any one of the following options:
    - (6, 3)
    - (8, 4)
    - (9, 5)
    - (10, 6)
    - (11, 6)
    - (12, 6)

By Default, (10, 6) is selected.

- **Mode Register 3**
  - **Output Drive Strength:** Select any one of the following options:
    - 34.3 typical pull-down/pull-up
    - 40 typical pull-down/pull-up
    - 48 typical pull-down/pull-up
    - 34.3 typical pull-down, 40 typical pull-up
    - 40 typical pull-down, 48 typical pull-up
    - 34.3 typical pull-down, 48 typical pull-up
- **Mode Register 11**
  - **DQ ODT:** Select any one of the following options:
    - Disable
    - RZQ/4
    - RZQ/2
    - RZQ/1
  - **Power down ODT:** Disabled or Enabled

### Memory Timing

The following figure shows the **Memory Timing** tab.

Figure 2.59. LPDDR3 Memory Timing Tab

General	Memory Initialization	Memory Timing	Controller	Misc.
<b>Timing parameters dependent on speed bin</b>				
tRAS (ns)		42		
tRCD (ns)		15		
tRP (ns)		15		
tRC (ns)		57		
tWR (ns)		18.0		
tFAW (ns)		50.0		
tMRR (cycles)		4		
tMRW (cycles)		10		
<b>Timing parameters dependent on speed bin and clock frequency</b>				
tWTR (cycles)		4		
tRRD (ns)		10.0		
tRTP (ns)		8.0		
<b>Timing parameters dependent on operating condition</b>				
tREFI (us)		3.9		
<b>Timing parameters dependent on speed bin and page size</b>				
<b>Other Timing parameters</b>				
tZQinit (us)		1		
ZQ Calibration Type		Long		
tZQCS (ns)		90.0		
tZQCL (ns)		360.0		
tZQRESET (ns)		50.0		

As shown in the preceding figure, the **Memory Timing** tab provides the following options.

- **Timing parameters dependents on speed bin**
  - **tRAS (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tRCD (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tRP (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tRC (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tWR (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tRP (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tFAW(ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tMRR (cycles)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tMRW (cycles)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **Timing parameters dependents on speed bin and clock frequency**
  - **tWTR (cycles)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tRRD (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
  - **tRTP (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **Timing parameters dependents on operating condition**
  - **tREFI (us)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **Timing parameters dependents on speed bin and page size**
  - **tRFC (ns)**: Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **Others Timing parameters**
  - **tZQinit (us)**: By default, 1 is selected. The value must be within [1, 6] range.
  - **ZQ Calibration Type**: Select Short or Long. By default, Short is selected.
  - **tZQCS (ns)**: By default, 90 is selected. The value must be within 2047.
  - **tZQCL (ns)**: This value is fixed to 360.
  - **tZQRESET (ns)**: By default, 50 is selected. The value must be within 2047.
  - **Enable User ZQ Calibration Controls**: Select this check box for using user-defined ZQ Calibration Controls
  - **Automatic ZQ Calibration Period (us)**: By default, 200  $\mu$ s is selected. For every 200  $\mu$ s, ZQ calibration is triggered based on the ZQCL or ZQCS cycles selected.

**Important:**

- By default, preset configuration is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.

**Controller**

The following figure shows the **Controller** tab.

**Figure 2.60.** LPDDR3 Controller Tab

As shown in the preceding figure, the **Controller** tab provides the following options.

- **Instance Select:** Enables the option to select of the LPDDR3 subsystem instance number that is used to unify the controller instance. PolarFire family supports a maximum of six LPDDR3 subsystem instances.
  - **Instance Number:**
- **User Interface**
  - **Fabric 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-2.30.** Supported AXI Bus Data Width

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

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\_LPDDR3 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-2.31.** 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																



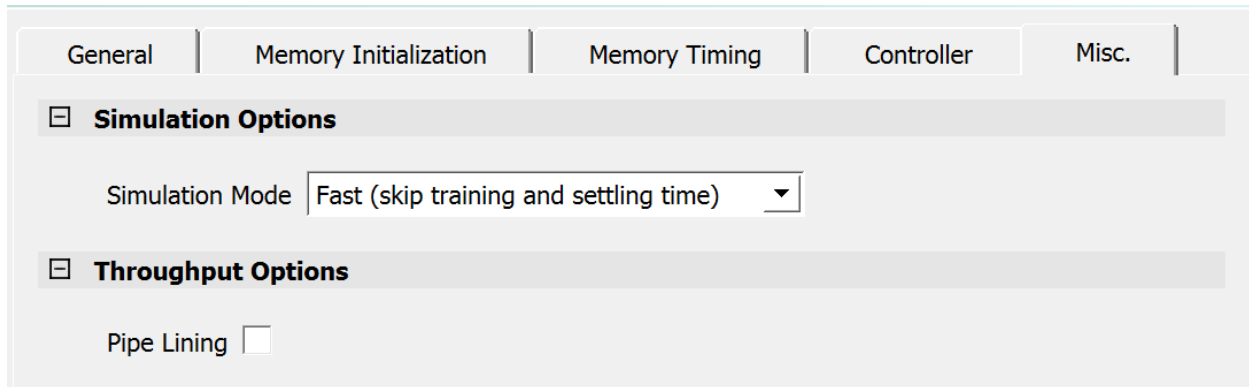
**Important:** The address mapping shown in the preceding table is applicable to Native interface also.

- **AXI Width:** 64, 128, or 256
- **AXI ID Width:** Valid range is 1–8
- **Efficiency:** Provides control over memory refresh, precharge, and address ordering options.
  - **Enable Activate/Precharge look-head:** Check box toggle option
  - **Enable User Refresh Controls:** Check box toggle option
  - **Address Ordering:** Any one of the following options is supported:
    - Chip-Bank-Row-Col
    - Chip-Row-Bank-Col
- **Misc**
  - **Enable RE-INIT Controls:** When selected, the configurator exposes the CTRLR\_INIT signal. The initialization begins when this signal is asserted high for minimum 4 clock cycles in the fabric clock domain.
- **ODT Activation Setting on Write:** Provides options to enable ODT on write operations.
  - **Enable Rank0 - ODT0:** Supported
  - **Enable Rank0 - ODT1:** Not supported
  - **Enable Rank1 - ODT0:** Not supported
  - **Enable Rank1 - ODT1:** Not supported
- **Low Power**
  - **Enable self request refresh control:** Use this option to save power.

### Miscellaneous

The **Misc.** tab provides the following options.

Figure 2.61. LPDDR3 Simulation Tab



As shown in the preceding figure, the **Misc.** tab provides the following options.

- **Simulation Options**
  - **Simulation mode:**
    - **Fast (skip training and settling time):** In this mode, the LPDDR3 subsystem skips the LPDDR3 training sequence and asserts CTRLR\_READY without waiting for the settling time. CTRLR\_READY is asserted within 8  $\mu$ s.
    - **Training (skip settling time):** In this mode, the LPDDR3 subsystem performs the LPDDR3 training sequence but asserts CTRLR\_READY without waiting for settling time. CTRLR\_READY is asserted within 240  $\mu$ s.
    - **Full (training and settling time):** In this mode, the LPDDR3 subsystem performs the LPDDR3 training sequence and also waits for settling time before asserting CTRLR\_READY. In this mode, CTRLR\_READY is asserted within 10 ms.
- **Throughput Options**
  - **Pipe Lining:** Adds pipeline registers to Training IP (TIP) write and read data path to improve static timing closure.

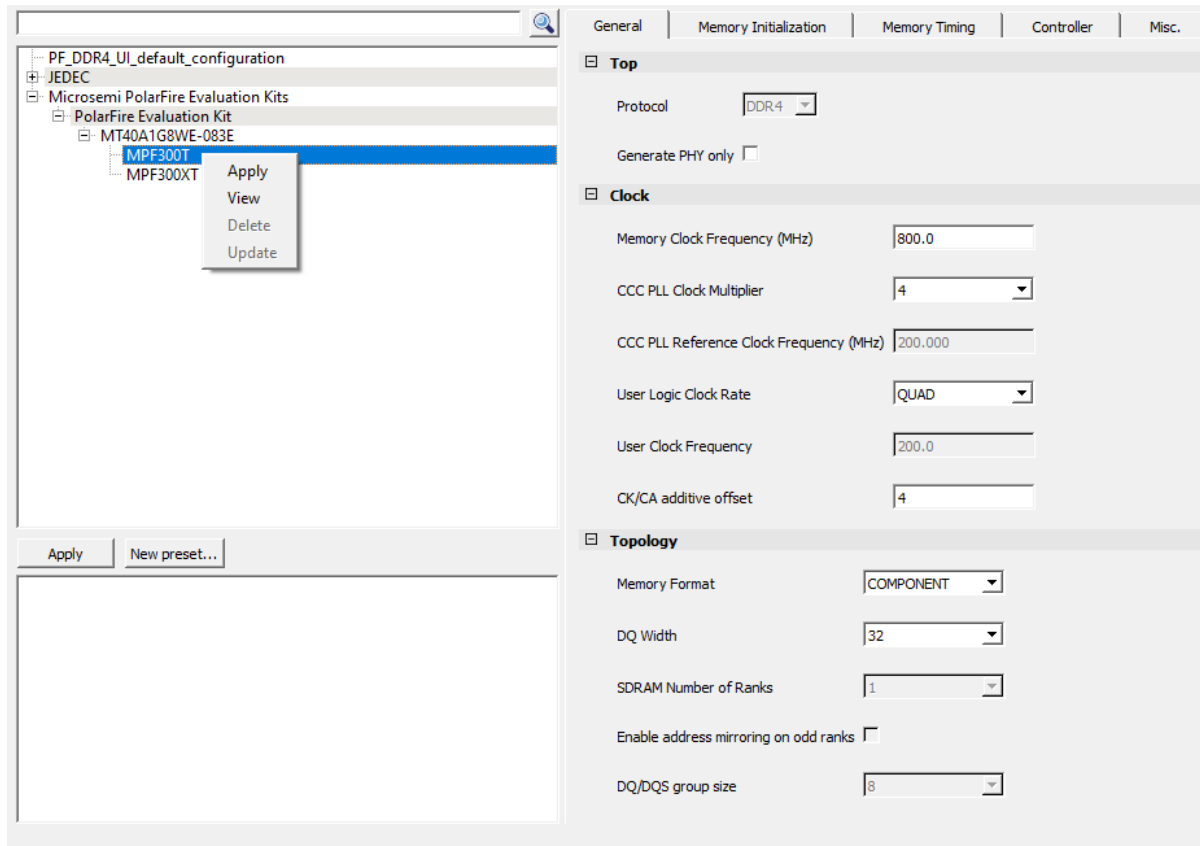
#### 2.9.4. Configuring the DDR4 Subsystem [\(Ask a Question\)](#)

The PF\_DDR4 macro located in the Libero IP catalog must be instantiated in SmartDesign to access the DDR memory from the FPGA fabric. The following figure shows the DDR4 Configurator that configures the DDR4 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 2.62](#).

**➔ Important:** For DDR4, preset configuration is supported for 1600, 1866, 2133, 2400, 2666, 2933, and 3200 at 800 MHz (minimum tCK ≥ 1.25 ns).

**Figure 2.62.** PF\_DDR4 Configurator—Preset Configuration



- User configuration—allows manual configuration of all memory initialization and timing parameters. This can be saved as a preset configuration, as shown in the preceding. See the DDR4 vendor datasheets before configuring DDR4 parameters.

The DDR4 Configurator provides the following configuration.

- [General](#)
- [Memory Initialization](#)
- [Memory Timing](#)
- [Controller](#)
- [Miscellaneous](#)

### General

The following figure shows the **General** tab.

Figure 2.63. DDR4 General Tab

As shown in the preceding figure, the **General** tab provides the following options.

- **Top**
  - **Protocol:** By default, this is set to DDR4.
  - **Generate PHY only:** Enable this option to create a PHY-only DDR4 subsystem.
- **Clock:** To set the DDR4 memory clock frequency and user logic clock rate. The user clock frequency is automatically populated.
  - **Memory Clock Frequency (MHz):** By default, the clock frequency is set to 800 MHz. The value must be within 0–800 MHz
  - **CCC PLL Clock Multiplier:** Valid range is 1–49
  - **CCC PLL Reference Clock Frequency (MHz) :** Depends upon **Memory Clock Frequency (MHz)** and **CCC PLL Clock Multiplier**. The value is 200 MHz when memory frequency is 800 MHz and clock multiplier is 4.
  - **User Logic Clock Rate:** QUAD
  - **User Clock Frequency:** Depends upon **Memory Clock Frequency (MHz)** and **CCC PLL Clock Multiplier**. The value is 200 MHz when memory frequency is 800 MHz and clock multiplier is 4.
  - **CK/CA additive offset:** The value must be within 0–7. The default value is 4.

- **Topology:** To set the topology of the memory system, including the address bits, bank configurations, DM modes, and ECC.
  - **Memory Format:** COMPONENT or type of DIMM (UDIMM, RDIMM, LRDIMM, or SODIMM)
  - **DQ Width:** 16-bit, 32-bit, or 64-bit



**Important:** AXI width must be larger than the data width and clock rate.

---

- **SDRAM Number of Ranks:** 1
- **Enable Address mirroring on odd ranks:** Disabled
- **DQ/DQS group size:** 8
- **Row Address width:** The value must be within [13, 17] range
- **Column Address Width:** 10
- **Bank Address width:** 2
- **Bank Group Address width:** The value must be within [1, 2] range
- **Enable DM:** OFF or DM (enabled)
- **Enable Parity/Alert:** When the memory format is RDIMM or LRDIMM, this option can be selected to support the Alert signal from DDR memory when it detects a CA parity error.
- **Enable ECC:** Check box toggle option
- **Number of clock outputs:** 1

### Memory Initialization

The following figure shows the **Memory Initialization** tab.

Figure 2.64. DDR4 Memory Initialization Tab

General	Memory Initialization	Memory Timing	Controller	Misc.
<b>Mode Register 0</b>				
Burst Length	Fixed BL8			
Read Burst Type	Sequential			
Memory CAS Latency	12			
<b>Mode Register 1</b>				
ODT Rtt Nominal Value	RZQ/4			
Memory Additive CAS Latency	Disabled			
Output Drive Strength	RZQ/7			
<b>Mode Register 2</b>				
Low Power Auto Self Refresh	Automatic			
Memory Write CAS Latency	11			
Dynamic ODT (Rtt_WR)	Disabled			
<b>Mode Register 3</b>				
Fine Granularity Refresh Mode	Normal mode (Fixed 1x)			
<b>Mode Register 4</b>				
Temperature Refresh Range	Normal			
Temperature Refresh Mode	Disabled			
Internal VRef Monitor	Disabled			
Self Refresh Abort Mode	Disabled			
READ Preamble	1CK			
WRITE Preamble	1CK			
<b>Mode Register 5</b>				
CA Parity Latency Mode	Disabled			

As shown in the preceding figure, the **Memory Initialization** tab provides the following options.

- **Mode Register 0**

- **Burst length:** Fixed BL8 or On-the-fly BC4 or BL8



**Important:** On-the-fly BC4 or BL8 option for burst length mode is not supported for QUAD clock rate.

---

- **Read Burst Type:** Sequential or Interleaved
- **Memory CAS Latency:** Value must be within 9–32

- **Mode Register 1**

- **ODT Rtt Nominal Value:** Any one of the following options can be selected:
  - Disabled
  - RZQ/4
  - RZQ/2
  - RZQ/6
  - RZQ/1
  - RZQ/5
  - RZQ/3
  - RZQ/7
- **Memory Additive CAS Latency:** Any one of the following option can be selected:
  - Disabled
  - CL-1
  - CL-2
- **Output Drive Strength:** RZQ/7 or RZQ/5

- **Mode Register 2**

- **Low Power Auto Self Refresh :** Any one of the following options can be selected:
  - Manual - Normal
  - Manual - Reduced
  - Manual - Extended
  - Automatic
- **Memory Write CAS Latency:** Any one of the following options can be selected:
  - 9
  - 10
  - 11
  - 12
  - 14
  - 16
  - 18
  - 20
- **Dynamic ODT (Rtt\_WR):** Any one of the following option can be selected:
  - Disabled

- RZQ/2
- RZQ/1
- High-Z
- RZQ/3
- **Mode Register 3**
  - **Fine Granularity Refresh Mode:** Any one of the following options can be selected:
    - Normal mode (Fixed 1x)
    - Fixed 2x
    - Fixed 4x
    - On-the-fly 1x/2x
    - On-the-fly 1x/4x
- **Mode Register 4**
  - **Temperature Refresh Range:** Normal or Extended
  - **Temperature Refresh Mode:** Disabled or Enabled
  - **Internal VRef Monitor:** Disabled or Enabled
  - **Self Refresh Abort Mode:** Disabled or Enabled
  - **Read Preamble:** 1CK or 2CK
  - **Write Preamble:** 1CK
- **Mode Register 5**
  - **CA Parity Latency Mode:** Any one of the following options can be selected:
    - Disabled
    - 4 clocks
    - 5 clocks
    - 6 clocks
    - 8 clocks
  - **ODT Input Buffer for Power-down:** Disabled or Enabled
  - **Park ODT Value(Rtt\_Park):** Any one of the following options can be selected:
    - Disabled
    - RZQ/4
    - RZQ/2
    - RZQ/6
    - RZQ/1
    - RZQ/5
    - RZQ/3
    - RZQ/7
- **Mode Register 6**
  - **Vref Calibration Range:** Range1 (60%–92.5%)
  - **Vref Calibration Value:** 70.40

### Memory Timing

The following figure shows the **Memory Timing** tab.

Figure 2.65. DDR4 Memory Timing Tab

General	Memory Initialization	Memory Timing	Controller	Misc.
<b>Timing parameters dependent on speed bin</b>				
tRAS (ns)		<input type="text" value="35"/>		
tRCD (ns)		<input type="text" value="15"/>		
tRP (ns)		<input type="text" value="15"/>		
tRC (ns)		<input type="text" value="50"/>		
tWR (ns)		<input type="text" value="15.0"/>		
tCCD_L (cycles)		<input type="text" value="5"/>		
tCCD_S (cycles)		<input type="text" value="4"/>		
<b>Timing parameters dependent on operating condition</b>				
tREFI (us)		<input type="text" value="7.8"/>		
<b>Timing parameters dependent on speed bin and page size</b>				
tRFC (ns)		<input type="text" value="160.0"/>		
tFAW (ns)		<input type="text" value="25"/>		
<b>Timing parameters dependent on speed bin and clock frequency</b>				
tWTR_L (cycles)		<input type="text" value="6"/>		
tWTR_S (cycles)		<input type="text" value="2"/>		
tRRD_L (cycles)		<input type="text" value="5"/>		
tRRD_S (cycles)		<input type="text" value="4"/>		
tRTP (ns)		<input type="text" value="7.5"/>		
<b>Other Timing parameters</b>				
tZQinit (cycles)		<input type="text" value="1024"/>		
ZQ Calibration Type		<input type="text" value="Short"/>		

As shown in the preceding figure, the **Memory Timing** options provides the following options.

- **Timing parameters dependents on speed bin**

- **tRAS (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tRCD (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tRP (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tRC (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tWR (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tCCD\_L (cycles):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.



**Important:** tCCD\_L is equivalent to the data rate in the JEDEC specification. tCCD\_L parameter field accepts a value in the range of 4–20, and the value entered gets routed to Mode Register 6 (MR6) bits[12:10].

- **tCCD\_S (cycles):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.

- **Timing parameters dependents on operating condition**

- **tREFI (us):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.

- **Timing parameters dependents on speed bin and page size**

- **tRFC (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tFAW (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.

- **Timing parameters dependents on speed bin and clock frequency**

- **tWTR\_L (cycles):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tWTR\_S (cycles):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.

- **tRRD\_L (cycles):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tRRD\_S (cycles):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **tRTP (ns):** By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- **Others Timing parameters**
  - **tZQinit (cycles):** By default, 1024 is selected. The value must be within [1024, 4095] range
  - **ZQ Calibration Type:** Short or Long
  - **tZQCS (cycles):** By default, 128 is selected. The value must be within [128, 2047]
  - **tZQoper (cycles):** This value is fixed to 512.
  - **Enable User ZQ Calibration Controls:** Enable or Disable. By default, this option is disabled.
  - **Automatic ZQ Calibration Period (us):** By default, 200  $\mu$ s is selected. For every 200  $\mu$ s, ZQ calibration is triggered based on the ZQCL or ZQCS cycles selected.

**Important:**

- By default, preset configuration for the MT40A1G8WE-083E part is selected. Preset values for different configurations according to the JEDEC standard are supported. See the speed bin and its corresponding value in the vendor datasheet.
- 

## Controller

The following figure shows the **Controller** tab.

Figure 2.66. DDR4 Controller Tab

The screenshot shows the 'Controller' tab of the DDR4 Controller configuration. The tabs at the top are General, Memory Initialization, Memory Timing, Controller (selected), and Misc. The configuration is organized into several sections:

- Instance Select:** Instance Number is set to 0.
- User Interface:** Fabric Interface is set to AXI4, AXI Width is 64, and AXI ID Width is 4.
- Efficiency:** Enable Activate/Precharge look-ahead is unchecked. Number of banks to lookup is 4. Address Ordering is set to Chip-Row-BG-Bank-Col.
- ODT Activation Settings on Write:** Enable Rank0 - ODT0 is checked, Enable Rank0 - ODT1 is unchecked, Enable Rank1 - ODT0 is unchecked, and Enable Rank1 - ODT1 is checked.
- Low Power:** Enable self refresh request control is unchecked.
- Misc:** Enable RE-INIT Controls is unchecked.

As shown in the preceding figure, the **Controller** tab provides the following options.

- **Instance Select:** Enables the option to select the DDR4 subsystem instance number that is used to unify the controller instance. PolarFire family supports a maximum of six DDR4 subsystem instances.
  - **Instance Number:**
- **User Interface**
  - **Fabric 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-2.32. Supported AXI Bus Data Width

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

**Table 2.32.** Supported AXI Bus Data Width (continued)

DQ Width	Supported AXI Data Bus Width
x64	512

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\_DDR4 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-2.33.** 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																



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

- **AXI Width:** 64, 128, 256, or 512
- **AXI ID Width:** Valid range is 1–8
- **Efficiency:** Provides control over memory refresh, precharge, and address ordering options.
  - **Enable Activate/Precharge look-head:** Check box toggle option
  - **Number of Banks to lookup:** 4, 8, or 16
  - **Address Ordering:** Any one of the following options can be selected:
    - Chip-Row-BG-Bank-Col
    - Chip-BG-Bank-Row-Col
    - Chip-Row-BG1-Bank-Col-BG0



**Important:** Number of banks to lookup must be 8 or 16 when address ordering is set to Chip-Row-BG1-Bank-Col-BG0.

- **ODT Activation Setting on Write:** provides options to enable ODT on write operations.
  - **Enable Rank0 - ODT0:** Enabled by default. Can Disabled or enabled.
  - **Enable Rank0 - ODT1:** Not supported
  - **Enable Rank1 - ODT0:** Not supported
  - **Enable Rank1 - ODT1:** Enabled (fixed).
- **Low Power**
  - **Enable self request refresh control:** When selected, this option saves power
- **Misc**

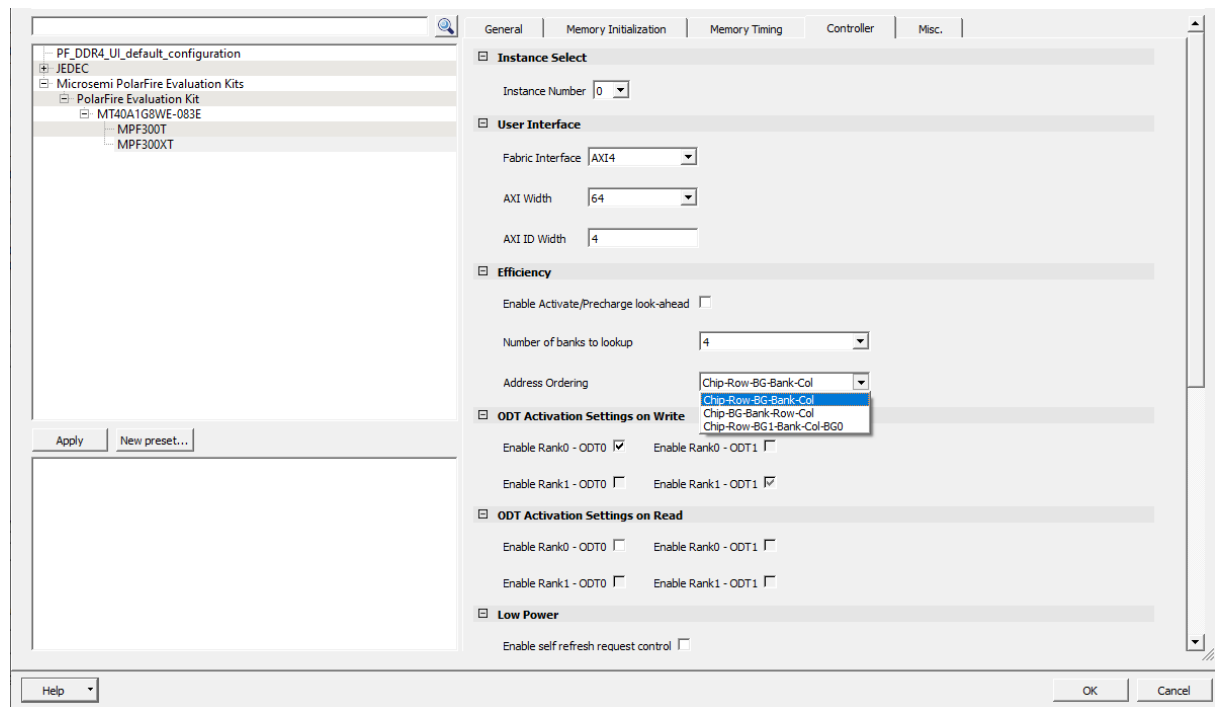
- **Enable RE-INIT Controls:** When selected, the configurator exposes the CTRLR\_INIT signal. The initialization begins when this signal is asserted high for minimum 4 clock cycles in the fabric clock domain.

