External Memory Interfaces Agilex™ 7 M-Series FPGA IP User Guide

Updated for Quartus® Prime Design Suite: **24.1**

IP Version: **6.1.0**

Online Version
Send Feedback

772538
2024.04.01

Contents

1. About the External Memory Interfaces Agilex™ 7 M-Series FPGA IP

1.1. Release Information

IP versions are the same as the Quartus® Prime Design Suite software versions up to v19.1. From Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

The IP version (X.Y.Z) number may change from one Quartus Prime software version