### DDR4 Additional Controller Options

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

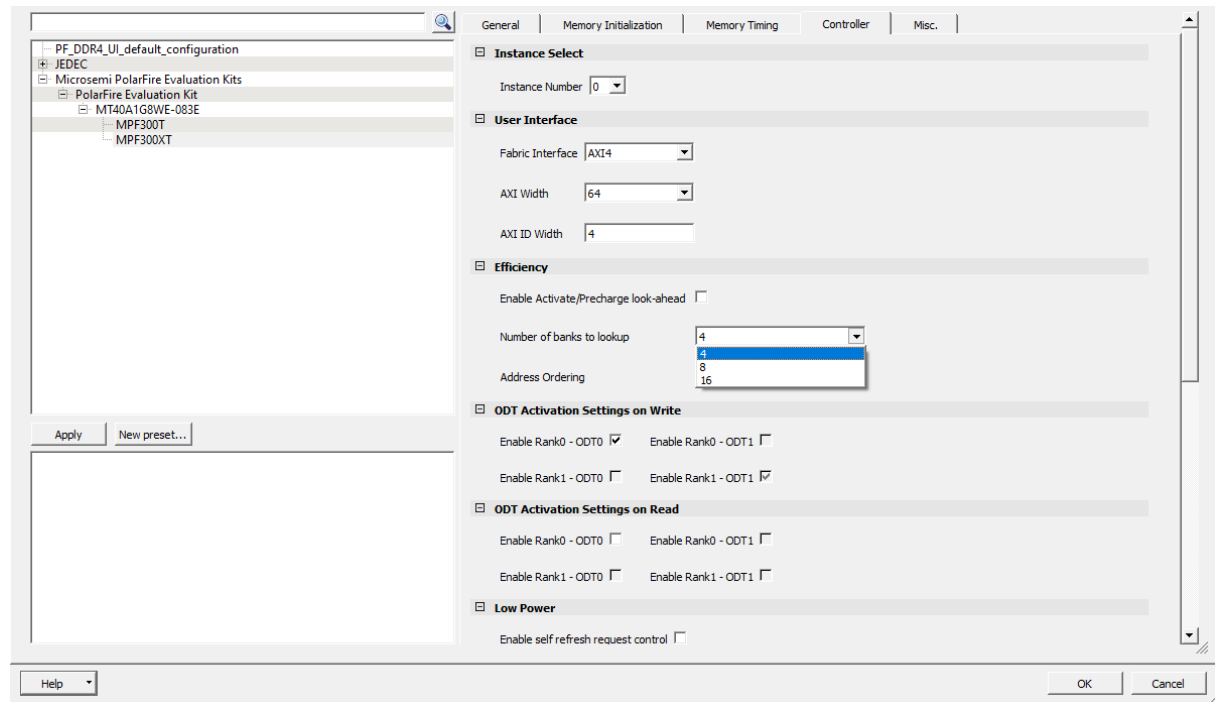
- The **Address Ordering** includes the additional setting Chip-Row-BG1-Bank-Col-BG0. This selection will map the LSBs of an address to the BG0 field of the memory. This results in alternating Bank Groups during a burst operation based on the tCCD\_S timing parameter (see memory device datasheet for 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 2.67.** Address Ordering—Chip-Row-BG1-Bank-Col-BG0



- The **Number of banks to look up** option includes additional selections for 8 and 16. The default selection is 4. This must 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 of 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.

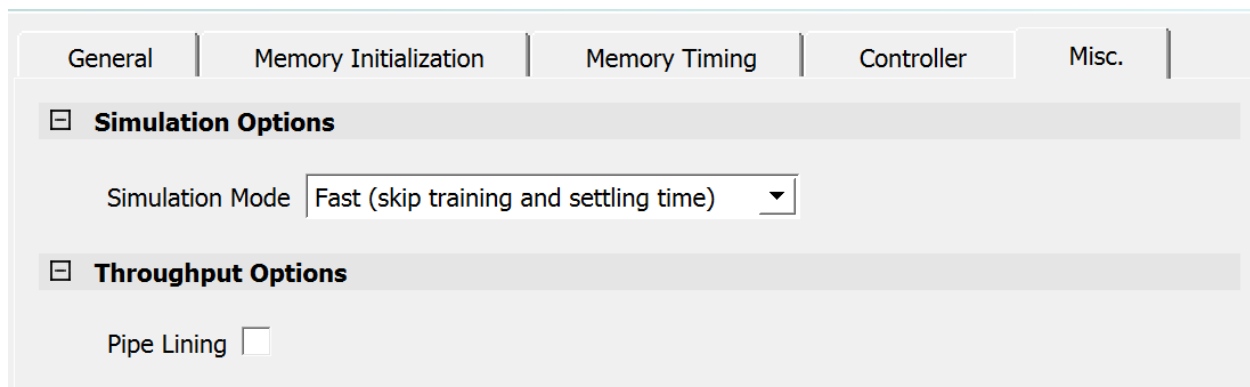
Figure 2.68. Number of Banks to Look Up



## Miscellaneous

The following figure shows the **Misc.** tab.

Figure 2.69. DDR4 Simulation Tab



As shown in the preceding figure, the **Misc.** options provides the following options.

- **Simulation Options**

- **Simulation mode:**

- **Fast (skip training and settling time):** In this mode, the DDR4 subsystem skips the DDR4 training sequence and asserts CTRLR\_READY without waiting for the settling time. CTRLR\_READY is asserted within 8  $\mu$ s.
    - **Training (skip settling time):** In this mode, the DDR4 subsystem performs the DDR4 training sequence but asserts CTRLR\_READY without waiting for settling time. CTRLR\_READY is asserted within 240  $\mu$ s.

- **Full (training and settling time):** In this mode, the DDR4 subsystem performs the DDR4 training sequence and also waits for settling time before asserting CTRLR\_READY. In this mode, CTRLR\_READY is asserted within 10 ms.
- **Throughput Options**
  - **Pipe Lining:** Adds pipeline registers to Training IP (TIP) write and read data path to improve static timing closure.

### 2.9.5. Simulating the DDR Subsystem [\(Ask a Question\)](#)

Libero SoC provides a simulation model for the DDR subsystem, which includes a training sequence (three simulation modes are supported). 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 [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.

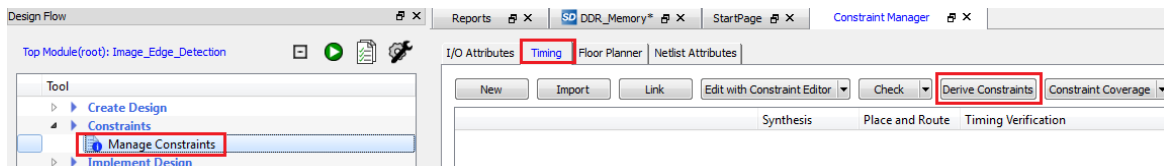
### 2.9.6. Design Constraints [\(Ask a Question\)](#)

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

#### 2.9.6.1. Timing Constraints [\(Ask a Question\)](#)

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 2.70.** Constraint Manager—Derive Constraints Tab



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

**➔ Important:** Synthesizing the PF\_DDR4 memory controller generates the “CL118” warning, which reports the generation of a latch for the `apb_data_out[7:0]` signal in the `dq_align_dqs_optimization.v` file. The issue originates from the unused COREDDR\_TIP APB interface, which is not utilized. This warning does not impact the functionality of the PF\_DDR4 memory controller and can be ignored by the user.

#### 2.9.6.2. Physical Constraints [\(Ask a Question\)](#)

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.



**Important:** Enabling Synplify soft Triple Module Redundancy (TMR) option, `syn_radhardlevel` on these memory controllers is not recommended. Performance impacts due to the spatial TMR of the logic can result in timing closure failures in these high performance controllers. The recommendation is to monitor the memory subsystem for signs of SEU failures during normal operation. If failure is detected, reset the subsystem.

#### 2.9.6.2.1. Using PDC [\(Ask a Question\)](#)

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 {PF_DDR3_C0_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 packages in the PolarFire family. For more information, see the respective user guides as follows:

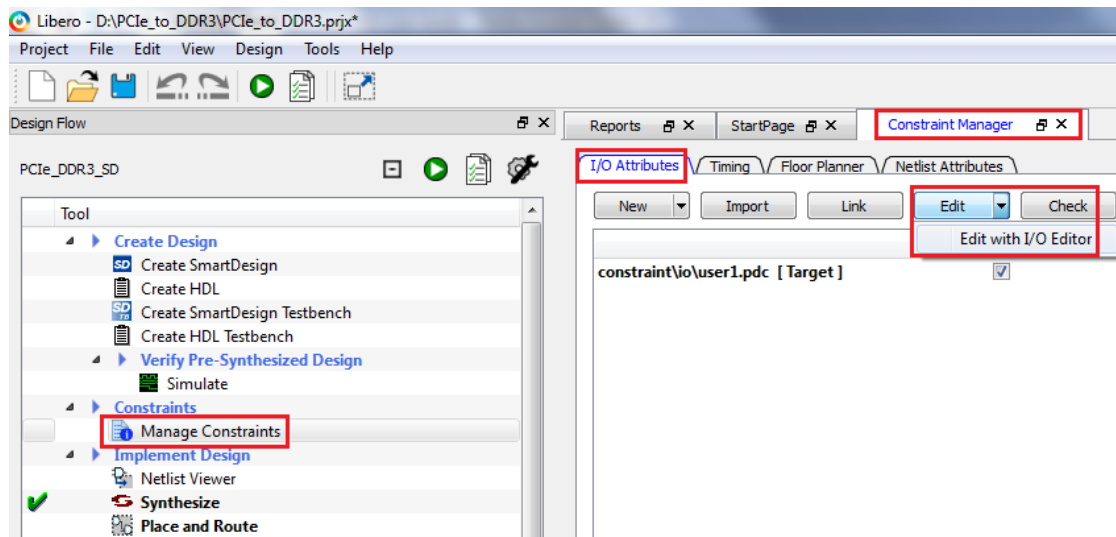
- [UG0722: PolarFire FPGA Packaging and Pin Descriptions User Guide](#)
- [UG0902: PolarFire SoC FPGA Packaging and Pin Descriptions User Guide](#)

#### 2.9.6.2.2. Using I/O Editor [\(Ask a Question\)](#)

To select the DDR subsystem location, follow these steps:

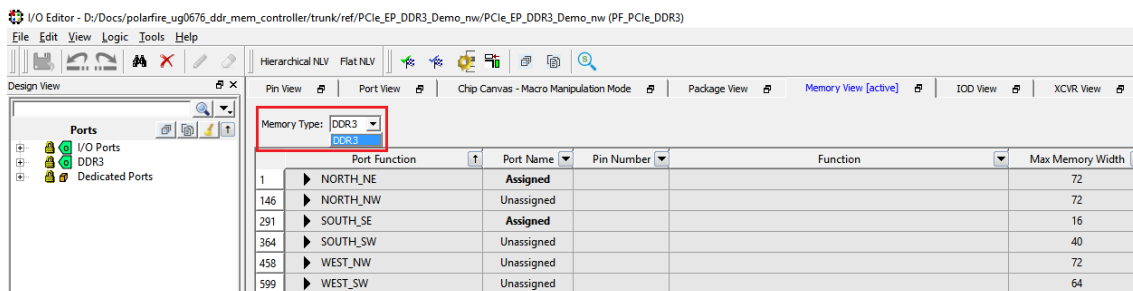
1. Click **Design Flow > Constraint Manager > I/O Attributes > Edit with I/O Editor**.

Figure 2.71. I/O Editor



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

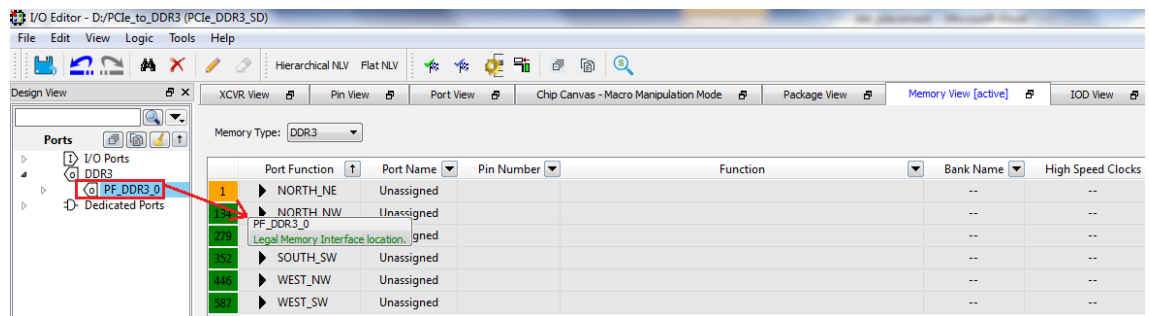
Figure 2.72. I/O Editor—Memory View



**➔ Important:** 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.

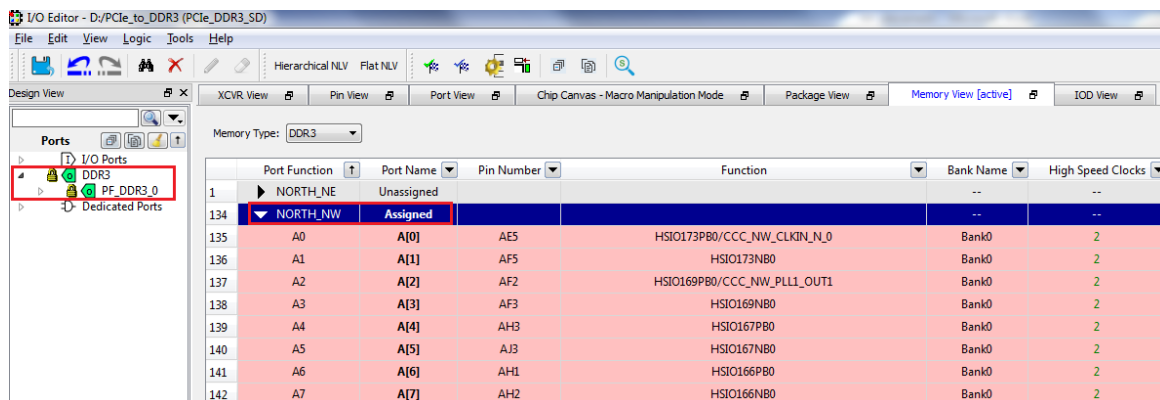
3. Select the appropriate DDR instance from the **Design View** window, and drag it onto **Port Function**. For the selected DDR instance, the I/O Editor shows the valid locations in green.

Figure 2.73. I/O Editor—Valid DDR Locations



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 2.74. I/O Editor—Assigned DDR Location



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.

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

Figure 2.75. Setting ODT

Port Name	Flash/Freeze mode	Clamp Diode	Resistor Pull	Schmitt Trigger	Vcm Input Range	On Die Termination	Qdt Value (Ohm)
DM[0]	AST_VALUE	ON	--	--	--	--	--
DM[1]	AST_VALUE	ON	--	--	--	--	--
DQ	AST_VALUE	ON	None	OFF	MID	ON	40
DQ[0]	AST_VALUE	ON	None	OFF	MID	OFF	40
DQ[1]	AST_VALUE	ON	None	OFF	MID	10T	40
DQ[2]	AST_VALUE	ON	None	OFF	MID	ON	40
DQ[3]	AST_VALUE	ON	None	OFF	MID	ON	40
DQ[4]	AST_VALUE	ON	None	OFF	MID	ON	40
DQ[5]	AST_VALUE	ON	None	OFF	MID	ON	40
DQ[6]	AST_VALUE	ON	None	OFF	MID	ON	40
DQ[7]	AST_VALUE	ON	None	OFF	MID	ON	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)

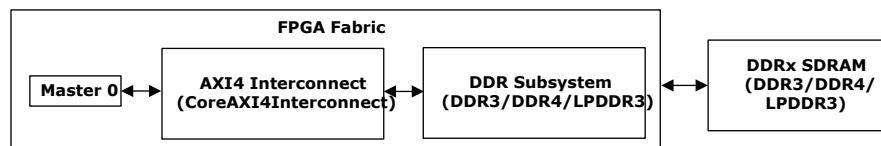
## 2.10. Functional Examples [\(Ask a Question\)](#)

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

### 2.10.1. Accessing DDR Subsystem Through AXI4 Interface [\(Ask a Question\)](#)

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 2.76. 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 [Implementation](#), or apply the preset configuration (if any), as shown in [Figure 2.49](#). 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 2.77.** CoreAXI4Interconnect IP Configurator

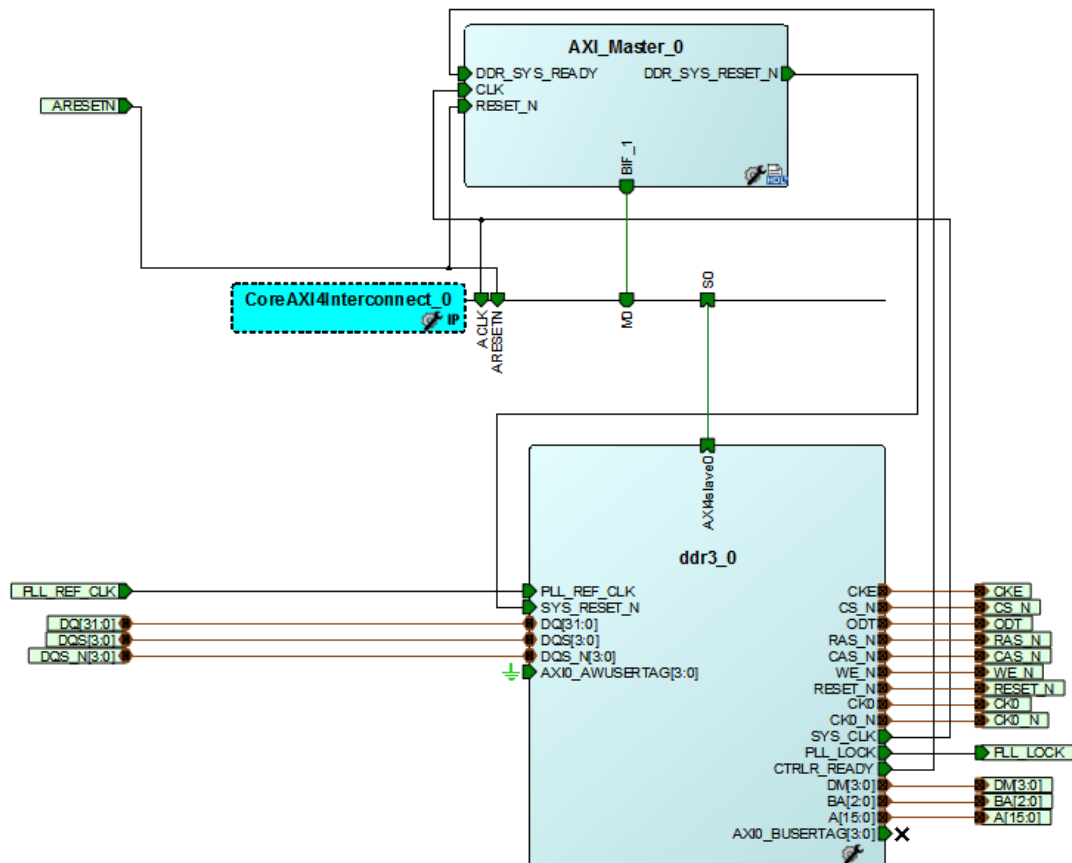
The screenshot shows the CoreAXI4Interconnect IP Configurator dialog box. It is titled "Configuration" and contains the following sections and parameters:

- 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:

At the bottom of the dialog, there is a "Help" dropdown menu, and "OK" and "Cancel" buttons.

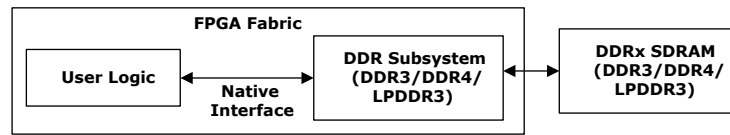
5. The AXI master and CoreAXI4Interconnect IP clocks must be driven from the SYS\_CLK clock of the DDR3 subsystem.
6. In the SmartDesign canvas, connect the blocks as shown in the following figure.

Figure 2.78. 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.

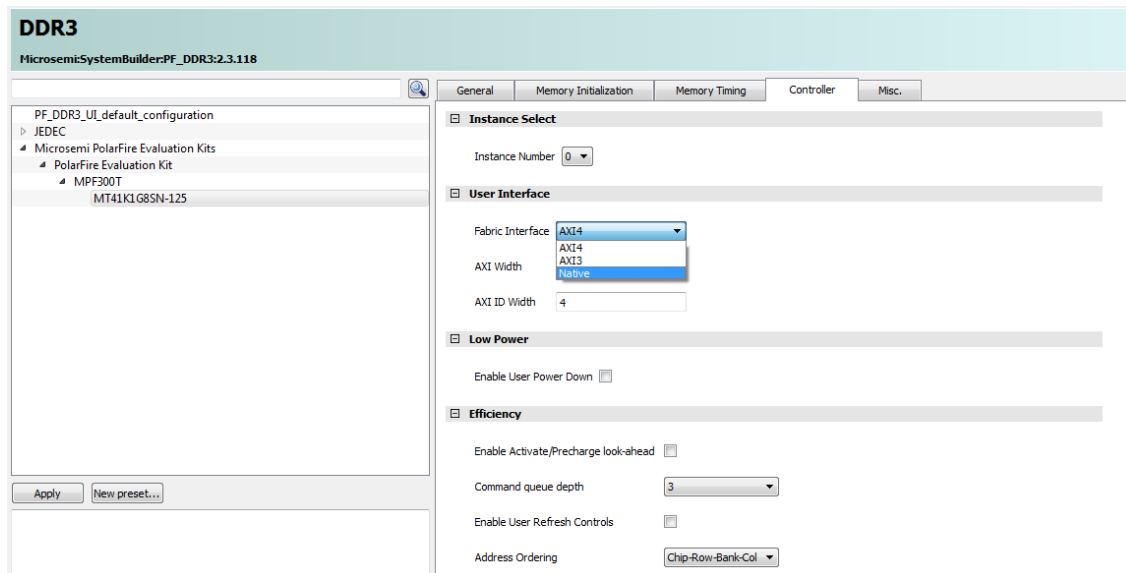


**Figure 2.81.** DDR Subsystem Accessed Through Native Interface

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

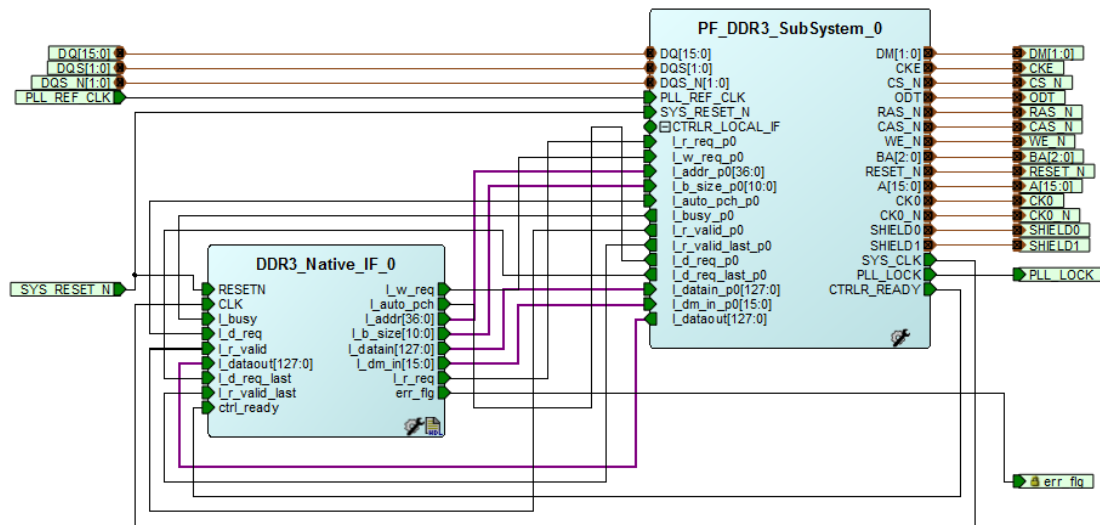
After successful DDR initialization, the native interface master initiates reads from or writes to the DDR memory. The following steps describe how to create a design to access the DDR3 memory from the native interface master in the FPGA fabric:

1. Create a SmartDesign, and instantiate the DDR3 macro.
2. Configure the DDR3 subsystem as described in [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 2.82.** Native Interface Selection

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 2.83. SmartDesign Connection—Native Interface



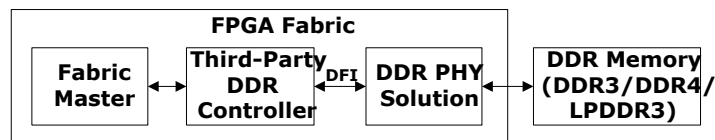
- Follow steps 7–11 of [Accessing DDR Subsystem Through AXI4 Interface](#) to simulate the design and see the read and write transactions.

### 2.10.3. Accessing DDR Memory Using a Third-Party DDR Controller and the DDR PHY-Only

#### Solution [\(Ask a Question\)](#)

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 2.84. DDR Memory Accessed Through Third-Party DDR Controller



### 3. Initialization Sequence

After the asynchronous system reset (SYS\_RESET\_N) is deasserted and PLL\_LOCK is asserted, 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.

#### 3.1. DDR3 [\(Ask a Question\)](#)

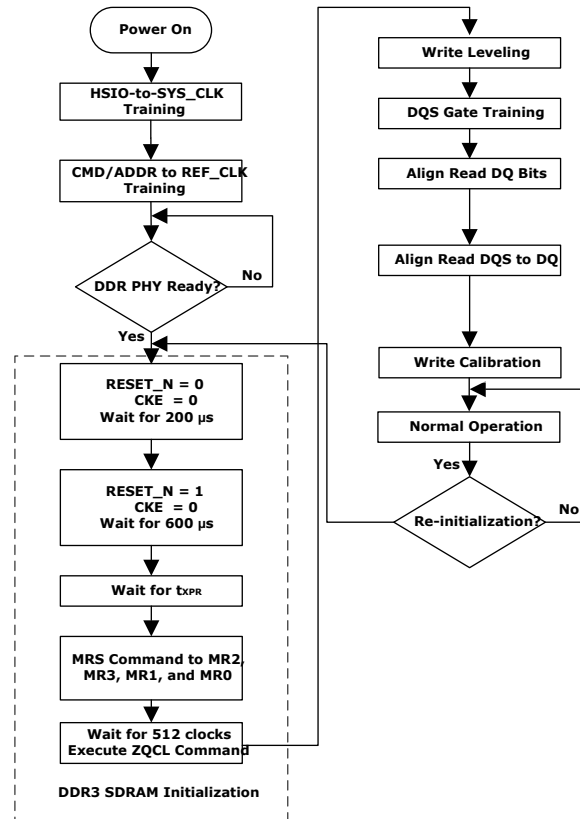
Automatic initialization of DDR3 memory involves the following steps:

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  $\mu$ s, and then deasserts CKE after an additional 600  $\mu$ s.
6. Controller waits for tXPR.
7. The MR2, MR3, MR1, and MR0 registers are configured (in that order) using the MRS command.
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 3.1. Automatic Initialization Flow for DDR3 Memory



**Note:** If the CTRLR\_READY signal is not asserted, find the state of the failure using the **DDR Debug** option of SmartDebug.

### 3.2. DDR4 [Ask a Question](#)

Automatic initialization of DDR4 memory involves the following steps:

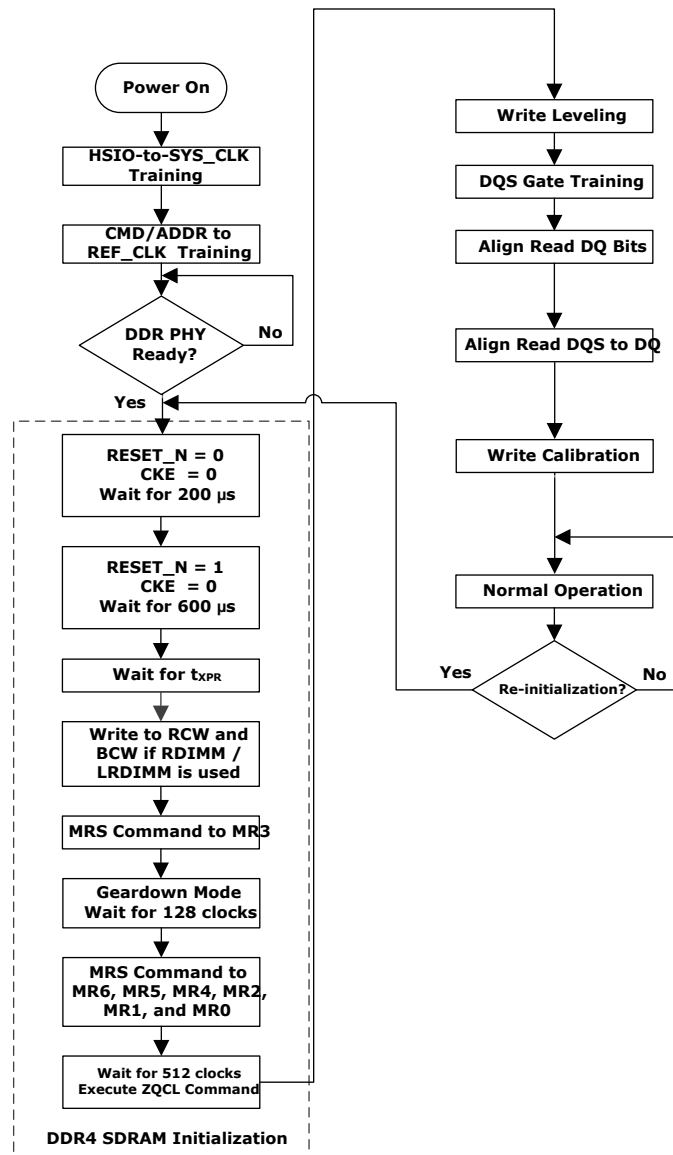
1. Training logic performs HS\_IO\_CLK-to-SYS\_CLK training.
2. Training logic performs CMD/ADDR to REF\_CLK training.
3. DDR controller waits for the PHY to be ready. At this point, it is assumed that the PHY outputs stable clocks and signal levels.
4. Controller asserts RESET\_N and CKE low.
5. Controller deasserts RESET\_N after 200  $\mu$ s, and then deasserts CKE after an additional 600  $\mu$ s.
6. Controller waits for tXPR.
7. Controller writes to RCW and BCW if RDIMM/LRDIMM is used.
8. The MR3, MR6, MR5, MR4, MR2, MR1, and MR0 registers are configured (in that order) using the MRS command.
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 3.2.** Automatic Initialization Flow for DDR4 Memory



### 3.3. LPDDR3 [\(Ask a Question\)](#)

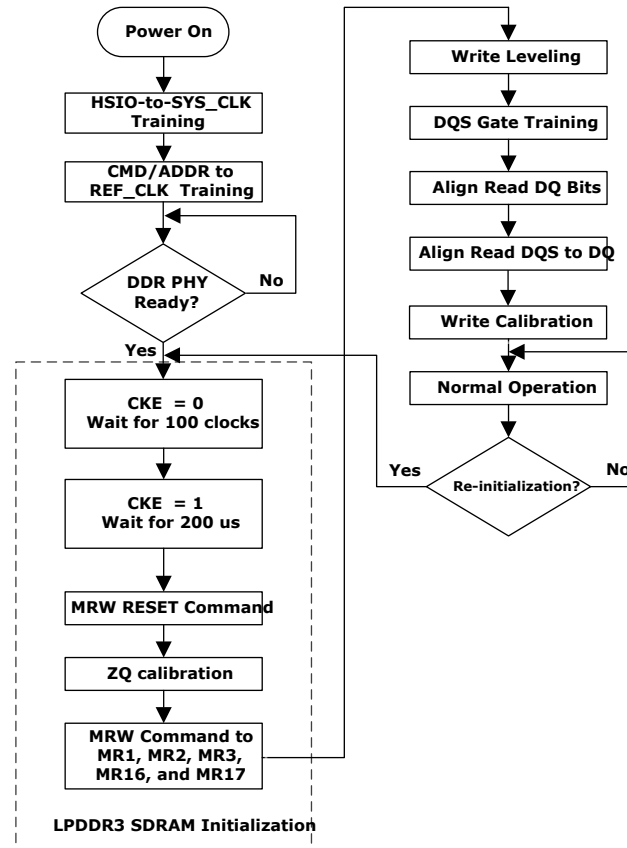
Automatic initialization of LPDDR3 memory involves the following steps:

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  $\mu$ s.
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. The MR1, MR2, MR3, MR16, and MR17 registers are configured (in that order) using the MRS command.
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 3.3. Automatic Initialization Flow for LPDDR3 Memory



### 3.4. LPDDR4 (For PolarFire SoC FPGA Only) [\(Ask a Question\)](#)

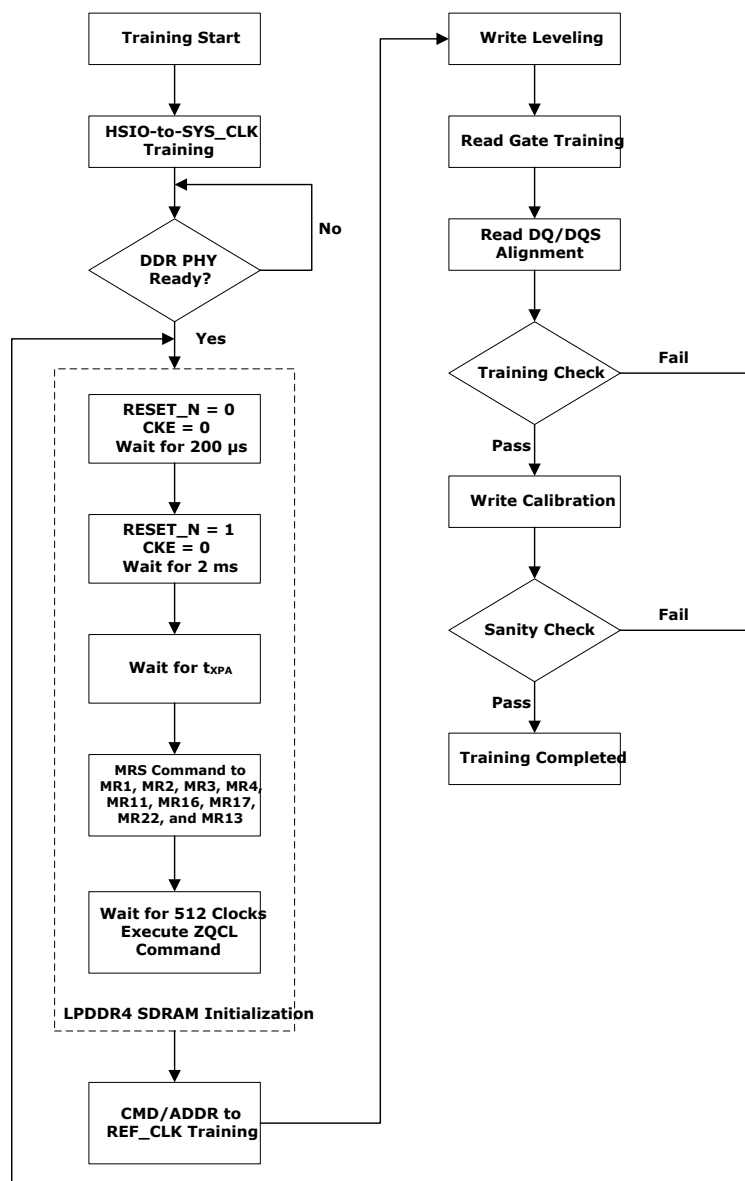
Automatic initialization of LPDDR4 memory (for embedded MSS DDR controller only) involves the following steps:

1. Training logic performs HS\_IO\_CLK-to-SYS\_CLK training.
2. DDR controller waits for the PHY to be ready. At this point, it is assumed that the PHY outputs stable clocks and signal levels.
3. Controller asserts RESET\_N, and CKE is held LOW.
4. Controller deasserts RESET\_N after 200  $\mu$ s, and then deasserts CKE after an additional 2 ms.
5. Controller waits for tXPR.
6. The MR13, MR1, MR2, MR3, MR4, MR11, MR16, MR17, MR22, and MR13 registers are configured (in that order) using the MRS command.
7. 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.
8. Training logic performs CMD/ADDR to REF\_CLK training.
9. Training logic performs write leveling.
10. Training logic performs Read Gate (Read DQS gate) training.

11. Training logic aligns read data (DQ) bits.
12. Training logic aligns read data strobe (DQS) to DQ.
13. Write calibration is performed.
14. Sanity check is performed.
15. Normal operation begins.

The following figure shows the LPDDR4 automatic initialization flow.

**Figure 3.4.** Automatic Initialization Flow for LPDDR4 Memory



## 4. Training Logic

The training logic manages the 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.
- CMD/ADDR to REF\_CLK training
  - Offset of the REF\_CLK to CMD/ADDR buses relies on internal silicon delays and can be affected by board signal integrity.
  - The user can include an additional phase offset of the signals through the memory configurator on the general tab using the CK/CA additive offset field. Each offset increment equates to 45 degrees.
  - The default setting of 0 was derived from test results using a DIMM based validation platform.
  - The user may need to adjust this parameter based on the specific board requirements. It is recommended to derive the accurate values of skew and termination based on simulations.

### Write Leveling

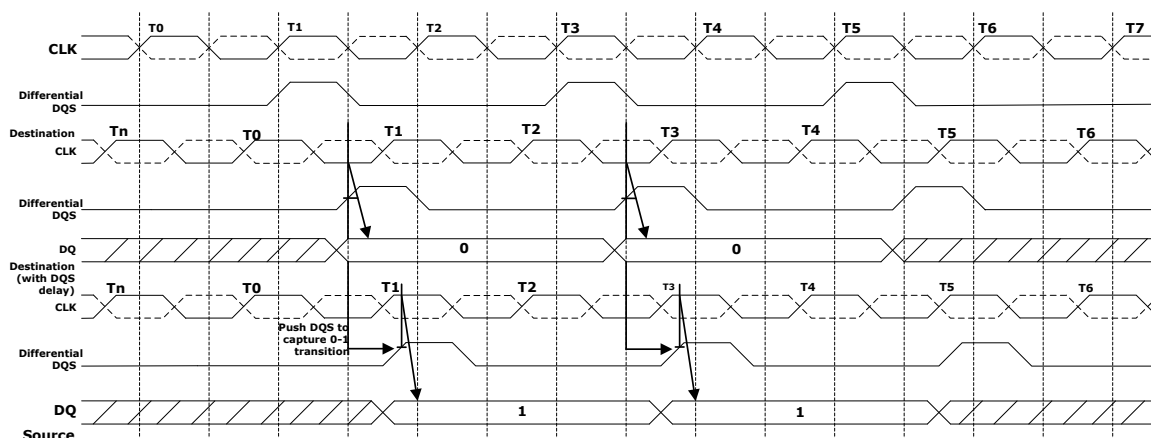
Write leveling is a training mode used during DRAM initialization. The write leveling process identifies the delay when the write DQS rising edge aligns with the rising edge of REF\_CLK. By identifying this delay, the system can accurately align the write DQS within REF\_CLK. When it is aligned, data 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  $\pm 0.25$  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 shown 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, REF\_CLK is aligned with DQS.

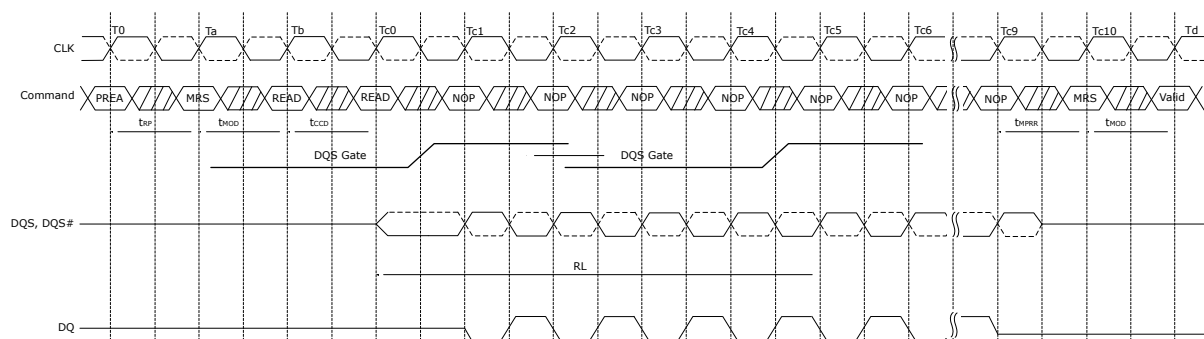
Figure 4.1. DQS Delay



### 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 shown 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 [PolarFire Board Design Recommendations](#) and [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 4.2. 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 show DQ/DQS before and after centering.

Figure 4.3. DQ/DQS Before Centering

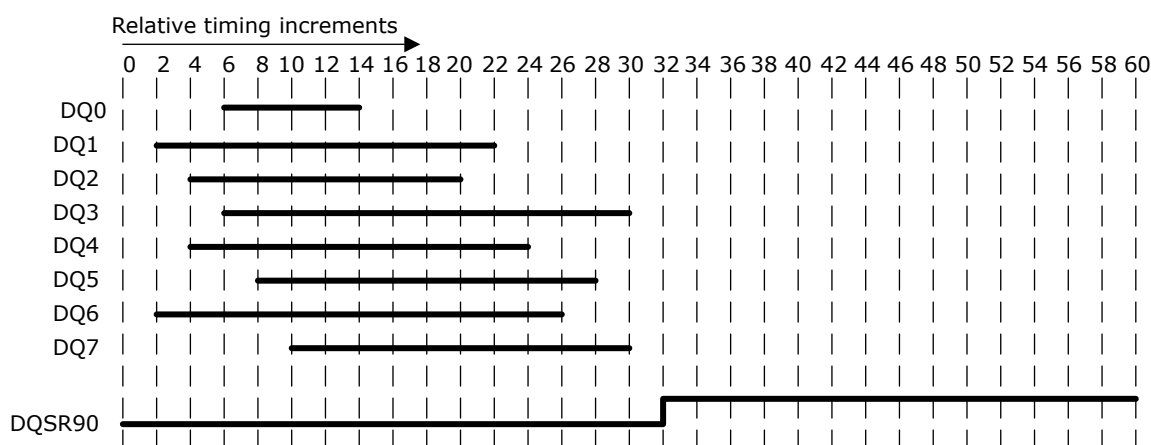
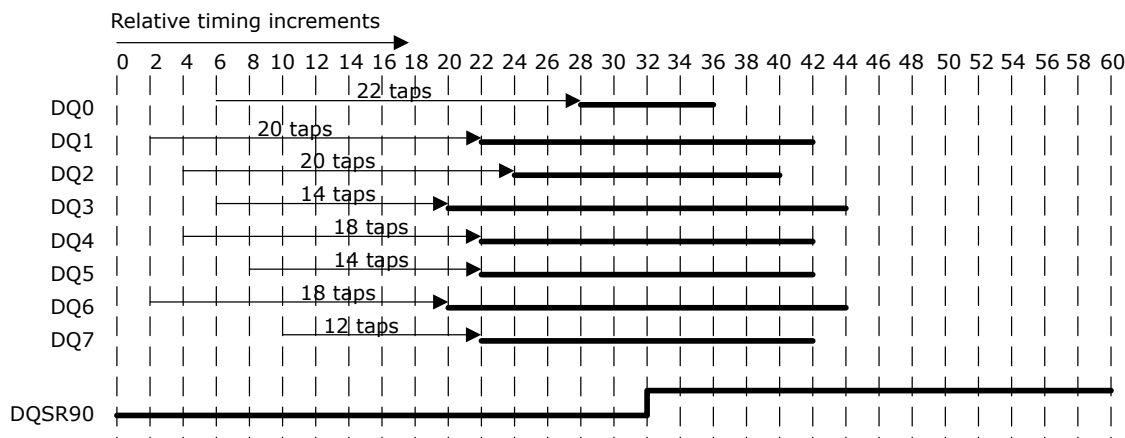


Figure 4.4. DQ/DQS After Centering



## 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 Family Power-Up and Resets User Guide](#).

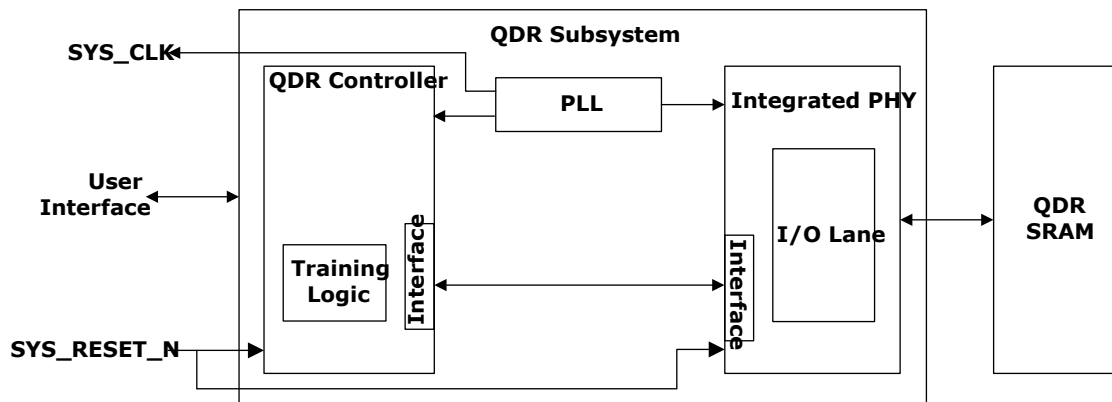
## 5. QDR Memory Controller

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 data 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 5.1. QDR Controller Block Diagram



### 5.1. Features [\(Ask a Question\)](#)

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, and 36 bits)
- Configurable address width (18–22 bits)

**➔ Important:** The 22-bit address width is supported only for 18-bit data width.

### 5.2. Performance [\(Ask a Question\)](#)

The following table lists the performance of QDR families.

**Table 5.1.** 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, or 144 Mb	36 or 72 Mb

### 5.2.1. Read Latency [\(Ask a Question\)](#)

The following table lists the QDR Read latency.

**Table 5.2.** 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

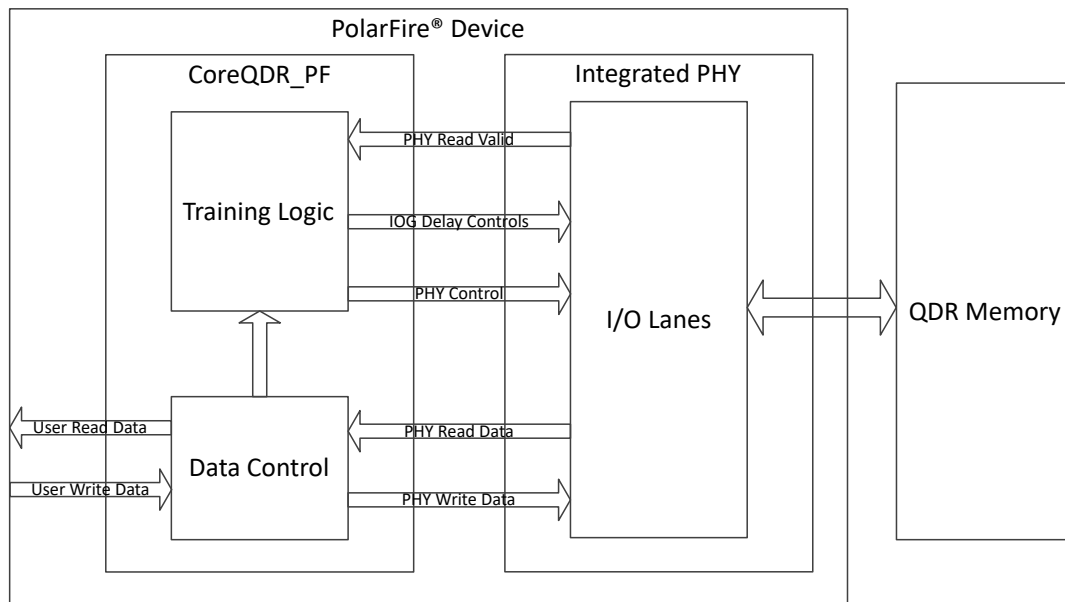
## 5.3. Functional Description [\(Ask a Question\)](#)

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

### 5.3.1. QDR Controller [\(Ask a Question\)](#)

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 and De-skews (due to board and internal delays) Input data, Input Clocks, and Input Valid Signals.

**Figure 5.2.** QDR Memory Controller

### 5.3.1.1. Integrated PHY [\(Ask a Question\)](#)

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.

#### 5.3.1.1.1. I/O Lane [\(Ask a Question\)](#)

An I/O Lane contains: Twelve I/Os, a Lane Controller, I/O Gearing Logic, and a set of High-speed and 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.



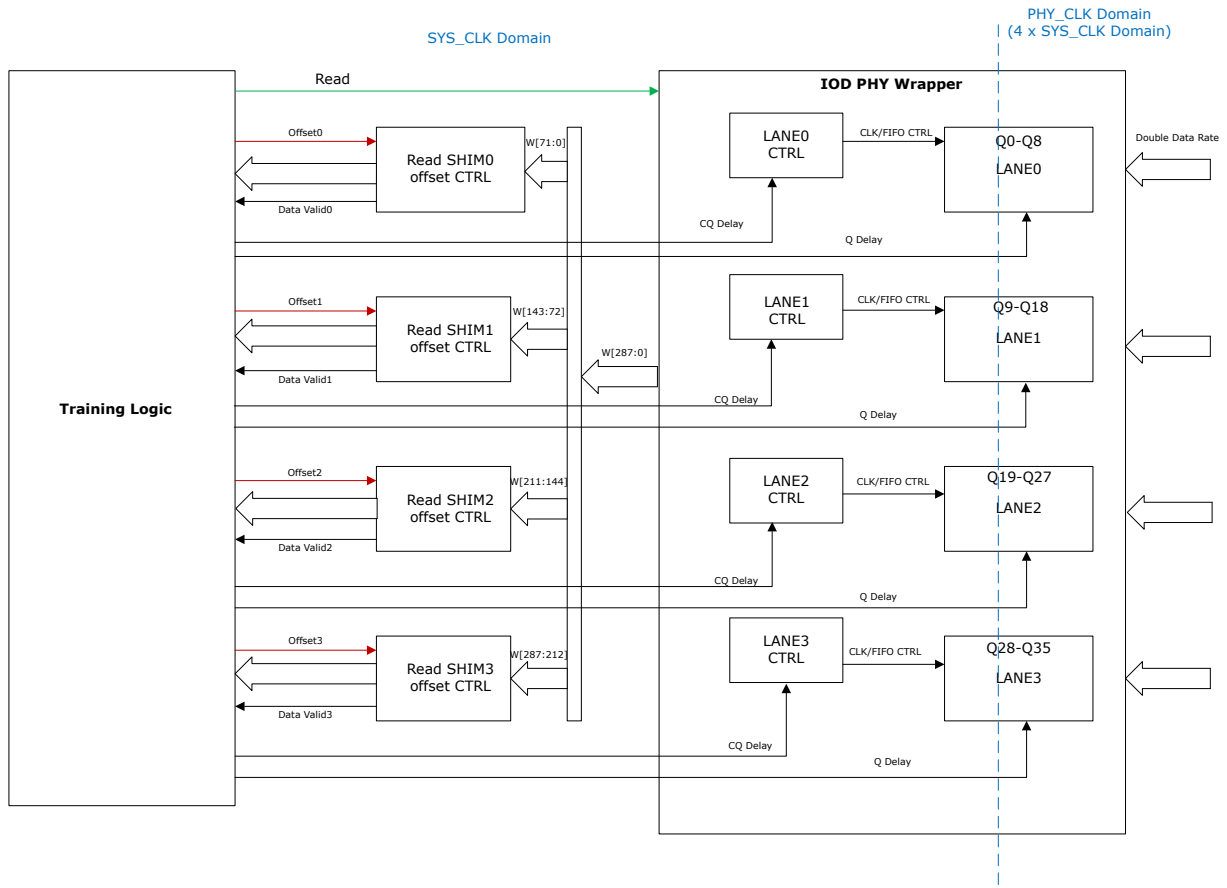
**Important:** Depending on the configuration of the QDR memory, few of the I/Os in Address/Command Lanes are used. Some I/Os in these Lanes can be reused as normal I/Os. For Data Lanes, all 12 I/Os in the Lane are used.

### 5.3.1.2. Training Logic [\(Ask a Question\)](#)

The training logic is implemented as a soft FPGA core that de-skews read data and center aligns the clock using IOD capabilities in the read direction. A known training pattern, generated by the QDR controller, is used for training.

The IOD PHY receives the read data and performs a serial to parallel conversion. In the following figure, the receive path for the QDR data width = 36 case is shown in detail. The IOD PHY collects the bytes for a total of 72 bits received per lane. The bytes are then passed to the respective SHIM block. The training logic shifts the pattern by full bit times in the SHIM if required. This is called data framing and it is performed iteratively until the received pattern matches the expected pattern. The training logic compares the bytes to the expected pattern and adjusts delay taps in the IOD PHY to center align the data ("Q") around memory clock CQP. This is called Q alignment. This operation is performed sequentially for the entire 288-bit word (4 lanes x 72 bits) byte by byte (lane by lane). The following figure shows the receive path.

Figure 5.3. Receive Path



**Note:** The above example is applicable for DQ Width=36 bits. For DQ Width=18 bits, two I/O Lanes are used. For DQ Width =9 bits, one I/O Lanes is used.

**➔ Important:** QDR training is performed on receive path only. Following QDR vendor-specific PCB guidelines ensures proper write operations.

The QDR RAM uses the CQP and CQN clock pair to capture data in the PHY. IODs in the PHY only use CQP for sampling data. The Lane Controller adjusts delay taps to center CQP within the valid data eye. This tuning occurs during CQ alignment.

### Training Pattern

- Q alignment– Implemented to adjust the delay taps of the serial bits received. A pseudo random generator is used with a standard PRBS8 sequence having a period of 256 words.
  - Math formula is  $x^8 + x^6 + x^5 + x^4 + 1$ .
  - Seed value is 0xAA.
- CQ Alignment– Used to center align the clock with respect to the received data. Training pattern is fixed to 0xA5.

### Training FSM

The FSM is designed to iterate across lanes and across each IOD data pad. Data framing, Q Alignment, CQ Alignment operations are all performed using the same methodology. A sequence is written into memory and read back; received data is compared with the transmitted data. Based on this comparison, some actions are taken. What differentiates the three phases is the block where these tuning actions are performed.

- Data framing- It is used to identify the beginning of the data frame. There is a data valid signal that is a delayed version of the read command and is used as a gate to sample the data. In [Figure 5.3](#), it applies to Q0 (Lane0), Q9 (Lane 1), Q18 (Lane2), and Q27 (Lane4). The CQP clock is used for sampling the data, looking for a match between write and read sequences. All the IOD delay taps are tried as it searches for a match. If there is no match, the read sequence and/or the data valid signal are shifted by an integer number of bits using the shift register in the read SHIM layer. This process continues until there is a match, and by definition, data valid is now framing the data.
- Q alignment- It is used to find the delay tap setting for each bit that results in the widest data valid window across the data byte. Only the bits sampled on the falling edge are used to compare the written and read data.
- CQ alignment- After the delay taps for all IODs are set in a lane during Q alignment, then CQ alignment is performed. This step fine tunes the delay taps of the CQP. As we only use CQP to capture the data, we only adjust the delay taps on the CQP IOD. Only the bits sampled on the rising edge are used to compare written and read data.

The algorithm pseudo code of the whole process (data framing, Q alignment, CQ alignment) is as follows:

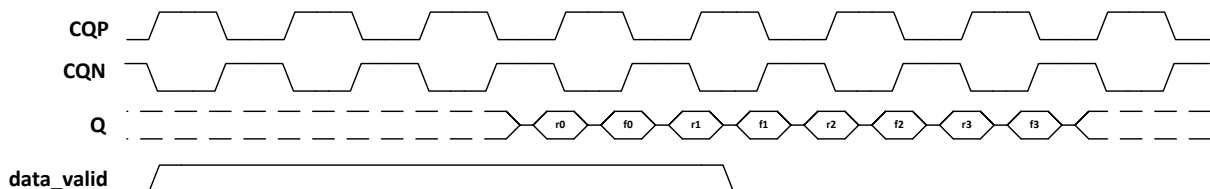
```

for k=1:num_lanes
  for g=1:num_IODs_in_lane(9)
    for j=1:delay_taps(256)
      for i=1:byte_len(8)
        // 1. Write a test pattern to memory generated from the PRBS8 generator.
        // 2. Read it back and compare it with bits 7,5,3,1 (falling edge bits)
        //    in the byte to the corresponding bits of the test pattern.
        // 3. Generate the next pattern from the PRBS8 generator.
      END for // byte_len
      // 1. Compare the eight generated patterns.
      // 2. Increment IOD delay taps as the training logic searches for all 8 patterns
      //    to compare.
      // 3. Once found, it indicates the start of the valid window, and the delay
      //    value (delay left) is recorded.
      // 4. Continue to increment the delay taps as the patterns match. When 7(*8 patterns)
      //    consecutive matches and there is no longer a match, save this second delay value
      //    (delay right) calculate mid value (left+right)/2 and exit loop.
    END for //delay_taps
    // 1. After trying all delay values (or exited the loop):
    // 2. Once a solution is found, the IOD delay tap is set to the midpoint of the
    //    valid window.
    // 3. If (a solution is not found on the first IOD of the lane), increase the offset
    //    (see data framing). Reload default delay value and set j=1 and retry to delay.
    // 4. If out of range and there is no solution. Increase fail number and try again
    //    from beginning. If number of failures is greater than 15, declare ERROR.
  END for //num_IODs_in_lane
  Perform CQ alignment
END for //num_lanes

```

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

**Figure 5.4. Data Capture Before Training Begins**

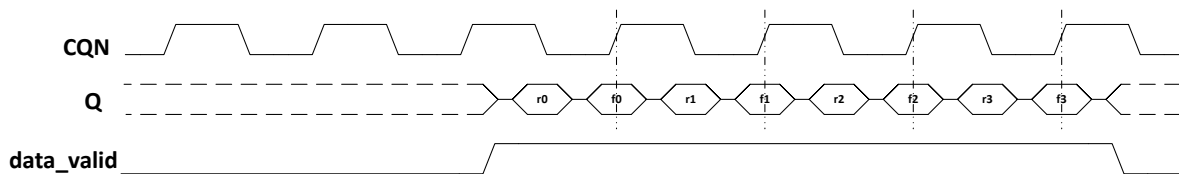


At first, the falling edge data alignment is done concurrently with the data\_valid alignment. As a result, the data\_valid properly frames the 8-bit burst coming from the QDR device and the CQN centers the falling edge data (f0, f1, f2, f3).

**➔ Important:** 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>.

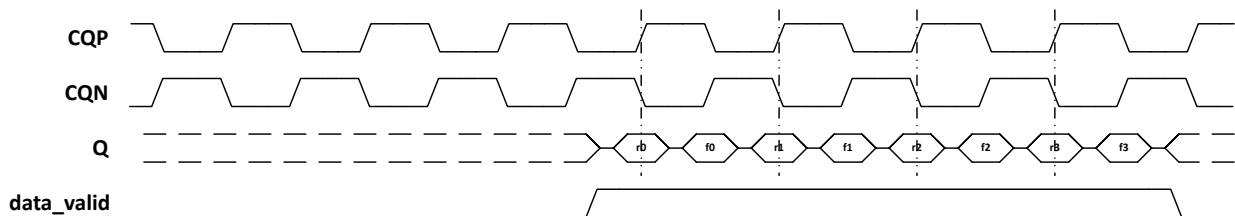
A high-level timing diagram of alignment is shown in the following figure.

**Figure 5.5.** 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 5.6.** Data Capture After Moving CQP to Center Rising Edge Data

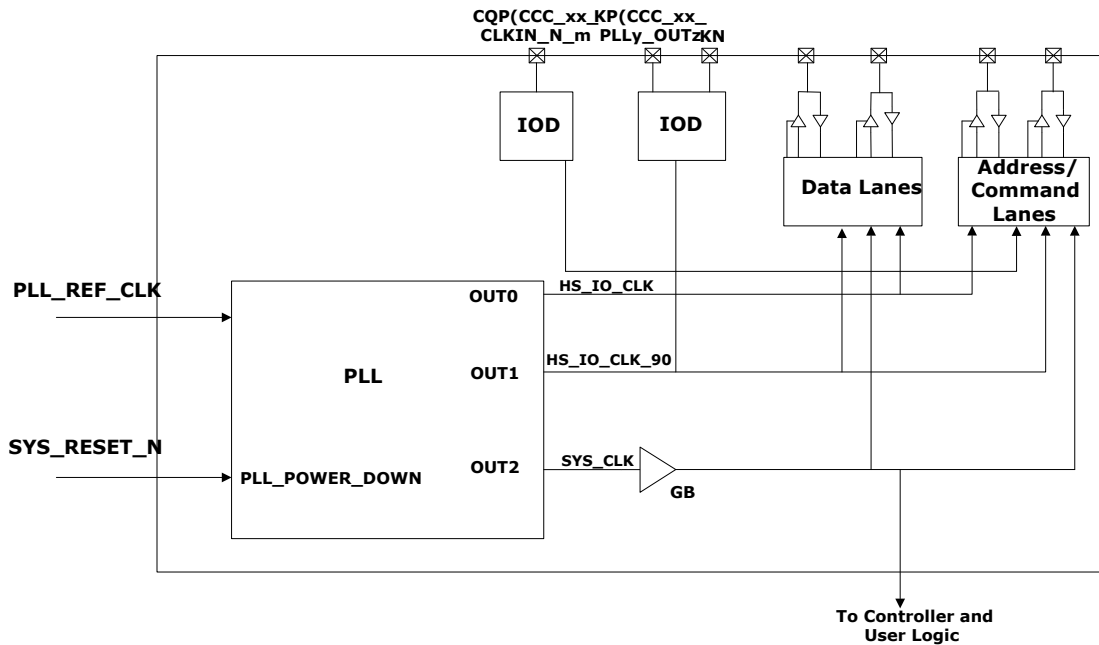


### 5.3.2. Clocking Structure [\(Ask a Question\)](#)

The QDR subsystem requires 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 5.7. 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.

### 5.3.3. Port List [\(Ask a Question\)](#)

The following table lists the QDR ports.

Table 5.3. QDR Signals and Descriptions<sup>1</sup>

Name	Width	Direction	Description
<b>Generic Signals</b>			
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. <b>Note:</b> The RESET_N signal must be generated by ANDing the DEVICE_INIT_DONE and BANK_X_CALIB_STATUS signals of the INIT_MONITOR IP core. BANK_X refers to the BANK where the QDR controller is placed.
PLL_REF_CLK		Input	Reference clock to the PLL.
PLL_LOCK	1	Output	Lock signal. This signal is 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.

**Table 5.3. QDR Signals and Descriptions<sup>1</sup> (continued)**

Name	Width	Direction	Description
TRAINING_ERROR[2:0]	3	Output	<ul style="list-style-type: none"> <li>- Bit 0: Reserved</li> <li>- Bits [2:1] <ul style="list-style-type: none"> <li>• 0: No error.</li> <li>• 1: Timeout error, data VALID is received during Q training.</li> <li>• 2: No solution found during training.</li> </ul> </li> </ul> <p>If no solution is found, restart the Training process described in <a href="#">Training Logic</a>.</p>
<b>User Interface Signals</b>			
READ_N	4	Input	<p>Read request with address specified on RADDR bus.</p> <ul style="list-style-type: none"> <li>• 4-bit Wide if Burst Size is 2.</li> <li>• 2-bit Wide if Burst Size is 4.</li> </ul>
WRITE_N	4	Input	<p>Write request with address specified on WADDR bus.</p> <ul style="list-style-type: none"> <li>• 4-bit Wide if Burst Size is 2.</li> <li>• 2-bit Wide if Burst Size is 4.</li> </ul>
WSTRB_N	SRAM_DWIDTH/9 * 8	Input	Write strobe for write transaction.
WADDR	SRAM_AWIDTH*4	Input	<p>Write addresses for each beat.</p> <p>Width is (SRAM_AWIDTH)*4 if burst size is 2, (SRAM_AWIDTH)*2 if burst size is 4</p> <ul style="list-style-type: none"> <li>• In Burst of 2: {ADDR3, ADDR2, ADDR1, ADDR0}</li> <li>• In Burst of 4: {ADDR1, ADDR0}</li> </ul>
RADDR	SRAM_AWIDTH*4	Input	<p>Read addresses for each beat.</p> <p>Width is (SRAM_AWIDTH)*4 if burst size is 2, (SRAM_AWIDTH)*2 if burst size is 4</p> <ul style="list-style-type: none"> <li>• In Burst of 2: {ADDR3, ADDR2, ADDR1, ADDR0}</li> <li>• In Burst of 4: {ADDR1, ADDR0}</li> </ul>
WDATA	(SRAM_DWIDTH)*8	Input	Write data
RDATA	(SRAM_DWIDTH)*8	Output	Read data, valid when RVALID is asserted
RVALID	1	Output	Read data valid
<b>QDR SRAM Interface</b>			
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] <sup>2</sup>	18, 19, 20, 21, or 22 (X=17,18,19, 20, or 21)	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
KN	1	Output	Negative Clock Output
CQP	1	Input	Synchronous echo Clock input

**Table 5.3.** QDR Signals and Descriptions<sup>1</sup> (continued)

Name	Width	Direction	Description
DOFF_N	1	Output	Initially, DOFF_N is 0. After calibration, DOFF_N is set to 1. If <b>Expose DOFF_N port</b> option is selected in the configurator, the DOFF_N signal is exposed and taken care by the QDR controller. If <b>Expose DOFF_N port</b> option is not selected in the configurator, the DOFF_N signal does not get exposed and must be taken care on the board using a pull-up resistor (check the memory model data sheet).

**Notes:**

1. In the QDR configurator, the **QDR data width** field is same as SRAM\_DWIDTH. In the QDR configurator, the **address width** field is same as SRAM\_AWIDTH.
2. The 22-bit address width is supported only for the 18-bit data width.

**5.4. Functional Timing Diagrams** [\(Ask a Question\)](#)

This section describes the read/write rules and sequences in the User Interface. For information about User Interface signals, see [Port List](#).

**5.4.1. Write Strobe Mapping** [\(Ask a Question\)](#)

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 5.10](#) and [Figure 5.11](#). 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 5.4.** 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 5.5.** Write Data to Write Strobe Mapping (x36)

WSTRB_MAPPI NG_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

**Table 5.5.** Write Data to Write Strobe Mapping (x36) (continued)

WSTRB_MAPPING_INDEX	WDATA INDEX									RISING/ FALLING	CLOCK CYCLE
27	261	262	263	264	265	266	267	268	269	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
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

<sup>(1)</sup> 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 5.6.** 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

<sup>(1)</sup> 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 5.7.** 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

**Table 5.7.** Write Data to Write Strobe Mapping (x9) (continued)

WSTRB_MAPPING_INDEX	WDATA INDEX										RISING/ FALLING	CLOCK CYCLE
5	54	55	56	57	58	59	60	61	62	63	FALLING	2
6	27	28	29	30	31	32	33	34	35	36	RISING	3
7	63	64	65	66	67	68	69	70	71	72	FALLING	4

#### 5.4.1.1. Burst of 2 [\(Ask a Question\)](#)

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.

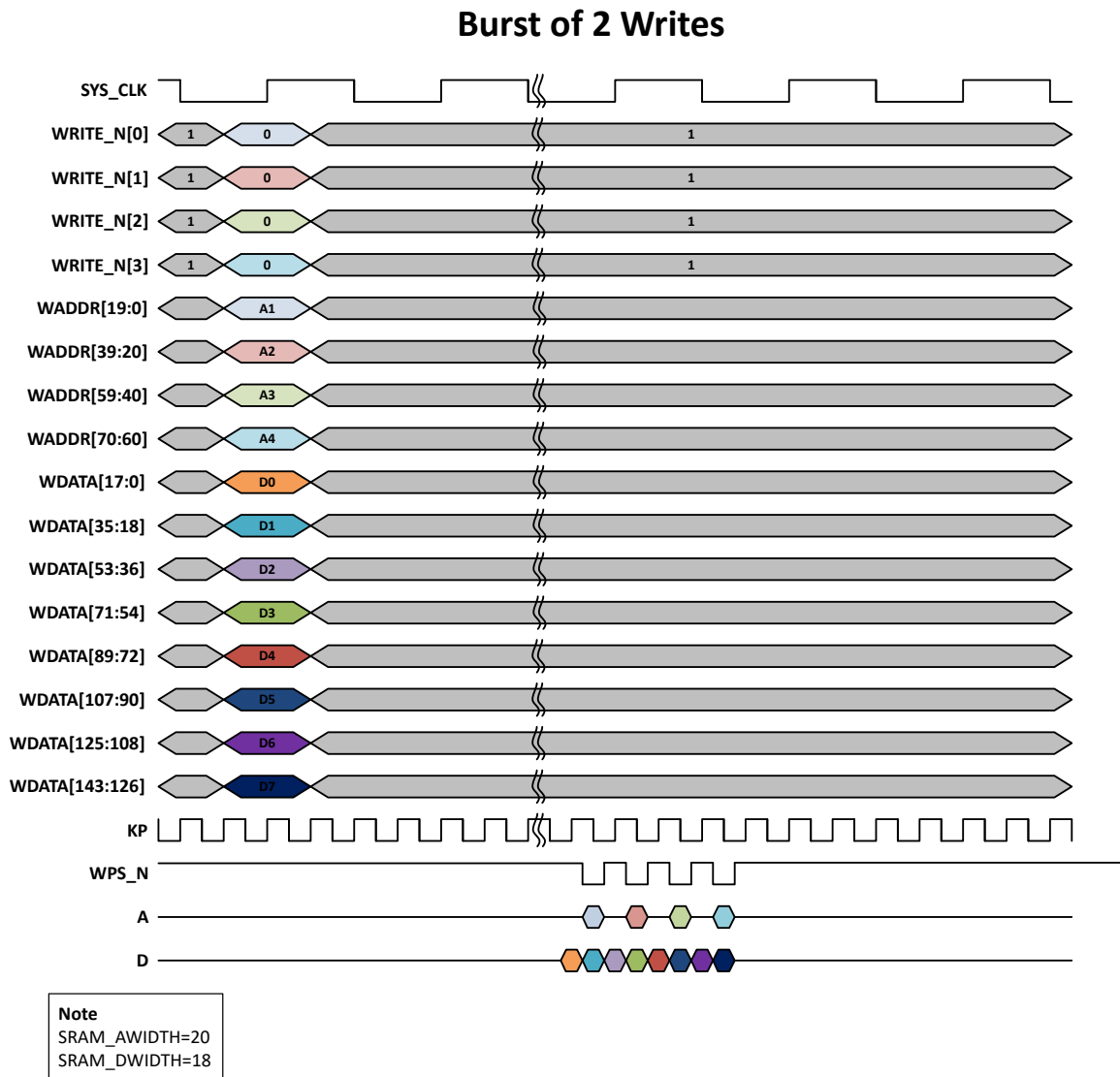
In burst of 2 mode, 4 write transactions (WRITE\_N[0,1,2,3]) are completed in one SYS\_CLK cycle. In each transaction data is 18-bit wide and address is 20-bit wide. The 4 write transactions must be triggered with respect to KP clock, which is derived from SYS\_CLK internally. The first write (when WRITE\_N[0] = 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 5.8. Burst of 2 Local Interface Timing



#### 5.4.1.2. Burst of 4 [\(Ask a Question\)](#)

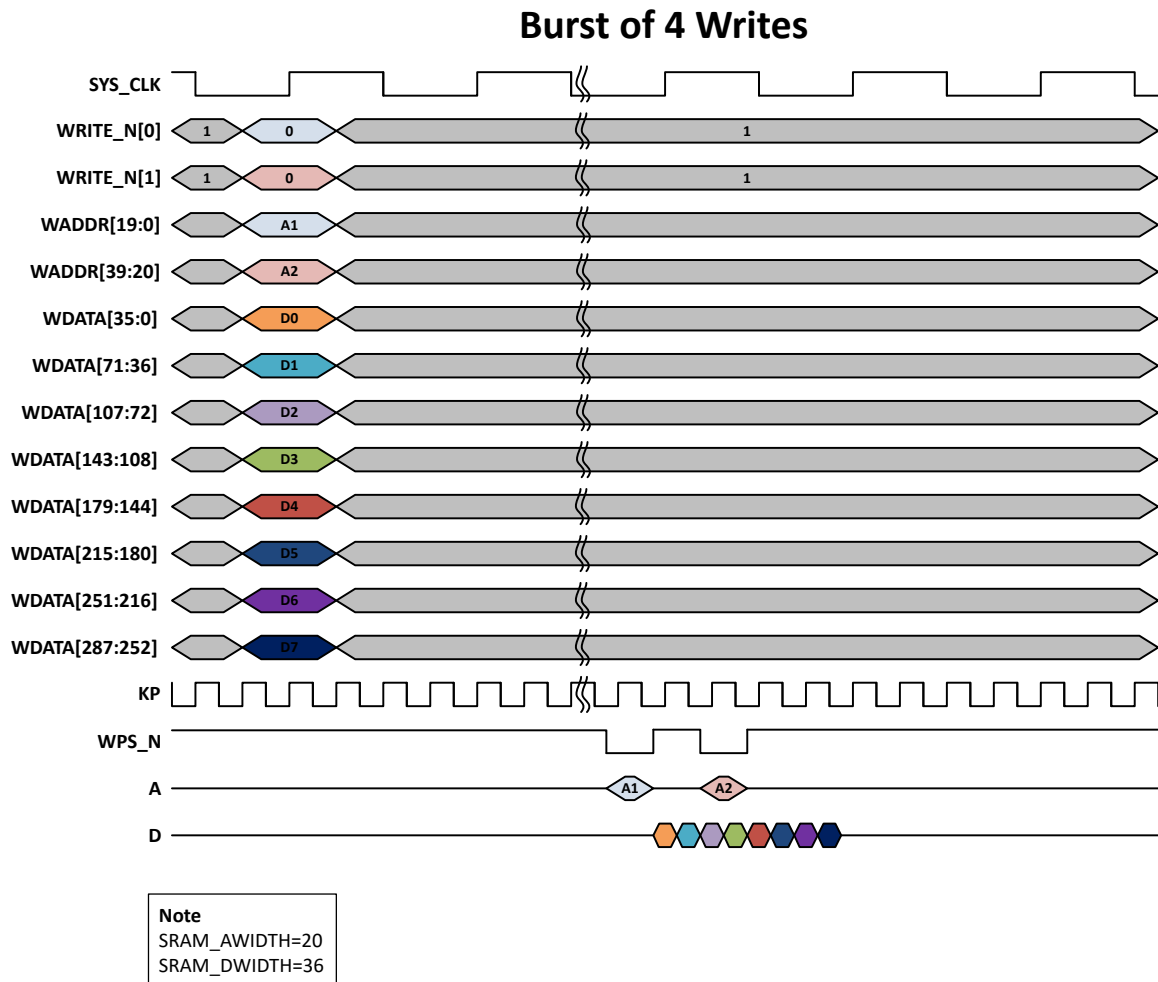
In the burst of 4 mode (SRAM\_BURST = 0), only half the WADDR/RADDR and WRITE\_N/READ\_N are utilized, because each SYS\_CLK cycle transaction corresponds to only 2 (instead of 4) K clock transactions. A sample write transaction in burst of 4 mode with SRAM\_AWIDTH=20 and SRAM\_DWIDTH=36 is shown in 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:
  - 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 5.9. Burst of 4 Local Interface Timing



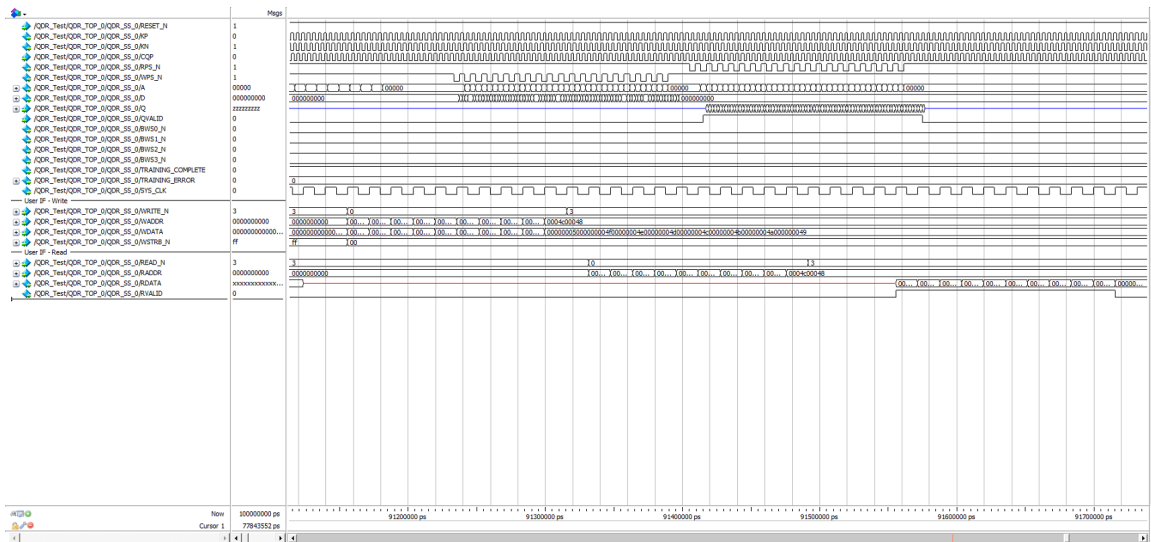
#### 5.4.2. QDR SRAM Write [\(Ask a Question\)](#)

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.

- Multiple Writes can be initiated by asserting the multiple WRITE\_N[x] signals to low. The address WADDR[SRAM\_AWIDTH\*(x+1) -1: 0] and DATA[SRAM\_DWIDTH\*(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].
- Maximum of 8 SRAM\_DWIDTH Bit writes to QDR SRAM are initiated in a single SYS\_CLK clock cycle.

Figure 5.10. Write in 4 Burst Mode



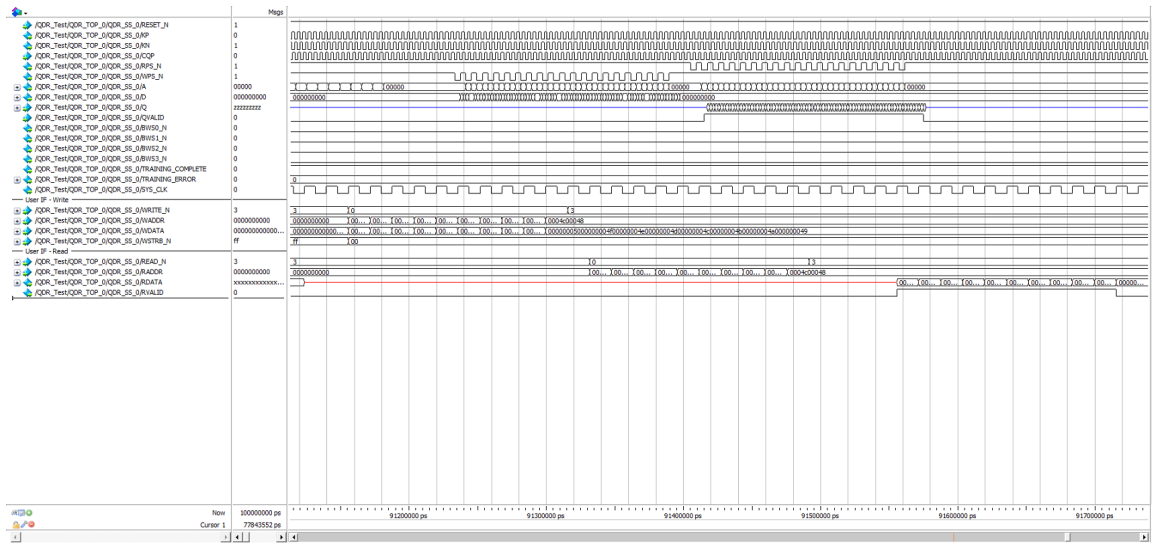
### 5.4.3. QDR SRAM Read [\(Ask a Question\)](#)

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:

- 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.
- Multiple Reads can be initiated by asserting the multiple READ\_N[x] signals to Low. The address RADDR[SRAM\_AWIDTH\*(x+1) -1 : 0] and RDATA[SRAM\_DWIDTH\*(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 5.11. Read in 4 Burst Mode



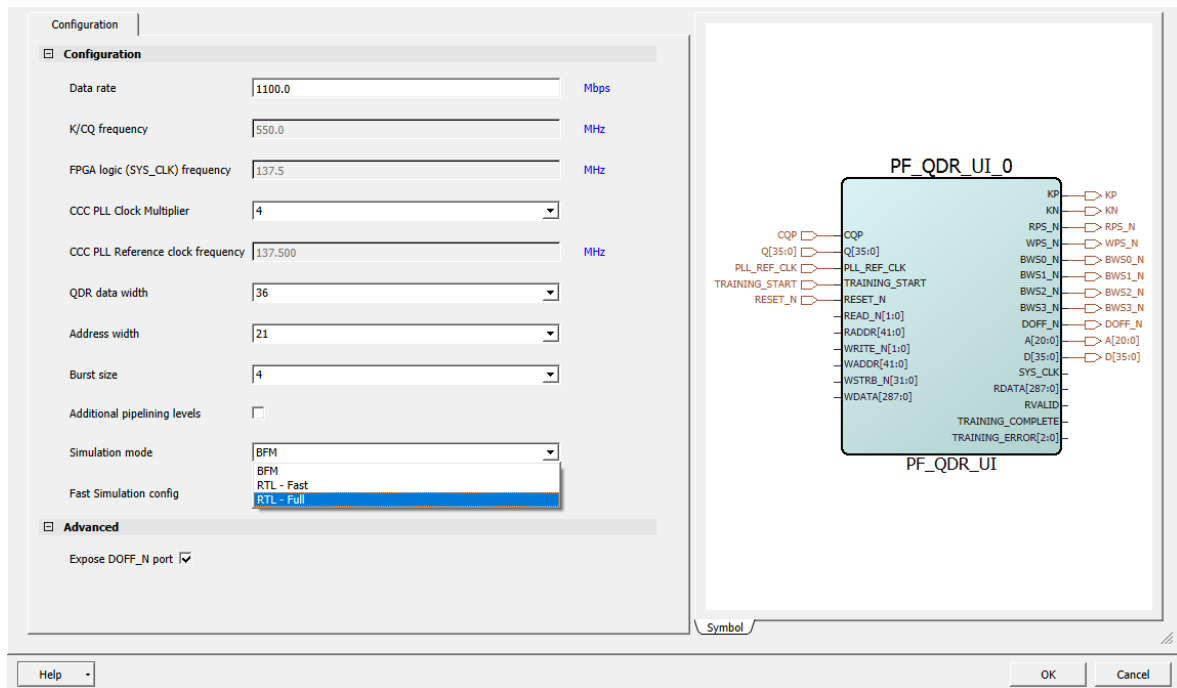
## 5.5. Implementation [\(Ask a Question\)](#)

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

### 5.5.1. Configuring QDR Memory Controller IP [\(Ask a Question\)](#)

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 5.12. QDR Configurator



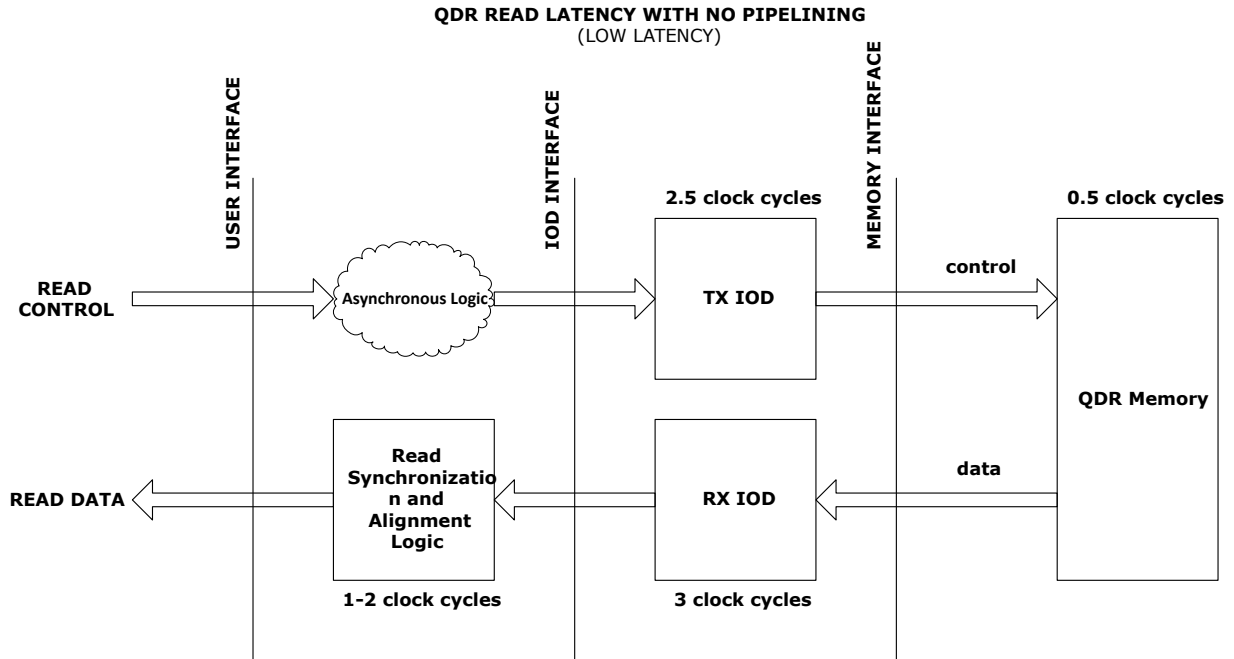
The following fields are available for configuring the QDR Subsystem:

- **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\_Fast:** In this mode, the QDR subsystem loads the training delays as per the selected configuration, skips the training sequence and asserts the TRAINING\_COMPLETE signal. QDR memory model is required for RTL\_Fast 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:** This option is used for static timing closure, if necessary. Enabling this option increases the read latency by 2 - 3 SCLK cycles.

#### 5.5.1.1. Pipelining Latency [\(Ask a Question\)](#)

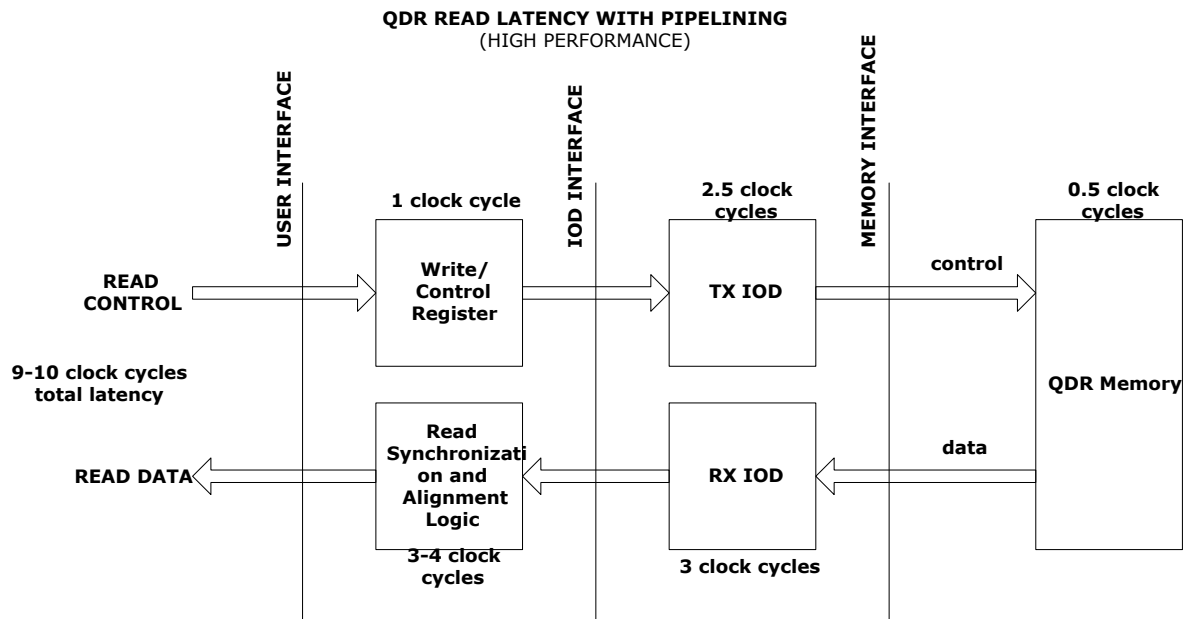
This section describes the latency with pipelining enabled and disabled. The following figure shows the QDR read latency with pipelining disabled.

Figure 5.13. Latency without Pipelining



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

Figure 5.14. Latency with Pipelining



### 5.5.2. Simulating QDR Memory Controller [\(Ask a Question\)](#)

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 [Functional Example](#).

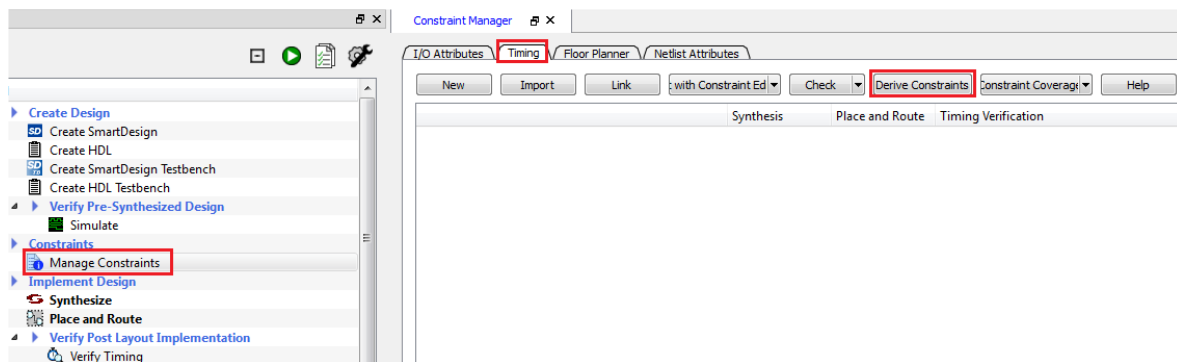
### 5.5.3. Design Constraints [\(Ask a Question\)](#)

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

#### 5.5.3.1. Timing Constraints [\(Ask a Question\)](#)

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 5.15.** Constraint Manager-Derive Constraints Tab



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

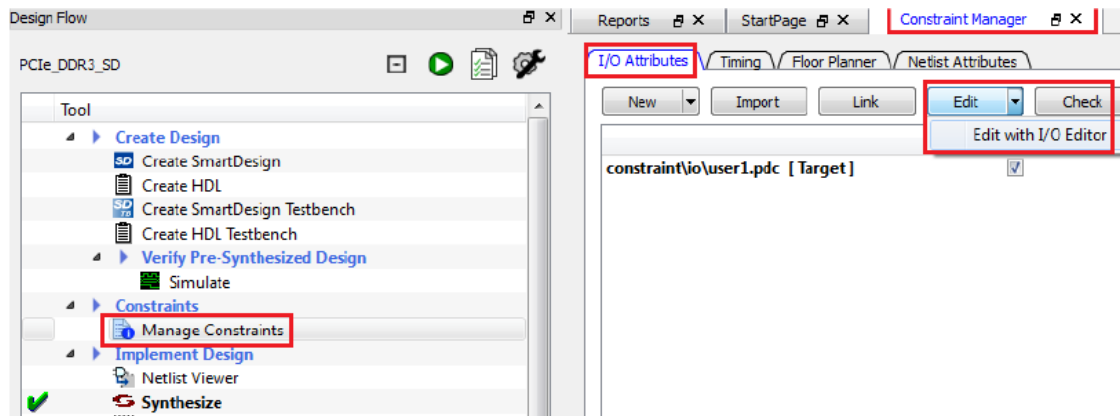
#### 5.5.3.2. Physical Constraints [\(Ask a Question\)](#)

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, PolarFire SoC, and RT PolarFire 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:

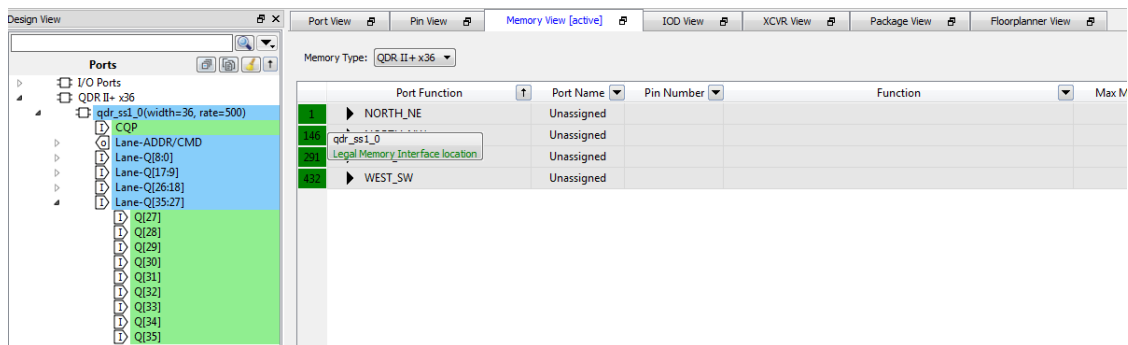
1. Click **Design Flow > Constraint Manager > I/O Attributes > Edit with I/O Editor**.

Figure 5.16. Design Flow Window: I/O Editor



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 5.17. Design View Window: Port Function



#### 5.5.4. QDR Resource Utilization [\(Ask a Question\)](#)

This section provides the post-Synthesis QDR resource utilization reports from Libero<sup>®</sup> SoC for the following QDR configurations.

- For data width = 18-bit, address width = 22-bit, burst size = 4, and Simulation mode = RTL—Fast, see [Figure 5.18](#).
- For data width = 18-bit, address width = 21-bit, burst size = 4, and Simulation mode = RTL—Full, see [Figure 5.19](#).
- For data width = 36-bit, address width = 21-bit, burst size = 4, and Simulation mode = RTL—Full, see [Figure 5.20](#).
- For data width = 9-bit, address width = 21-bit, burst size = 4, and Simulation mode = RTL—Full, see [Figure 5.21](#).
- For data width = 9-bit, address width = 21-bit, burst size = 4, and Simulation mode = BFM, see [Figure 5.22](#).

- For data width = 18-bit, address width = 21-bit, burst size = 4, and Simulation mode = BFM, see [Figure 5.23](#).
- For data width = 36-bit, address width = 21-bit, burst size = 4, and Simulation mode = BFM, see [Figure 5.24](#).

The following figure shows the QDR resource utilization report for 18-bit data width and RTL—Fast mode.

**Figure 5.18.** QDR Resource Utilization 1

Type	Used	Total	Percentage
4LUT	335	299544	0.11
DFF	90	299544	0.03
I/O Register	0	1536	0.00
User I/O	311	512	60.74
-- Single-ended I/O	309	512	60.35
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	2	48	4.17
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

The following figure shows the QDR resource utilization report for 18-bit data width and RTL—Full mode.

**Figure 5.19.** QDR Resource Utilization 2

Type	Used	Total	Percentage
4LUT	1562	299544	0.52
DFF	911	299544	0.30
I/O Register	0	1536	0.00
User I/O	466	512	91.02
-- Single-ended I/O	464	512	90.62
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	3	48	6.25
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

The following figure shows the QDR resource utilization report for 36-bit data width and RTL—Full mode.

**Figure 5.20.** QDR Resource Utilization 3

Type	Used	Total	Percentage
4LUT	2754	299544	0.92
DFF	1570	299544	0.52
I/O Register	0	1536	0.00
User I/O	488	512	95.31
-- Single-ended I/O	486	512	94.92
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	3	48	6.25
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

The following figure shows the QDR resource utilization report for 9-bit data width and RTL—Full mode.

**Figure 5.21.** QDR Resource Utilization 4

Type	Used	Total	Percentage
4LUT	926	299544	0.31
DFF	484	299544	0.16
I/O Register	0	1536	0.00
User I/O	295	512	57.62
-- Single-ended I/O	293	512	57.23
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	2	48	4.17
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

The following figure shows the QDR resource utilization report for 9-bit data width and BFM mode.

**Figure 5.22.** QDR Resource Utilization 5

Type	Used	Total	Percentage
4LUT	339	299544	0.11
DFF	128	299544	0.04
I/O Register	0	1536	0.00
User I/O	295	512	57.62
-- Single-ended I/O	293	512	57.23
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	2	48	4.17
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

The following figure shows the QDR resource utilization report for 18-bit data width and BFM mode.

**Figure 5.23.** QDR Resource Utilization 6

Type	Used	Total	Percentage
4LUT	491	299544	0.16
DFF	170	299544	0.06
I/O Register	0	1536	0.00
User I/O	466	512	91.02
-- Single-ended I/O	464	512	90.62
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	2	48	4.17
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

The following figure shows the QDR resource utilization report for 36-bit data width and BFM mode.

Figure 5.24. QDR Resource Utilization 7

Type	Used	Total	Percentage
4LUT	475	299544	0.16
DFF	95	299544	0.03
I/O Register	0	1536	0.00
User I/O	488	512	95.31
-- Single-ended I/O	486	512	94.92
-- Differential I/O Pairs	1	256	0.39
uSRAM	0	2772	0.00
LSRAM	0	952	0.00
Math	0	924	0.00
H-Chip Global	2	48	4.17
PLL	1	8	12.50
DLL	1	8	12.50
Transceiver Lanes	0	16	0.00
Transceiver PCIe	0	2	0.00
ICB_CLKINT	1	72	1.39

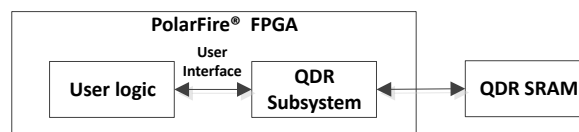
## 5.6. Functional Example [\(Ask a Question\)](#)

This section describes how to perform the RTL\_FULL, BFM, and RTL\_Fast simulations.

### 5.6.1. RTL\_FULL Simulation [\(Ask a Question\)](#)

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.

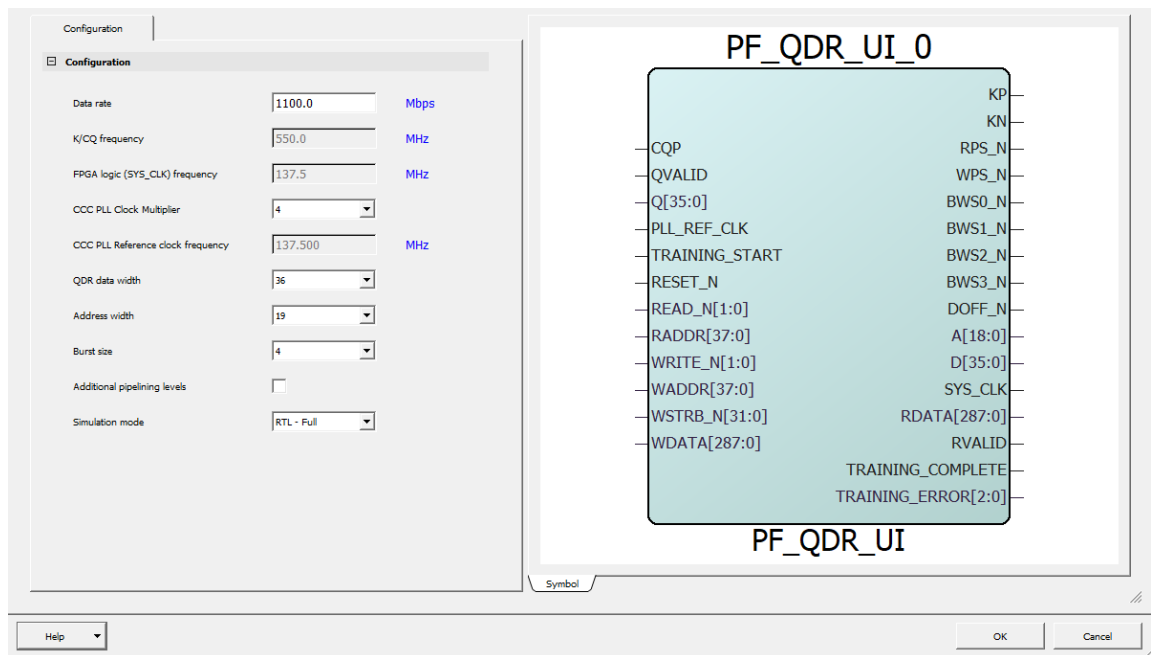
Figure 5.25. QDR Subsystem accessed through User Interface



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:

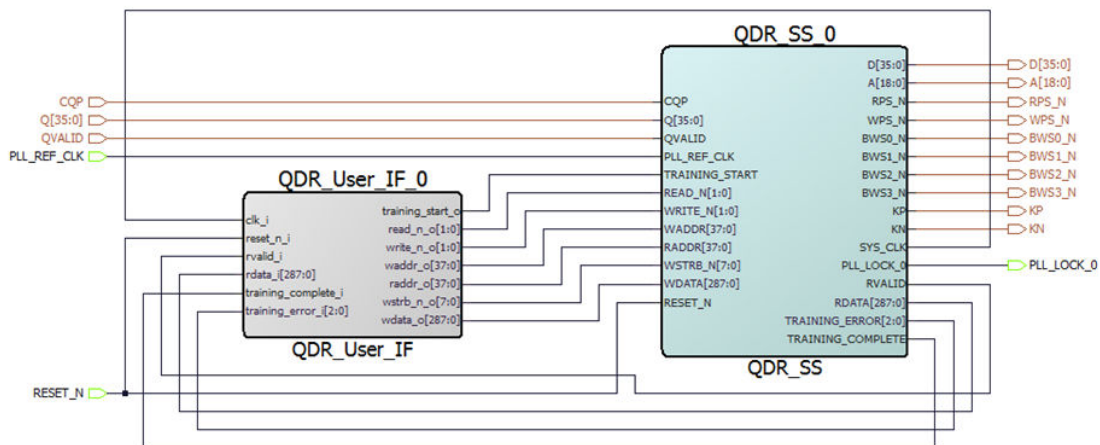
1. Create a SmartDesign, and instantiate the QDR component.
2. Configure the QDR Subsystem as described in [Implementation](#). In the following example, a design is created to access the QDR memory with 36-Bit Data Width, 19-Bit Address Width and Burst size of 4.

Figure 5.26. QDR Configuration- RTL\_FULL



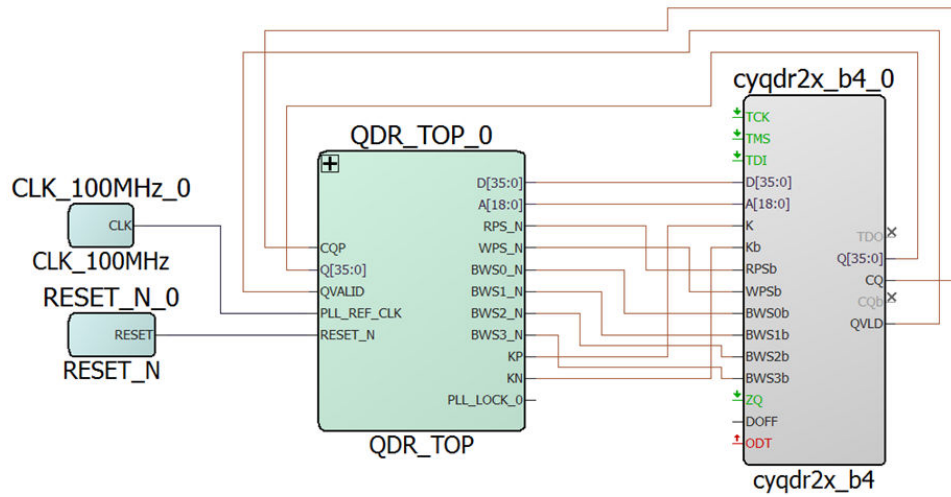
3. Instantiate the User Interface Master logic in the SmartDesign canvas. Ensure that the User Interface Master logic accesses the QDR subsystem only after TRAINING\_COMPLETE is high.
4. In the SmartDesign canvas, connect the blocks, as shown in the following figure.

Figure 5.27. SmartDesign Connections



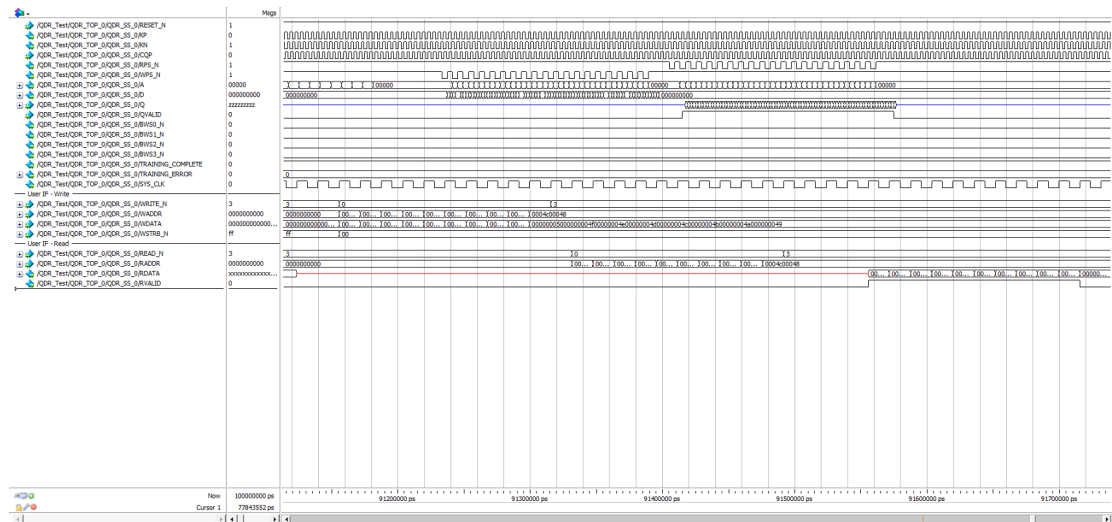
5. Create a new SmartDesign Testbench to simulate the design.
6. Instantiate the top-level design component and the QDR memory simulation model.
7. Configure CLK\_GEN to generate the PLL reference clock and connect to PLL\_REF\_CLK.
8. Connect the blocks in SmartDesign Testbench, as shown in the following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

Figure 5.28. 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 5.29. QDR Read and Write Transactions- RTL\_FULL



### 5.6.2. BFM Simulation [\(Ask a Question\)](#)

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 [RTL\\_FULL Simulation](#).
2. Select **Simulate** from Libero **Design Flow -> Verify Pre-synthesized Design**.



## 5.7. PCB Recommendations [\(Ask a Question\)](#)

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

**Table 5.8.** Maximum Length and Delay Constraints

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

**Table 5.9.** 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

## 6. PolarFire Board Design Recommendations

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

- GPIO supports DDR3.
- HSIO supports DDR4, DDR3, DDR3L, and LPDDR3.

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

**Table 6.1.** DDR3, LPDDR3, DDR3L, and DDR4 Parameters

Parameter	DDR3	LPDDR3	DDR3L	DDR4
VDDIO	1.5V	1.2V	1.35V	1.2V
VTT, VREF	0.75V	0.6V	0.675V	0.6V
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 <sup>1</sup>
RZQ	240_1%	240_1%	240_1%	240_1%

### Note:

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*



**Tip:** DQ bit swapping within a nibble is supported for PCB routing flexibility between DDR controller and the DDR\* memory. This must be done at PCB design, not by Libero I/O editor.

### 6.1. DDR3 [\(Ask a Question\)](#)

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 60Ω, and 120Ω. 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 50Ω, and  $Z_{diff}$  (differential) is 100Ω.

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

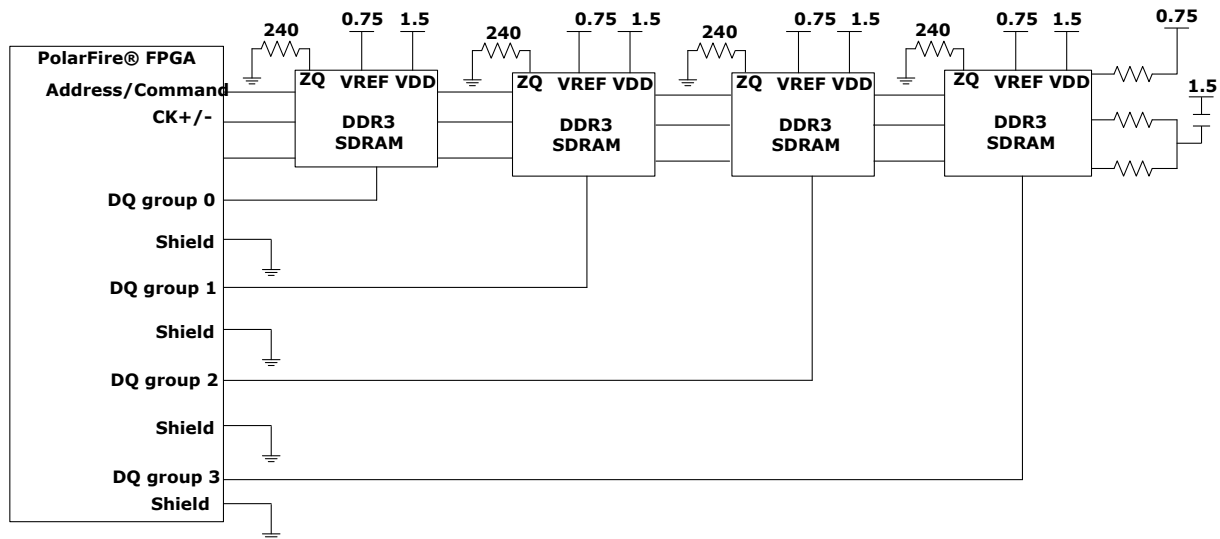
**Table 6.2.** Interfaces Supported in DDR3 Memory Devices

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

**Table 6.2.** Interfaces Supported in DDR3 Memory Devices (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 FPGAs.

**Figure 6.1.** DDR3 Interface Example

### 6.1.1. DDR3 Layout Guidelines [\(Ask a Question\)](#)

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 6.3.** DDR3 Interface Signals

Clock Signal	Description
CK0:1 P/N	Differential clock signals
<b>Control signals</b>	
CS	Chip Select
CKE	Clock Enable
ODT	On Die termination enable
Reset_n	Reset signal
<b>Address Signal</b>	
A[15:0]	Address Signals
BA[2:0]	Bank address
<b>Command Signal</b>	
RAS_n	Row address select
CAS_n	Column address select
WE_n	Write enable
<b>Data signals</b>	

Clock Signal	Description
DQ[71:0]	Data bit
DQM[8:0]	Data mask
DQS_P/N[8:0]	Data strobe

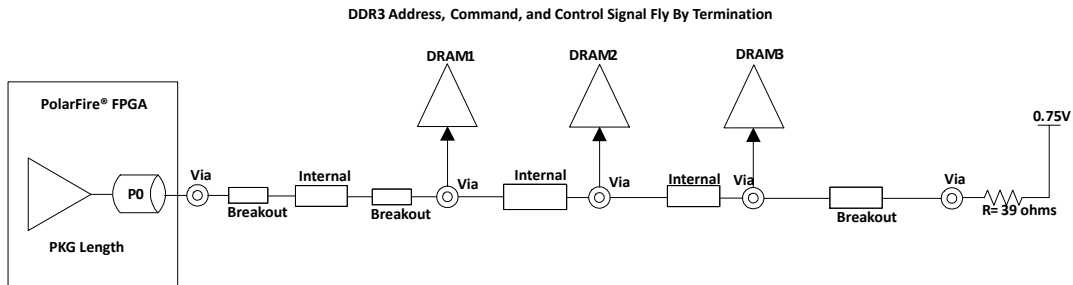
Due to high speed signaling, DDR3 uses fly-by routing topology for Address, Command, Control signals to achieve best performance. As shown in the following figure, the ADDR/CMD are routed as single-ended in fly-by topology. It is recommended to have all the signals being routed in one signal layer to 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.

DQ bit swapping within a byte lane is supported on the PCB between the DDR controller and the DDR3 memory for routing flexibility.

**6.1.1.1. DDR3 Routing Topology** [\(Ask a Question\)](#)

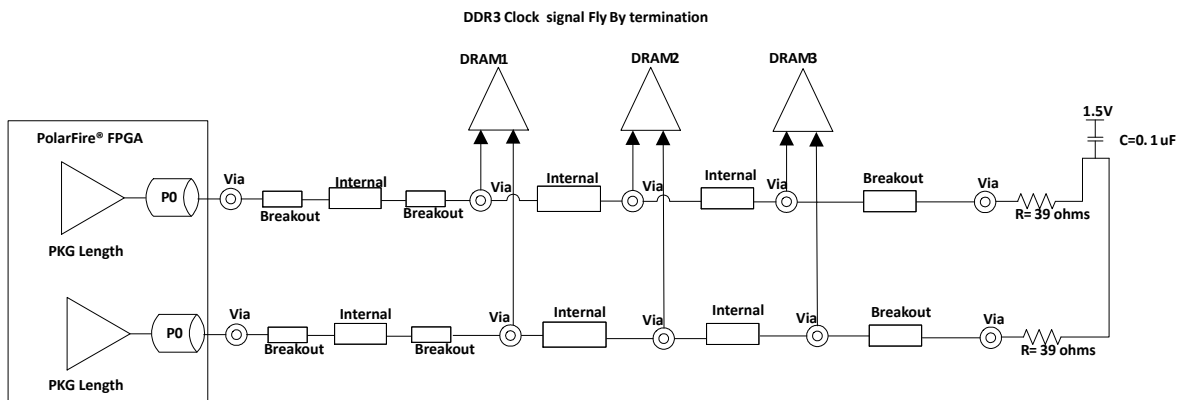
The following figure shows the FlyBy termination of the DDR3 address, command, and control signals (ACC).

**Figure 6.2.** DDR3 ACC Signals Fly By Termination

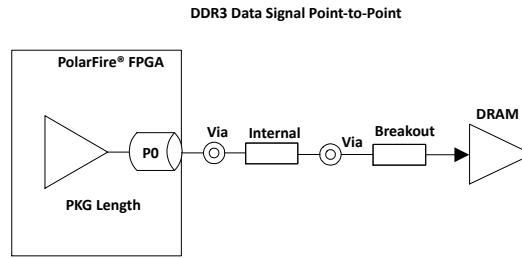


The following figure shows the FlyBy termination of the DDR3 clock signal.

**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 [\(Ask a Question\)](#)

1. The address and command groups must be skew matched with respect to CK signals within  $\pm 100$  ps.
2. The DQ, DM signals must be skew matched with respect to DQS within  $\pm 25$  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  $40\Omega$ – $50\Omega$  must be used for single ended signals and  $90\Omega$ – $100\Omega$  for differential signals.

## 6.2. LPDDR3 [\(Ask a Question\)](#)

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  $120\Omega$  and  $240\Omega$ .
- DQ lines do not need VTT termination.
- The characteristic impedance,  $Z_o$  is typically  $50\Omega$ , and  $Z_{diff}$  (differential) is  $100\Omega$ .

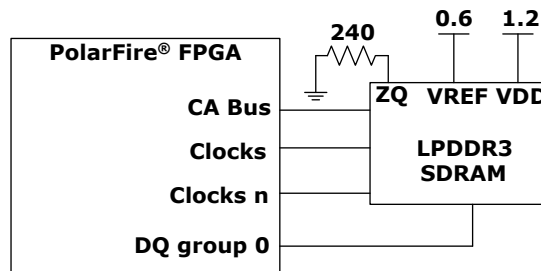
The following table lists the LPDDR3 memory interface features supported in PolarFire devices.

**Table 6.4.** Interfaces Supported in LPDDR3 Memory Devices

Interface	LPDDR3
Voltage	1.2V
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 6.5. LPDDR3 Interface Example



### 6.2.1. LPDDR3 Layout Guidelines [\(Ask a Question\)](#)

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.

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

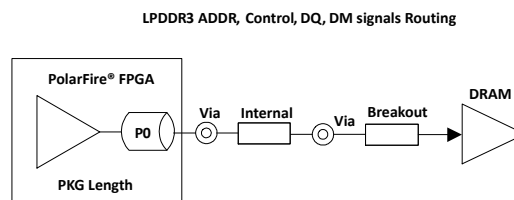
Microchip recommends simulating the interface to tightly match the characteristic impedance of trace with driver impedance to reduce reflections and for any possible electrical and timing violations in meeting JEDEC requirements.

DQ bit swapping within a byte lane is supported on the PCB between the DDR controller and the LPDDR3 memory for routing flexibility.

#### 6.2.1.1. LPDDR3 Routing Topology [\(Ask a Question\)](#)

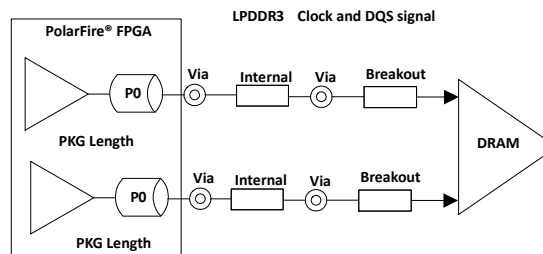
The following figure shows the routing of the LPDDR3 address, control, DQ, and DM signals.

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 [\(Ask a Question\)](#)

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

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

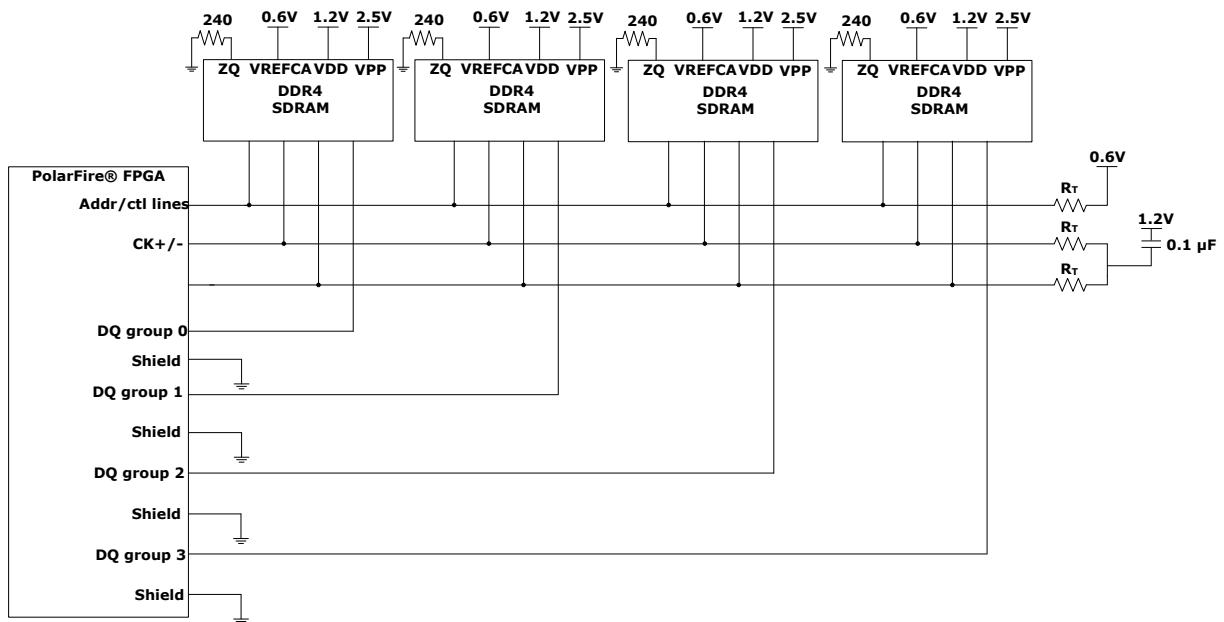
### 6.3. DDR4 [\(Ask a Question\)](#)

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 $\Omega$ , 120 $\Omega$ , and 240 $\Omega$ . 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 $\Omega$  and  $Z_{diff}$  (differential) is 100 $\Omega$ .

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

Figure 6.8. DDR4 Interface Example



**Note:** Each address and command line should use fly-by routing to VTT at the end of the net. Termination resistor ( $R_T$ ) value, between 30–50 $\Omega$ , is typically used for this. The user must simulate to optimize and ensure that optimal termination resistor ( $R_T$ ) value is used.

### 6.3.1. DDR4 Layout Guidelines [\(Ask a Question\)](#)

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 6.5. DDR4 Interface Signals

Clock Signals	Description
CK[1:0]_P/N	Differential Clock signals

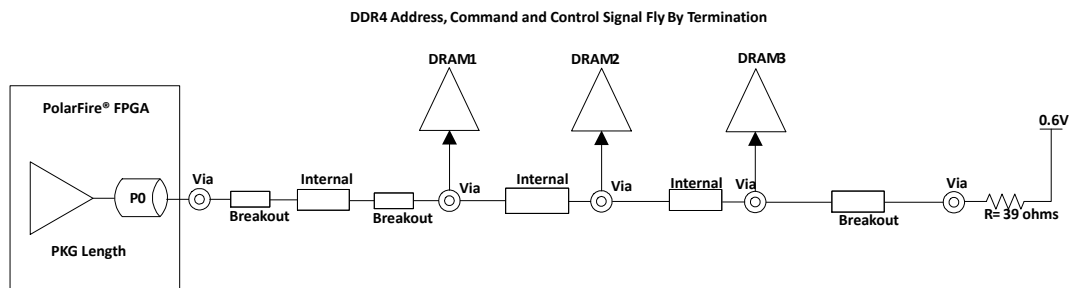
Table 6.5. DDR4 Interface Signals (continued)	
Clock Signals	Description
<b>Address and Command Signals</b>	
A[15:0]	Address signals
BA[1:0]	Bank Address signals
BG[1:0]	Bank Group signals
WE	Write Enable
RAS_n	Row address strobe
CAS_n	Column address strobe
ACT_n	Activation Command Input
PAR	Command and address parity
<b>Control Signals</b>	
CKE	Clock Enable
CS_n[1:0]	Chip Select
ODT	On die termination
Reset	Reset
<b>Data Group</b>	
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.

### 6.3.1.1. DDR4 Routing Topology [\(Ask a Question\)](#)

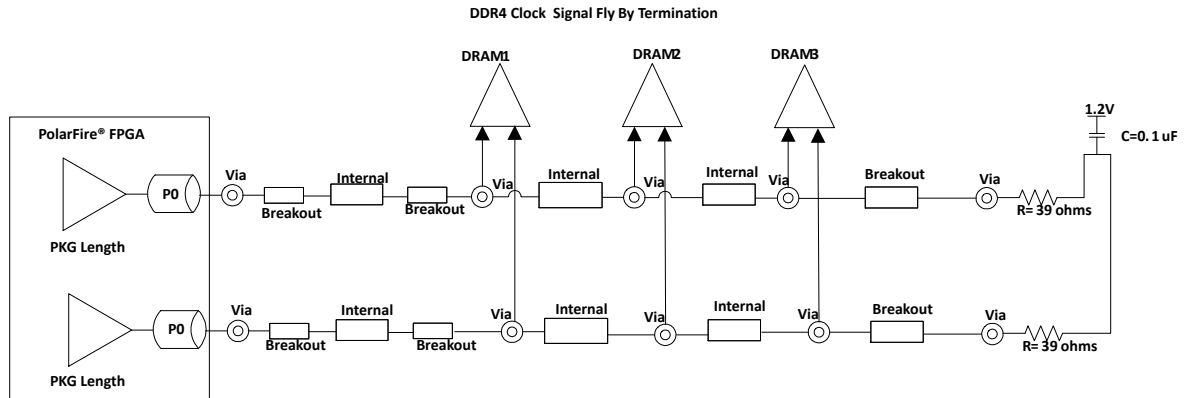
The following figure shows the Flyby termination of DDR4 ADDR/CMD signals.

Figure 6.9. DDR4 ACC Signals FlyBy Termination



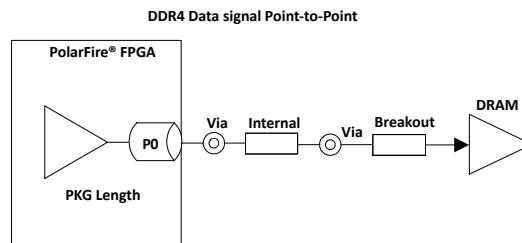
The following figure shows the FlyBy termination of the DDR4 clock signal.

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



### 6.3.1.2. Skew matching constraints [\(Ask a Question\)](#)

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

## 7. PolarFire SoC Board Design Recommendations

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

- GPIO supports DDR3.
- HSIO supports DDR4, DDR3, DDR3L, LPDDR3, and LPDDR4.

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

**Table 7.1.** DDR3, LPDDR3, DDR3L, LPDDR4, and DDR4 Parameters

Parameter	DDR3	LPDDR3	DDR3L	DDR4	LPDDR4
VDDIO	1.5V	1.2V	1.35	1.2V	1.1V
VTT, VREF	0.75V	0.6V	0.675V	0.6V	0.55V
Clock, Address, and Command (CAC) Layout	Daisy Chained (Fly-by)	Point to point	Daisy Chained (Fly-by)	Daisy Chained (Fly-by)	Point-to-Point <sup>2</sup>
Data Strobe	Differential	Differential	Differential	Differential	Differential
ODT	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Match Addr/CMD/Ctrl to Clock Tightly	Yes	Yes	Yes	Yes	Yes
Match DQ/DM/DQS Tightly	Yes	Yes	Yes	Yes	Yes
Match DQS to Clock Loosely	Not Required	Not Required	Not Required	Not Required	Not Required
I/O Standard	SSTL_15	HSUL-12	SSTL_135	POD-12 <sup>1</sup>	LVSTL 1.1V
RZQ	240_1%	240_1%	240_1%	240_1%	240_1%

### Notes:

1. HSTL\_12 IO standard is used for DDR4 address, command, and control signals.
2. For dual channel, LPDDR4 needs to go with T-topology routing. For more information, see [LPDDR4](#).

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*



**Tip:** DQ bit swapping within a nibble is supported on the PCB for routing flexibility as follows:

- Between the Fabric DDR controller and DDR3, DDR3L, LPDDR3, and DDR4 memories
- Between the MSS DDR controller and DDR3, DDR3L, LPDDR3, and DDR4 memories

### 7.1. DDR3 [\(Ask a Question\)](#)

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

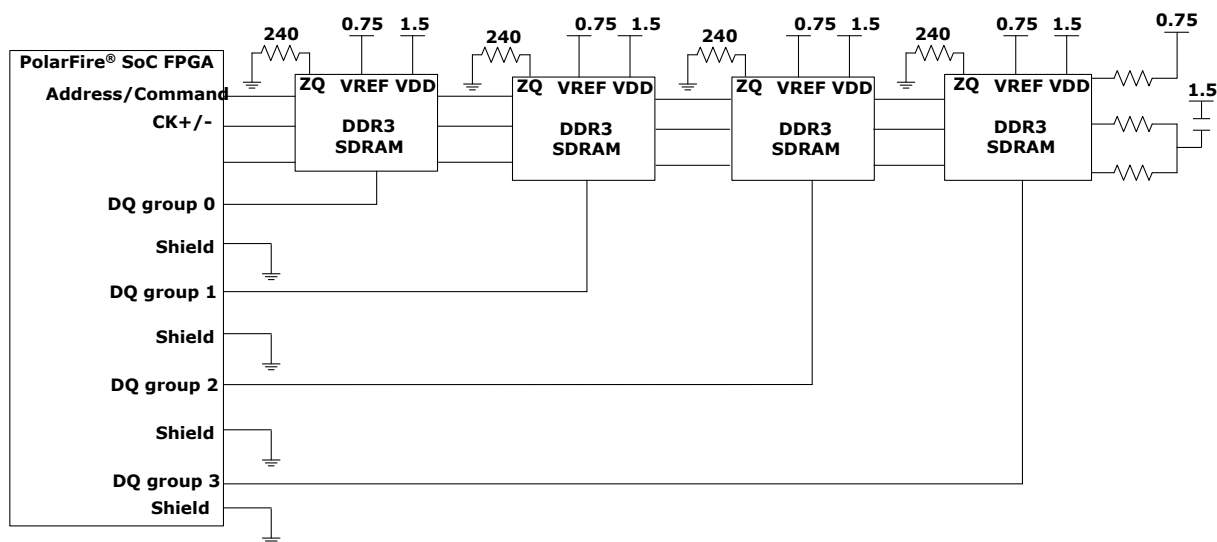
- DDR3 data nets have dynamic ODT built into the controller and SDRAM. The configurations are 60Ω and 120Ω. DQ lines do not need VTT termination. However, VTT termination resistors must be placed at the end of address and control lines on the PCB.
- Characteristic impedance:  $Z_0$  is typically 50Ω, and  $Z_{diff}$  (differential) is 100Ω.

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

**Table 7.2.** Interfaces Supported in DDR3 Memory Devices

Interface	DDR3
Voltage	1.5V
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
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 7.1.** DDR3 Interface Example

### 7.1.1. DDR3 Layout Guidelines [\(Ask a Question\)](#)

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 7.3.** DDR3 Interface Signals

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

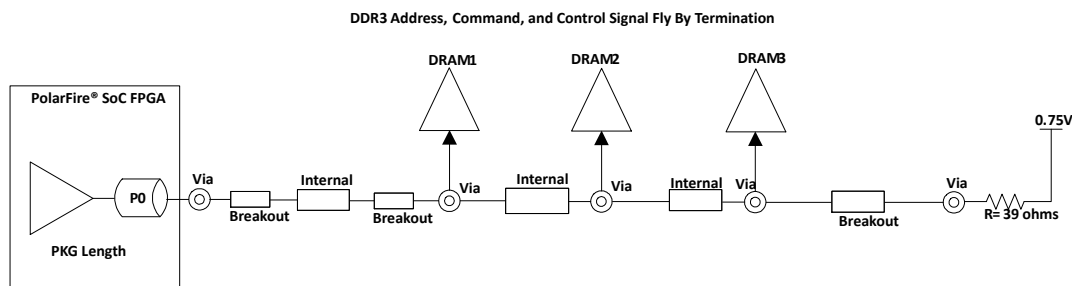
Table 7.3. DDR3 Interface Signals (continued)	
Clock Signal	Description
CAS_n	Column address select
WE_n	Write enable
Data Signals	
DQ[31:0]	Data bit
DM[3:0]	Data mask
DQS_P/N[8:0]	Data strobe

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.

### 7.1.1.1. DDR3 Routing Topology [\(Ask a Question\)](#)

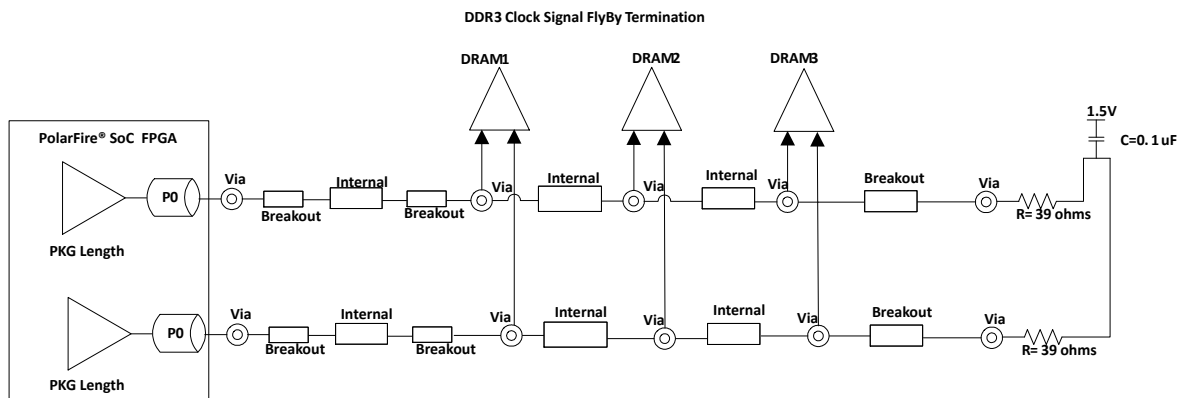
The following figure shows the FlyBy termination of the DDR3 Address, Command, and Control signals (ACC).

Figure 7.2. DDR3 ACC Signals Fly By Termination

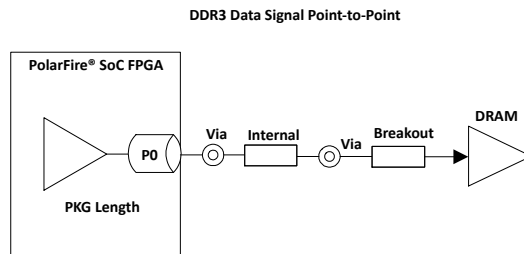


The following figure shows the FlyBy termination of the DDR3 clock signal.

Figure 7.3. DDR3 Clock Signal FlyBy Termination



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

**Figure 7.4.** DDR3 Data Signal Point-to-Point

### 7.1.1.2. Skew Constraints For Signal Groups [\(Ask a Question\)](#)

1. The address and command groups must be skew matched with respect to CK signals within  $\pm 100$  ps.
2. The DQ, DM signals must be skew matched with respect to DQS within  $\pm 25$  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  $40\Omega$ – $50\Omega$  must be used for single-ended signals and  $90\Omega$ – $100\Omega$  for differential signals.

## 7.2. LPDDR3 [\(Ask a Question\)](#)

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  $120\Omega$  and  $240\Omega$ .
- DQ lines do not need VTT termination.
- The characteristic impedance,  $Z_o$  is typically  $50\Omega$  and  $Z_{diff}$  (differential) is  $100\Omega$ .

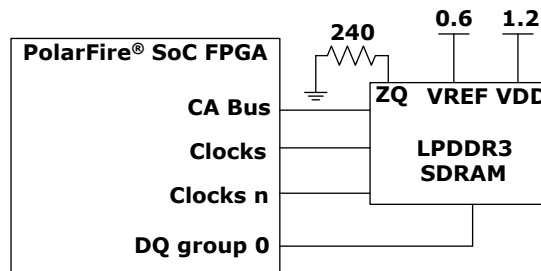
The following table lists the LPDDR3 memory interface features supported in PolarFire SoC devices.

**Table 7.4.** Interfaces Supported in LPDDR3 Memory Devices

Interface	LPDDR3
Voltage	1.2V
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 7.5. LPDDR3 Interface Example



**7.2.1. LPDDR3 Layout Guidelines** [\(Ask a Question\)](#)

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.

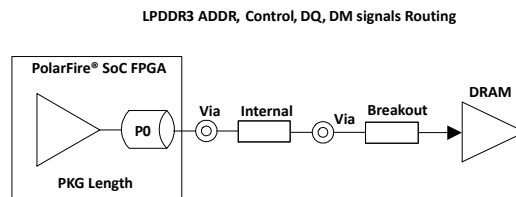
The characteristic impedance of 40Ω–50Ω must be used for single ended signals and 90Ω–100Ω for differential signals.

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.

**7.2.1.1. LPDDR3 Routing Topology** [\(Ask a Question\)](#)

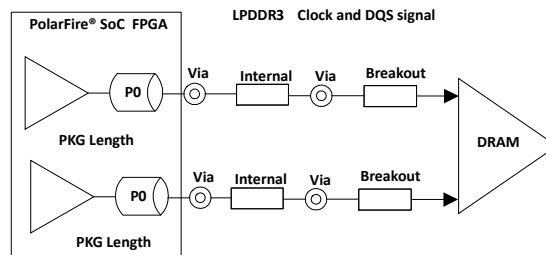
The following figure shows the routing of the LPDDR3 address, control, DQ, and DM signals.

Figure 7.6. LPDDR3 Address/Control/DQ/DM Signal Routing



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

Figure 7.7. LPDDR3 Clock and DQS signal Routing



**7.2.1.2. Skew Matching Constraints** [\(Ask a Question\)](#)

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  $\pm 30$  ps.
3. Skew between CA and CK group and data lane group must not be more than  $\pm 350$  ps.

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

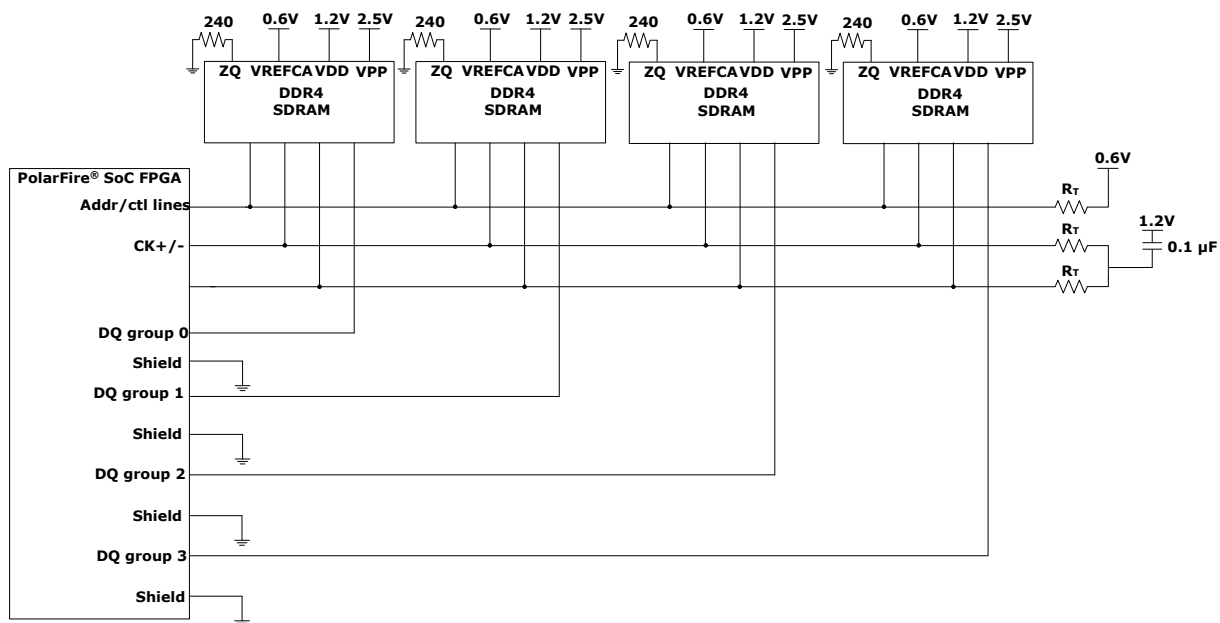
### 7.3. DDR4 [\(Ask a Question\)](#)

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 $\Omega$ , 120 $\Omega$ , and 240 $\Omega$ . 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 $\Omega$  and  $Z_{diff}$  (differential) is 100 $\Omega$ .

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

**Figure 7.8.** DDR4 Interface Example



**Note:** Each address and command line should use fly-by routing to VTT at the end of the net. Termination resistor ( $R_T$ ) value, between 30–50 $\Omega$ , is typically used for this. The user must simulate to optimize and ensure that optimal termination resistor ( $R_T$ ) value is used.

#### 7.3.1. DDR4 Layout Guidelines [\(Ask a Question\)](#)

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 7.5.** DDR4 Interface Signals

Clock Signals	Description
CK[1:0]_P/N	Differential Clock signals
<b>Address and Command Signals</b>	

**Table 7.5.** DDR4 Interface Signals (continued)

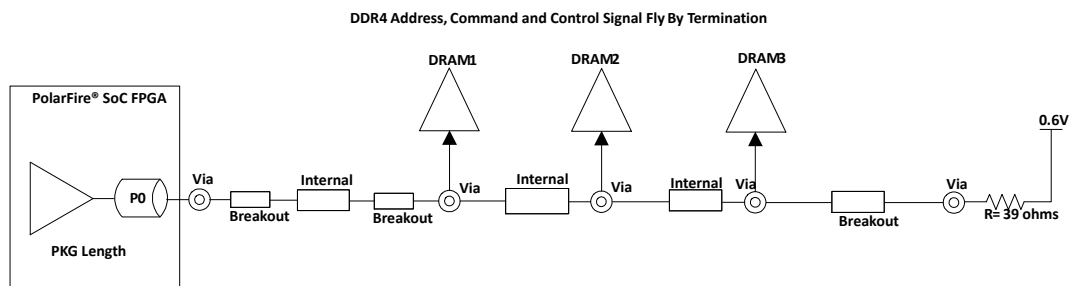
Clock Signals	Description
A[12:0]	Address signals
BA[1:0]	Bank Address signals
BG[1:0]	Bank Group signals
WE	Write Enable
RAS_n	Row address strobe
CAS_n	Column address strobe
ACT_n	Activation Command Input
PAR	Command and address parity
<b>Control Signals</b>	
CKE	Clock Enable
CS_n[1:0]	Chip Select
ODT	On die termination
Reset	Reset
<b>Data Group</b>	
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.

### 7.3.2. DDR4 Routing Topology [\(Ask a Question\)](#)

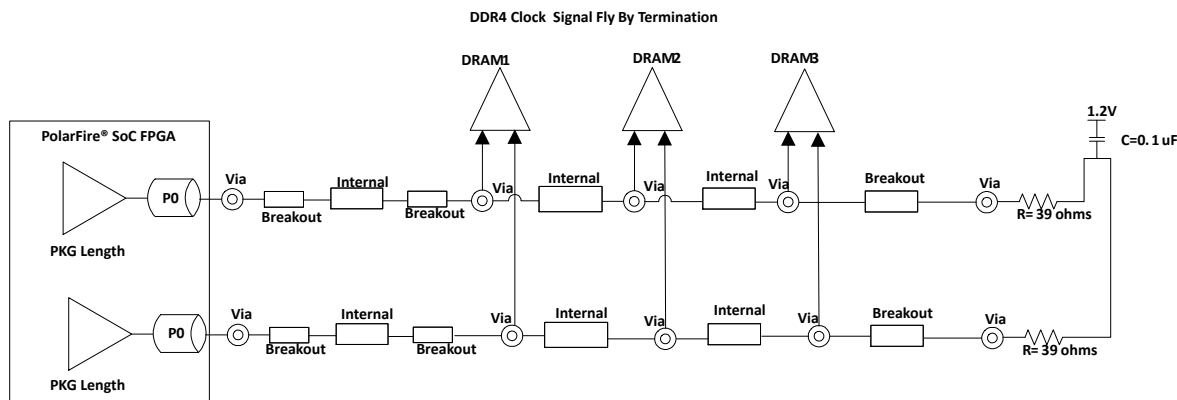
The following figure shows the Flyby termination of DDR4 ADDR/CMD signals.

**Figure 7.9.** DDR4 ACC Signals FlyBy Termination



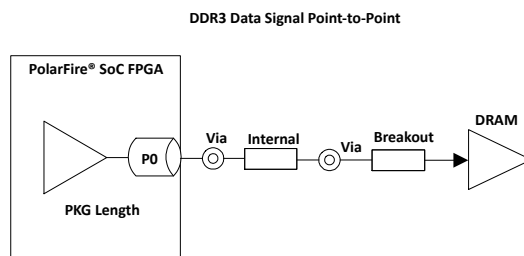
The following figure shows the FlyBy termination of the DDR4 clock signal.

Figure 7.10. DDR4 Clock Signal FlyBy Termination



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

Figure 7.11. DDR4 Data Signal Point-to-Point



#### 7.3.2.1. Skew Matching Constraints [\(Ask a Question\)](#)

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

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

## 7.4. LPDDR4 [\(Ask a Question\)](#)

The following are the guidelines for connecting the device to the LPDDR4 memory:

- LPDDR4 nets (Data and CA) have dynamic ODT built into the controller and SDRAM. The ODT configurations available at memory are 40, 48, 60, 80, 120, and 240 and on controller side options are 34, 40, 48, and 60.
- Characteristic impedance for traces:  $Z_0$  (single-ended) typically can be either  $50\Omega$  or  $40\Omega$  and  $Z_{diff}$  (differential) can be either  $100\Omega$  or  $80\Omega$ .

LPDDR4 memory comes with different options in terms of channel and rank depending upon the memory size used in the design. Commonly available options are:

- Single-channel, single rank
- Dual-channel, single rank
- Dual-channel, dual ranks

PolarFire SoC LPDDR4 controller supports single rank and single channel. Dual-channel LPDDR4 is supported by concatenating two single-channel on the controller side within FPGA design, and

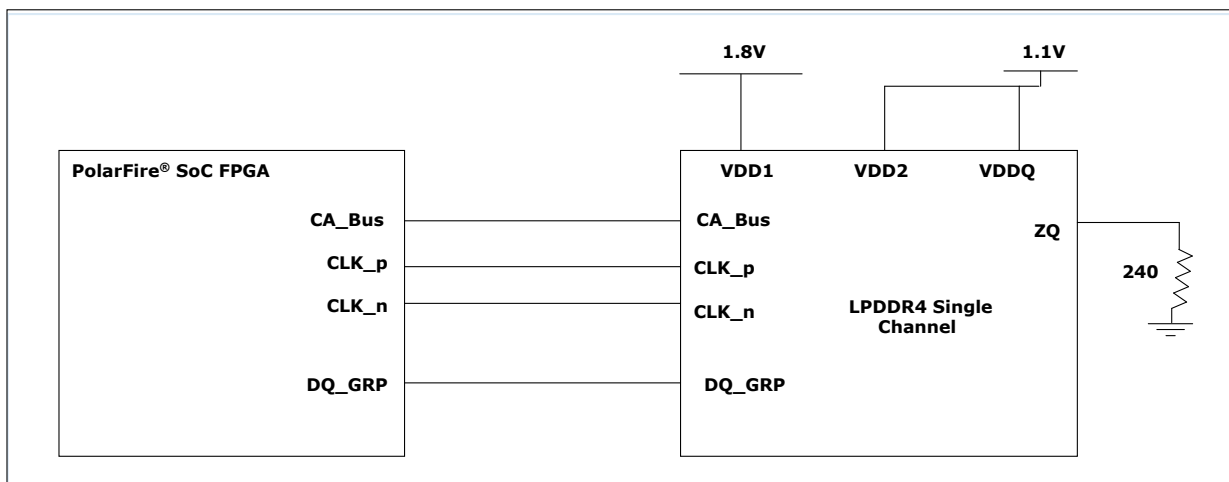
on-board needs to short the Channel-A address of LPDDR4 with the Channel-B address for full memory access.

**Table 7.6.** Supported LPDDR4 Use Models in PolarFire SoC FPGA

LPDDR4 Memory Configuration (Commonly Available)	PolarFire® SoC FPGA
Single-channel, single rank (x16)	Supported
Dual-channel, single rank (x32)	Supported
Dual-channel, dual rank (x32)	Not supported

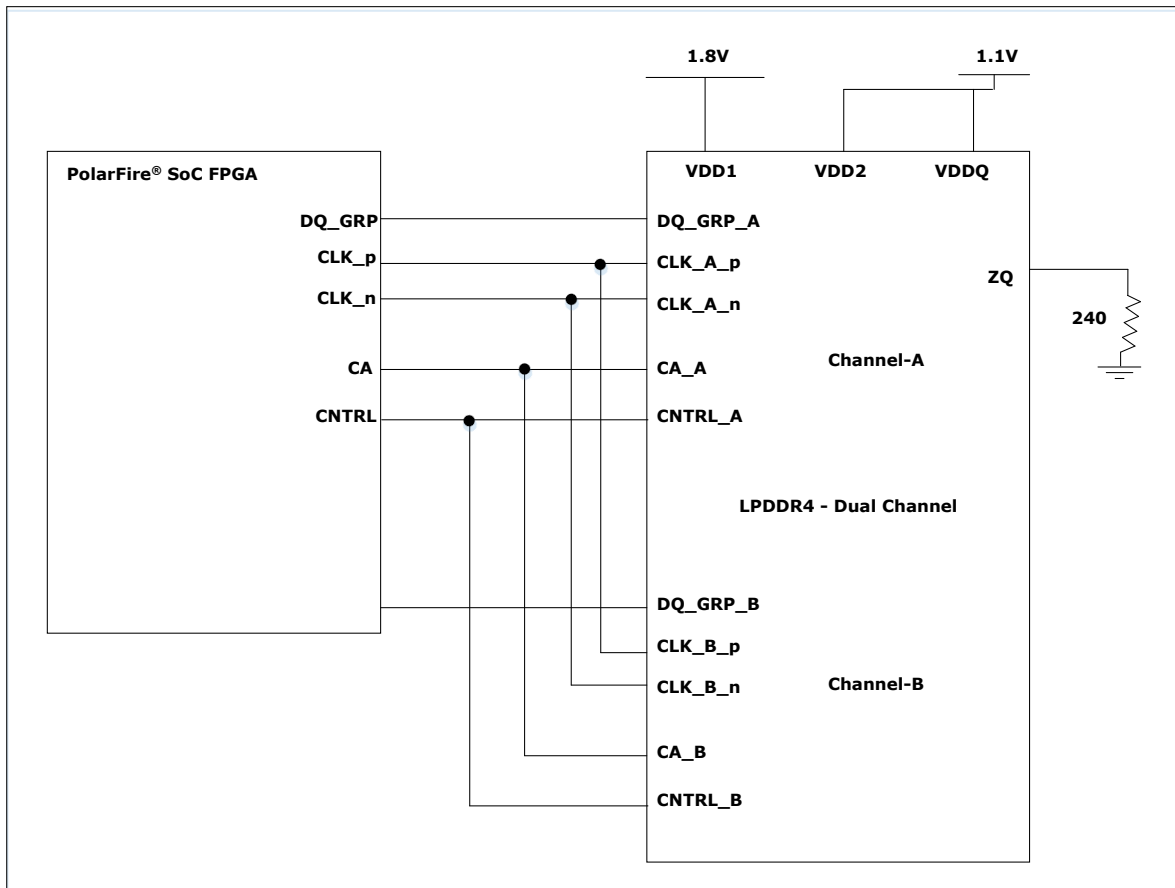
The following figure shows the PolarFire SoC interface with single-channel, single-rank LPDDR4.

**Figure 7.12.** PolarFire SoC Interface with Single-Channel, Single-Rank LPDDR4



The following figure shows the PolarFire SoC interface with dual-channel, single-rank LPDDR4.

Figure 7.13. PolarFire SoC Interface with Dual-Channel, Single-Rank LPDDR4



The following table lists the LPDDR4 memory interface features supported in PolarFire SoC FPGA.

Table 7.7. Interfaces Supported in LPDDR4 Memory Devices

Interface	LPDDR4
Voltage	1.1V
I/O standard	LVSTL 1.1V
Data rate	1600MT/S—MDDR I/Os (Bank 6)
Termination	ODT in $\Omega$ (34, 40, 48, and 60)
Routing Topology	Point-to-point

7.4.1. LPDDR4 Layout Guidelines [\(Ask a Question\)](#)

This section describes the routing guidelines for LPDDR4 interface for PolarFire SoC Family. The guidelines are with reference to maximum x32 data width from a signal integrity perspective. It is recommended to evaluate the interface by performing system-level signal integrity simulations. The user is assumed to have knowledge of the memory interface guidelines.

Table 7.8. LPDDR4 Interface Signals

Clock Signal	Description
CK_c/t	Differential clock signals
<b>Address and Control Signals</b>	
CA [5:0]	Address signals
CKE	Clock enable

**Table 7.8.** LPDDR4 Interface Signals (continued)

Clock Signal	Description
CS	Chip select
Reset_n	Memory reset signal
ODT	On die termination
Data Group Signals	
DQ [15:0]	Data signals
DMI[1:0]	Data mask signal
DQS_c/t [0:1]	Data strobe signals

### 7.4.2. LPDDR4 Routing Topology [\(Ask a Question\)](#)

LPDDR4 routing can be performed in following channels depending upon the memory size used in the design.

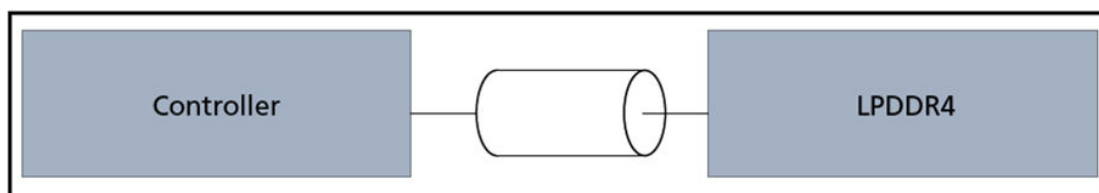
- [Single-channel LPDDR4 Interface Routing](#)
- [Dual-Channel LPDDR4 Interface Routing](#)

#### 7.4.2.1. Single-channel LPDDR4 Interface Routing [\(Ask a Question\)](#)

The following figure shows LPDDR4 routing with Single-channel and single-rank devices. The topology is used to point-to-point for single-channel LPDDR4. All single-ended signals (CA/Data/Cntrl) are routed point-to-point, as shown in the following figure.

**Figure 7.14.** Single-Channel LPDDR4 Interface Routing

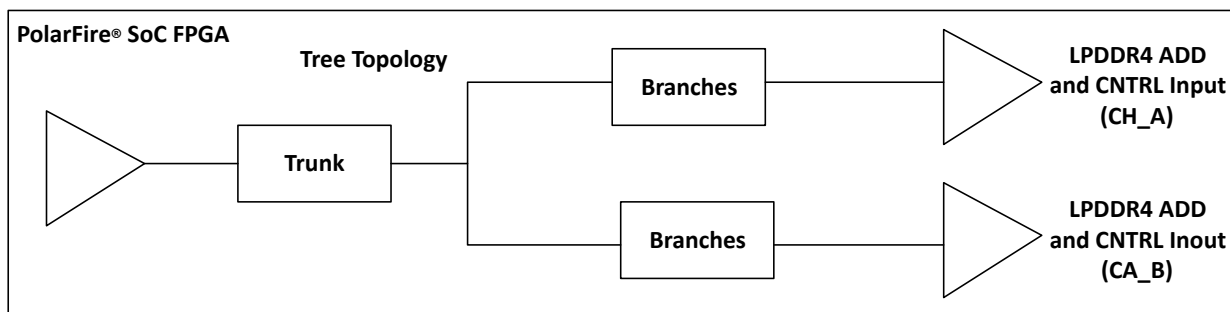
#### Point-to-point topology



#### 7.4.2.2. Dual-Channel LPDDR4 Interface Routing [\(Ask a Question\)](#)

For dual-channel LPDDR4 routing, balanced T-topology routing is preferred for CA and Control bus routing. Data group signals are always point-to-point routing.

**Figure 7.15.** Dual-Channel LPDDR4 Interface Routing



#### 7.4.2.3. Routing Recommendation [\(Ask a Question\)](#)

Following are the routing recommendations for LPDDR4.

- Trace impedance can be 40/80 or 50/100.
- All CA signals and CNTRL signals must length match with CLK.
- DQ/DQS signal must length match.
- Try to make branches short.
- Equal branch length gives better signal integrity performance.
- Reduce the number of VIA on overall DDR lines.
- All DQ/DQS and CLK/CA/Control group signal routing must be similar and transition must be same. For net layer transitions, Microchip recommends routing ground vias next to signal vias for good shielding and to maintain constant impedance.
- Typically, the trunk and the branches are the same impedance. If the branches are kept short, the reflection that returns to the component from the mismatch in impedance between the trunk and branches occurs during the rise time and is not observed as a separate event.
- Slew rate is dependent on the capacitance at the load and the source impedance. The source impedance is the driver impedance plus the transmission line impedance (mostly the trunk). To improve slew rate, consider shading the driver impedance to and the trunk impedance on the low side.
- Ensure that the signals do not cross the split plane and have a reference plane throughout.
- Limit the overall trace length to below two inches.

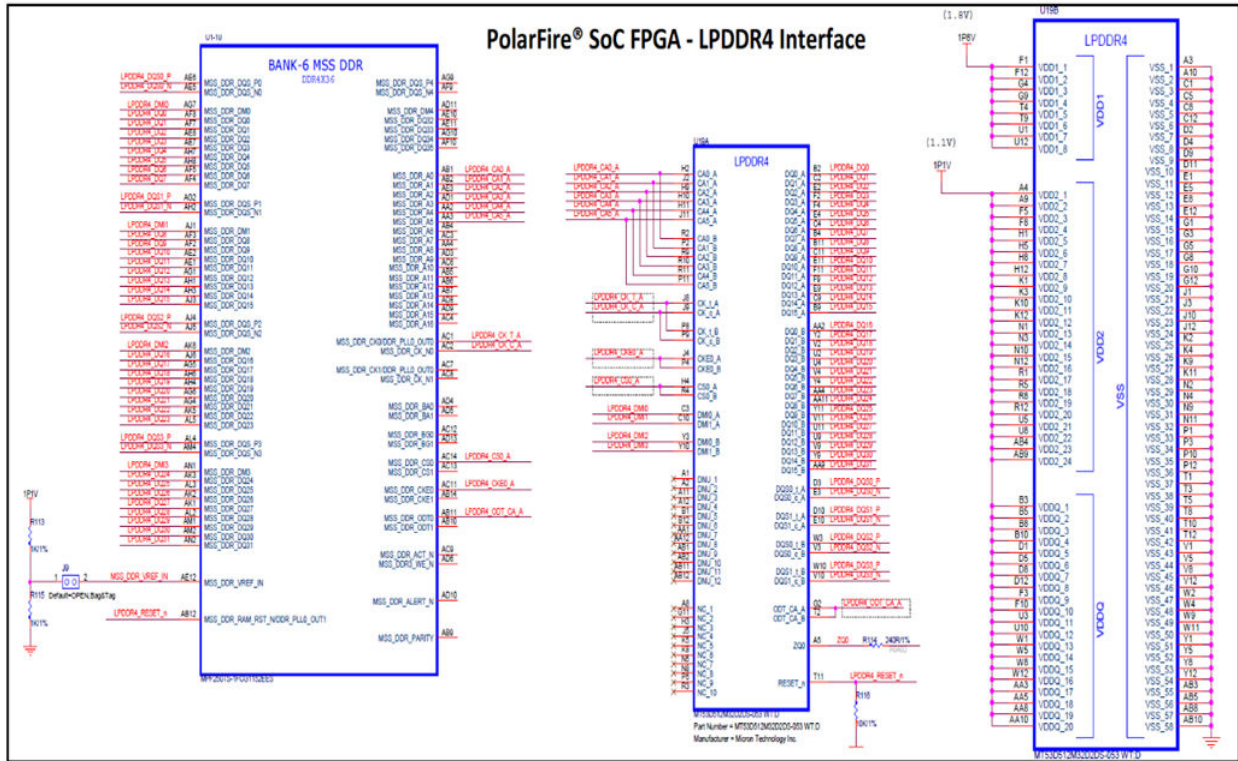
#### 7.4.2.4. Skew Machine Constraints [\(Ask a Question\)](#)

- The address and control signals group to be tightly length match, skew between them must not exceed 25 ps–50 ps.
- DQ/DQS group routing is point-to-point and the skew between DATA/DQS also must not exceed 50 ps.
- Skew between CLK/CA/control group and DQ/DQS group must not exceed 100 ps.
- Good to have a longer clock/CA trace than DQS.

#### 7.4.3. Designing LPDDR4 Interface with PolarFire SoC FPGA [\(Ask a Question\)](#)

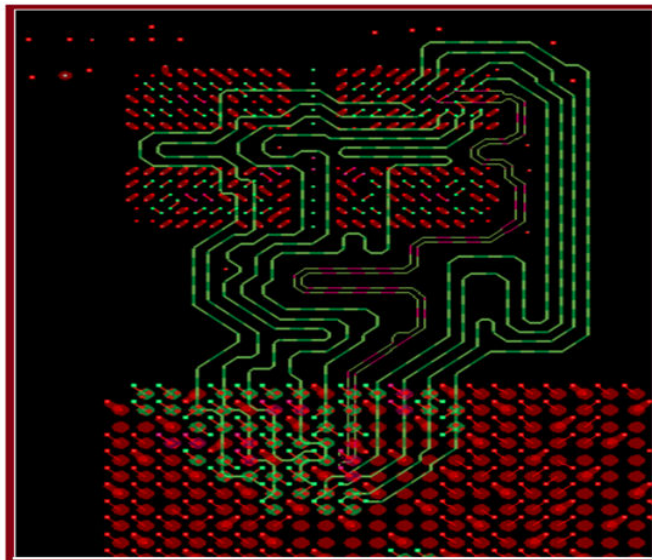
The following figure shows an example design for PolarFire SoC and LPDDR interface. In the example, the LPDDR4 memory is used from Micron and has dual-channel and single-rank with x32 data width.

Figure 7.16. Schematic of PolarFire SoC and LPDDR4 interface



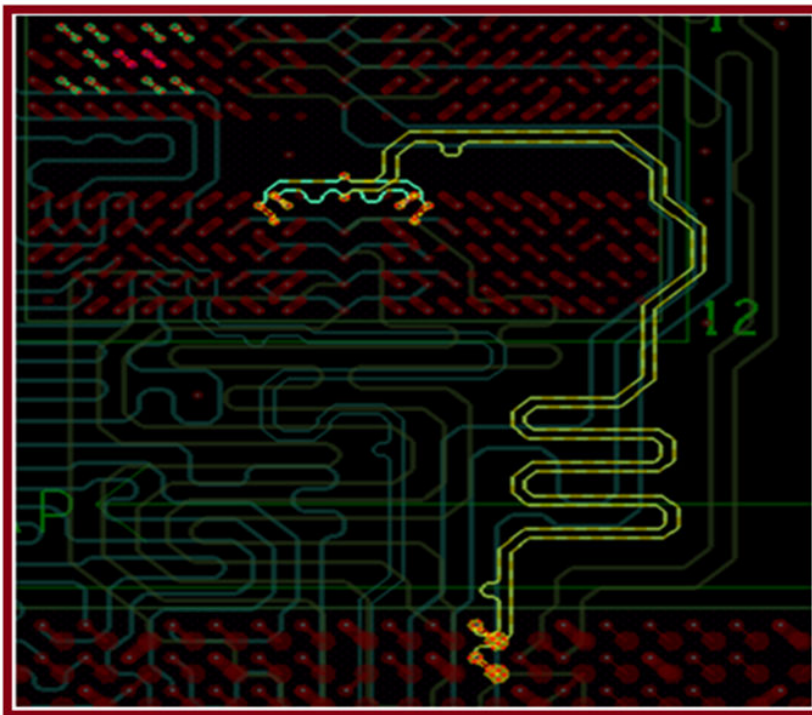
The following figure shows the DQ Bus Routing (Point-to-Point).

Figure 7.17. DQ Bus Routing (Point-to-point)



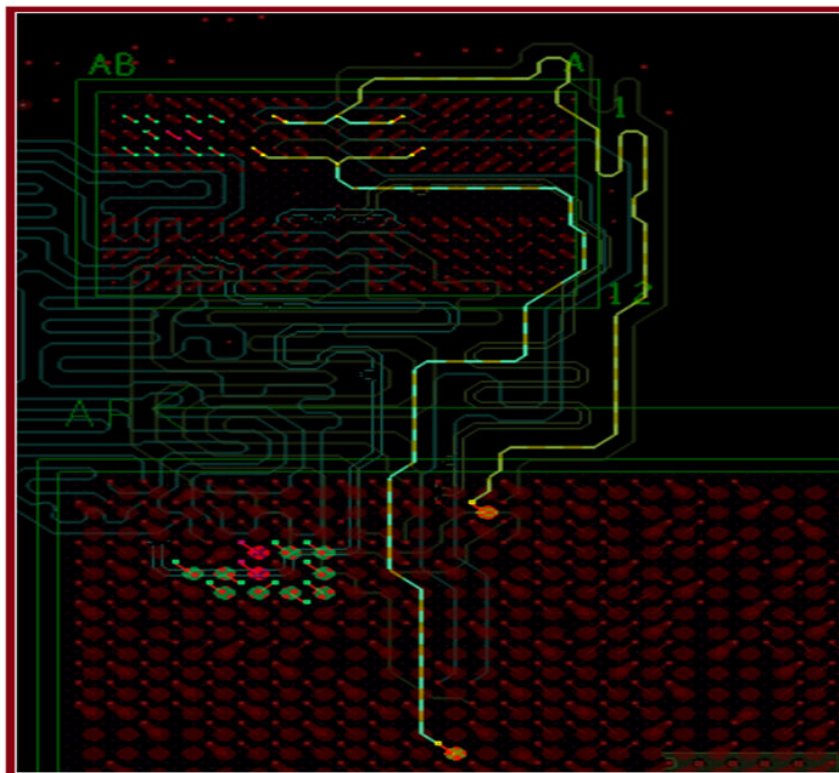
The following figure shows the Clk routing (Balance Topology).

Figure 7.18. Clk Routing (Balance Topology)



The following figure shows the CA bus routing (balance topology).

Figure 7.19. CA Bus Routing (Balance Topology)



## 8. 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.

### 8.1. Debug Tools [\(Ask a Question\)](#)

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

#### 8.1.1. SmartDebug [\(Ask a Question\)](#)

SmartDebug Debug DDR Memory GUI capability is automatically provided when a Microchip Fabric DDR Subsystem is hosted in the FPGA fabric and DDR I/O. Microchip's SmartDebug supports the capability of analyzing and displaying the DDR training data including DDR I/O margin data for all DDR memory controller instances used in a design. The information gathered by the SmartDebug tool provides a quick status of the DDR interface initialization between the FPGA DDR IO and the memory device. The information can be used to make design updates to better match the system interface to the FPGA. For more information, see [SmartDebug User Guide](#).

#### 8.1.2. Identify RTL Debugger [\(Ask a Question\)](#)

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 through the JTAG interface to first retrieve the values for probed signals from the programmed FPGA and then display the retrieved signals.

#### 8.1.3. Hardware Probes [\(Ask a Question\)](#)

PolarFire family of 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.

For debugging LPDDR4 and DDR4 training issues, use [Memory Log Analyzer](#) GUI. This is only applicable for embedded DDR memory controller.

## 8.2. Design Debug During Simulation [\(Ask a Question\)](#)

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.

## 8.3. Hardware Debug [\(Ask a Question\)](#)

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 the [References](#) section. 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.

## A. Appendix: Supported Memory Configurations [\(Ask a Question\)](#)

This section provides details on the available configurations for the different memory types supported in PolarFire devices.

### DDR4

The PolarFire DDR Subsystem supports non-3DS DDR4 devices.

DDR4 SDRAM is available in Dual In-Line Memory Modules (DIMM), Small Outline DIMMs (SO-DIMM), and as discrete devices. For information about the number of rows, columns, and bank bits required for DIMM or SO-DIMM configuration, refer to the data sheet associated with the DIMM module connected to the PolarFire device.

The following table lists the number of row bits, column bits, bank bits, and chip selects required for standard discrete DDR4 SDRAM devices in non-3DS configurations.

**Table A.61.** Non-3DS DDR4 SDRAM Device Configurations

Stack/Chip Size	Configuration	Row Bits	Column Bits	Bank Group Bits	Bank Bits
2	128M × 16	14	10	1	2
	64M × 32	13	10	1	2

### DDR3 and LPDDR3

The following table lists the number of row bits, column bits, and bank bits required for standard discrete DDR3, LPDDR3, and SDRAM devices.

**Table A.62.** Standard DDR3, LPDDR3, and SDRAM Device Configurations

Memory Type	Chip Size	Configuration	Row Bits	Column Bits	Bank Bits
DDR3	512 Mb	32M × 16	12	10	3
	1024 Mb	64M × 16	13	10	3
	2048 Mb	128M × 16	14	10	3
	4096 Mb	256M × 16	15	10	3
	8192 Mb	512M × 16	16	10	3
LPDDR3	4 Gb	128M × 32	14	10	3
	6 Gb	192M × 32	15	10	3
	8 Gb	128M × 32	15	10	3
	12 Gb	384M × 32	15	11	3
	16 Gb	512M × 32	15	11	3

## B. Appendix: Timing Parameters [\(Ask a Question\)](#)


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

**Table B.63.** 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

## C. Appendix: Fabric DDR Placement Locations [\(Ask a Question\)](#)

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

 **Important:** Migration between FPGA device and package must be considered prior to final pin out determination for memory interfaces. For DDR interface migration guidance, see the [PolarFire PPATs](#).

**Table C.64. 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	✓	✓	—	—
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	—	—	—	—

**➔ Important:** If a third party DDR Controller is used that supports data in place of ECC bits, the Max Width includes an additional 8 data bits. For example, a Max Width of 32 with an entry of “Yes” for ECC becomes a Max Width of 40 data bits.

Table C.65. 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 C.66. 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	✓	✓	—	—
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	✓	✓	—	—

Table C.66. NORTH\_NW Anchor (HSIO) (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_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 C.67. 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 C.68. 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	—	—	—	—

Table C.68. SOUTH\_SE Anchor (HSIO) (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	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 C.69. 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 C.70. 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	—	—	—	—

Table C.70. SOUTH\_SW Anchor (GPIO) (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 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	—	—	—	—
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 C.71. 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	—	—	—	—

Table C.71. WEST\_NW Anchor (GPIO) (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 C.72. 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 C.73. 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	—	—	—	—
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	—	—	—	—

Table C.73. WEST\_SW Anchor (GPIO) (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	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 C.74. 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	—	✓	✓	—
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	—	—	✓	✓

## Revision History [\(Ask a Question\)](#)

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
H	05/2025	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>• Added <a href="#">Table 2.6</a> in <a href="#">ECC</a> section.</li> <li>• Added a <a href="#">note</a> under <a href="#">Table 2.6</a>.</li> <li>• Changed the valid range of "CK/CA additive offset" configuration option to "0-7" in <a href="#">Configuring the DDR3 Subsystem</a>, <a href="#">Configuring the LPDDR3 Subsystem</a> and <a href="#">Configuring the DDR4 Subsystem</a>.</li> <li>• Removed the "ODT Activation Settings on Read" configuration options from <a href="#">Configuring the DDR3 Subsystem</a>, <a href="#">Configuring the LPDDR3 Subsystem</a> and <a href="#">Configuring the DDR4 Subsystem</a>.</li> <li>• Updated the assertion behavior of the Reinitialization Control signal (CTRLR_INIT) in <a href="#">Table 2.13</a> in <a href="#">DDR Subsystem Ports</a>, <a href="#">Configuring the DDR3 Subsystem</a>, <a href="#">Configuring the LPDDR3 Subsystem</a> and <a href="#">Configuring the DDR4 Subsystem</a>.</li> <li>• Added a <a href="#">note</a> under <a href="#">Table 2.26</a> to describe a high-level recommendation on synchronizing the ARST_N signal of the PF_IOD_OCTAL_DDR IP.</li> <li>• Updated the <a href="#">note</a> in <a href="#">PolarFire Board Design Recommendations</a>.</li> <li>• Added a <a href="#">note</a> under <a href="#">Figure 6.8</a> and <a href="#">Figure 7.8</a> to describe a high-level guideline for using an optimal termination resistor (R<sub>T</sub>) value.</li> <li>• Updated a <a href="#">note</a> describing that DQ bit swapping within a nibble is supported in <a href="#">PolarFire Board Design Recommendations</a> and <a href="#">PolarFire SoC Board Design Recommendations</a>.</li> <li>• Updated a <a href="#">note</a> under <a href="#">Place and Route Pin-out-Rules</a>.</li> <li>• Added a <a href="#">note</a> regarding the "CL118" warning generated during the synthesis of PF_DDR4 memory controller. See <a href="#">Timing Constraints</a>.</li> <li>• Updated the example PDC command in <a href="#">Using PDC</a>.</li> </ul>
G	10/2024	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>• Updated <a href="#">Configuring the DDR3 Subsystem</a> as follows: <ul style="list-style-type: none"> <li>– Updated <a href="#">Figure 2.49</a> and <a href="#">Figure 2.51</a>.</li> <li>– Added <b>CK/CA additive offset</b> parameter information.</li> </ul> </li> <li>• Updated <a href="#">Configuring the LPDDR3 Subsystem</a> as follows: <ul style="list-style-type: none"> <li>– Updated <a href="#">Figure 2.56</a> and <a href="#">Figure 2.57</a>.</li> <li>– Updated <b>CK/CA additive offset</b> parameter related information.</li> </ul> </li> <li>• Updated <a href="#">Configuring the DDR4 Subsystem</a> as follows: <ul style="list-style-type: none"> <li>– Updated <a href="#">Figure 2.62</a> and <a href="#">Figure 2.63</a>.</li> <li>– Added <b>CK/CA additive offset</b> parameter information.</li> </ul> </li> </ul>

## Revision History (continued)

Revision	Date	Description
F	06/2024	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>• Added footnote 2 under <a href="#">Table 1</a>.</li> <li>• Changed the section title from “Training Logic for Embedded DDR Controller” to <a href="#">Training Logic for the PolarFire SoC MSS DDR Controller</a>.</li> <li>• Added <a href="#">Maximum Density Supported by the PolarFire SoC MSS DDR Controller</a>.</li> <li>• Updated the description of the DM/DM_N port in <a href="#">Table 2.15</a>.</li> <li>• Added <a href="#">note</a> regarding the tCCD_L timing parameter in <a href="#">Configuring the DDR4 Subsystem</a>.</li> <li>• Updated <a href="#">note</a> in <a href="#">Configuring the DDR4 Subsystem</a>.</li> <li>• Updated <a href="#">Figure 2.15</a> and changed the value of L_B_SIZE from 32 to 128.</li> <li>• Updated <a href="#">Figure 2.16</a>, changed the value of L_B_SIZE from 32 to 128, and removed the last bullet point “L_R_VALID is deasserted”.</li> <li>• Updated <a href="#">note</a> regarding the DQS value in Octal DDR PHY-Only solution.</li> <li>• Added <a href="#">note</a> about disabling the TMR option during Synthesis, see <a href="#">Physical Constraints</a>.</li> <li>• Removed the text “(for fabric LPDDR3 memory controller)” next to CMD/ADDR to REF_CLK training in <a href="#">Training Logic</a> because CK/CA additive offset field is applicable for all the other DDR types.</li> <li>• Revised the note about the DQ bits swapping in <a href="#">PolarFire Board Design Recommendations</a> and <a href="#">PolarFire SoC Board Design Recommendations</a>.</li> <li>• Changed CS, CKE, and ODT signals to single-bit and Bank Address to 3-bit (BA[2:0]) in <a href="#">Table 6.3</a>.</li> <li>• Removed the termination resistor value of “39_1%” in <a href="#">Figure 6.8</a>, and replaced the value with a symbol “(R<sub>T</sub>)”.</li> <li>• Removed the termination resistor value of “39_1%” in <a href="#">Figure 7.8</a>, and replaced the value with a symbol “(R<sub>T</sub>)”.</li> </ul>
E	06/2023	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>• Changed the document title to <i>PolarFire Family Memory Controller User Guide</i>.</li> <li>• Added RT PolarFire support.</li> <li>• Updated <a href="#">MSS DDR Memory Controller</a> and <a href="#">Hardware Probes</a> mentioning that both LPDDR4 and DDR4 training issues are debugged using <a href="#">Memory Log Analyzer</a>.</li> <li>• Added a note under <a href="#">Table 2.9</a> mentioning the data limit to be ensured during DDR4 memory transfers using the AXI interface.</li> <li>• Updated <a href="#">Octal DDR PHY-Only Solution</a> by correcting that the Octal DDR PHY configurator supports up to four ranks per interface instead of two ranks.</li> <li>• Updated the description of the I{INTERFACE#}R{RANK#}_{CLOCK}_DATA output port.</li> <li>• Added information regarding the default value in the description of the <b>Automatic ZQ Calibration Period (us)</b> parameter in the DDR3, LPDDR3, and DDR4 configurator’s <b>Memory Timing</b> tab. Also, added a note describing that this default value is based on the preset configuration. See <a href="#">Configuring the DDR3 Subsystem</a>, <a href="#">Configuring the LPDDR3 Subsystem</a>, and <a href="#">Configuring the DDR4 Subsystem</a>.</li> <li>• Added a note in the description of the RESET_N signal. See <a href="#">Table 5.3</a>.</li> </ul>

Revision History (continued)		
Revision	Date	Description
D	10/2022	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>Added information about LPDDR4 initialization sequence. See <a href="#">LPDDR4 (For PolarFire SoC FPGA Only)</a>.</li> <li>Added board design recommendation for LPDDR4. See <a href="#">LPDDR4</a>.</li> <li>Changed the QDR SRAM Address Width and SRAM_DATA_WIDTH naming as per the configurator. See <a href="#">Table 5.3</a>.</li> <li>Updated the QDR address width information, see <a href="#">Features</a>.</li> <li>In <a href="#">QDR Resource Utilization</a>, removed the note that mentioned about the fixed address width, data width, and burst size in QDR RTL - Fast simulation mode.</li> </ul>
C	08/2022	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>Added a note regarding SHIELD signals in <a href="#">Table 2.15</a>.</li> <li>Added CMD/ADDR to REF_CLK training information specific to LPDDR3 in <a href="#">Training Logic</a>.</li> <li>Added <a href="#">Configuring the LPDDR3 Subsystem</a> and <a href="#">Configuring the DDR4 Subsystem</a> to describe the PF_LPDDR3 and PF_DDR4 Configurator options separately.</li> <li>Updated QDR training information, see <a href="#">Training Logic</a>.</li> <li>Added <a href="#">QDR Resource Utilization</a>.</li> <li>Added information about the DQ bit swapping support, see <a href="#">PolarFire Board Design Recommendations</a> and <a href="#">PolarFire SoC Board Design Recommendations</a>.</li> <li>Added a note about DDR interface migration, see <a href="#">Appendix: Fabric DDR Placement Locations</a>.</li> </ul>
B	11/2021	<p>The following is a summary of the changes made in this revision:</p> <ul style="list-style-type: none"> <li>Updated the clock frequency value of PolarFire Octal DDR PHY IP from 200 MHz to 250 MHz. See <a href="#">Octal DDR PHY-Only Solution</a> and <a href="#">Table 2.25</a>.</li> <li>Updated the description of the output signal <b>DOFF_N</b> of the QDR Controller, see <a href="#">Port List</a>.</li> <li>Added RTL_Fast simulation information for QDR memory, see <a href="#">Configuring QDR Memory Controller IP</a> and <a href="#">RTL_Fast Simulation</a>.</li> <li>Updated <a href="#">Figure 2.43</a> to show the latest configuration parameter <b>Enable Differential DQS</b> and added this configuration parameter to <a href="#">Table 2.25</a>.</li> <li>Added the <code>ALERT_N</code> signal and updated the description of the <code>stat_ca_parity_error</code> signal in <a href="#">Table 2.8</a>.</li> <li>Updated the description of the <code>ALERT_N</code> signal in <a href="#">Table 2.15</a>.</li> <li>Added the information about the <b>Memory Format</b> and <b>Enable Parity/Alert</b> options in <a href="#">General</a>.</li> </ul>
A	08/2021	<p>The first publication of this document. This user guide was created by merging the following documents:</p> <ul style="list-style-type: none"> <li>UG0676: PolarFire FPGA Memory Controller User Guide</li> <li>UG0906: PolarFire SoC FPGA Memory Controller User Guide</li> </ul> <p>For more information, see <a href="#">Table 12-1</a> and <a href="#">Table 12-2</a> respectively.</p>

The following revision history table describes the changes that were implemented in the *UG0676: PolarFire FPGA Memory Controller User Guide* document. The changes are listed by revision.

**Note:** *UG0676: PolarFire FPGA Memory Controller User Guide* document is now obsolete and the information in the document has been migrated to *PolarFire® FPGA and PolarFire SoC FPGA Memory Controller User Guide*.

**Table** Revision History of UG0676: PolarFire FPGA Memory Controller User Guide

Revision	Date	Description
Revision 8.0	04/21	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>• Updated that the DDR Controller complies with the AXI3/4 protocol.</li> <li>• Added DDR4 Additional Controller Options.</li> <li>• Added DDR4 Optimization Guidelines.</li> <li>• Added the prerequisite of setting VDDI and VREF before placing the DDR subsystem.</li> <li>• Removed DQ bit swapping information because it is not supported.</li> <li>• Added information about the minimum operating frequency of LPDDR3.</li> <li>• Updated for Libero SoC v2021.1.</li> </ul>
Revision 7.0	10/20	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>• Added information about Octal DDR PHY-Only Solution.</li> <li>• Updated support for ECC at maximum width for various device variants.</li> <li>• Added brief information about SmartDebug capability of displaying DDR training data.</li> <li>• Updated for Libero SoC v12.5.</li> </ul>
Revision 6.0	05/20	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>• Updated the description of the QDR output port DOFF_N.</li> <li>• Updated the QDR Training information.</li> <li>• Updated QDR related information for Burst of 2.</li> </ul>
Revision 5.0	04/19	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>• Updated the document for Libero SoC v12.0.</li> <li>• Added simulated timing waveforms for write and read sequence with and without lookahead precharge.</li> <li>• Added simulated timing waveforms for sample multi-burst write and read sequence.</li> <li>• Added simulated timing waveforms for sample SDRAM write and read sequence.</li> <li>• Updated the information about the DQ bit swapping supported.</li> <li>• Updated the table that lists the Write Data to Write Strobe Mapping for QDR x36 mode.</li> <li>• Added tables that list the Write Data to Write Strobe Mapping for QDR x18 and x9 modes.</li> <li>• Updated “Burst of 2 local interface timing” and “Burst of 4 local interface timing” diagrams in QDR Memory Controller.</li> <li>• Updated the QDR simulation options.</li> <li>• Added information about BFM simulation.</li> <li>• Added L_D_REQ_LAST_P0 and L_R_VALID_LAST_P0 signals in “Native Interface Signals” table.</li> </ul>
Revision 4.0	09/18	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>• The document was updated for Libero SoC PolarFire v2.3.</li> <li>• Added information about the DDR PHY-only solution.</li> <li>• Removed the VREFDQ leveling step from the initialization sequence of DDR3 and DDR4.</li> </ul>

**Table** Revision History of UG0676: PolarFire FPGA Memory Controller User Guide (continued)

Revision	Date	Description
Revision 3.0	04/18	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>Updated the title of the document and added the QDR memory controller information. Now, both DDR and QDR information is available in this document.</li> <li>Updated the Libero SoC PolarFire version to v2.1 throughout the document.</li> <li>Updated the DDR3 and DDR4 resource utilization data.</li> <li>Updated the information about connectivity between DDR3 and PolarFire devices.</li> <li>Updated the information about the native interface signals L_DATAIN[N:0], L_DATAOUT[N:0], and L_DM_IN[N:0].</li> <li>Updated the data widths and I/O standard information for various types of memories.</li> <li>Added the information about the simulation options available in the DDR3 configurator.</li> <li>Updated the DDR3 and DDR4 parameters for PolarFire devices.</li> <li>Updated the information about the standard LPDDR3 and DDR3 device configurations.</li> <li>Added the DDR3, LPDDR3, and DDR4 layout guidelines.</li> <li>Added the list of DDR placement location for each device package.</li> <li>Removed the instances that mention the LPDDR2 support throughout the document.</li> </ul>
Revision 2.0	06/17	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>Information about DDR memory interface data lanes was updated.</li> <li>Information about training logic was updated.</li> <li>Information about diagnostic options was updated.</li> <li>Information about physical constraints of the DDR subsystem was updated.</li> <li>Information about accessing DDR Subsystem using the native interface was updated.</li> </ul>
Revision 1.0	02/17	The first publication of UG0676: PolarFire FPGA Memory Controller User Guide.

The following revision history table describes the changes that were implemented in the *UG0906: PolarFire SoC FPGA Memory Controller User Guide* document. The changes are listed by revision.

**Note:** *UG0906: PolarFire SoC FPGA Memory Controller User Guide* document is now obsolete and the information in the document has been migrated to *PolarFire® FPGA and PolarFire SoC FPGA Memory Controller User Guide*.

**Table** Revision History of UG0906: PolarFire SoC FPGA Memory Controller User Guide

Revision	Date	Description
Revision 3.0	04/21	<p>The following is a summary of the changes made in this revision.</p> <ul style="list-style-type: none"> <li>Updated that DDR Controller complies with the AXI3/4 protocol.</li> <li>Added DDR4 Additional Controller Options.</li> <li>Added DDR4 Optimization Guidelines.</li> <li>Added the prerequisite of setting VDDI and VREF before placing the DDR subsystem.</li> <li>Removed the DQ bit swapping information because it is not supported.</li> <li>Updated for Libero SoC v2021.1.</li> </ul>
Revision 2.0	09/20	Updated for Libero SoC v12.5.

**Table** Revision History of UG0906: PolarFire SoC FPGA Memory Controller User Guide (continued)

Revision	Date	Description
Revision 1.0	04/20	The first publication of the UG0906: PolarFire SoC FPGA Memory Controller User Guide.

## Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at [www.microchip.com/support](http://www.microchip.com/support). Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

## Microchip Information

### Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>.

ISBN: 979-8-3371-1289-3

### Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at [www.microchip.com/en-us/support/design-help/client-support-services](http://www.microchip.com/en-us/support/design-help/client-support-services).

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.

## Microchip Devices Code Protection Feature

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

- 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, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- 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. Microchip is committed to continuously improving the code protection features of our products.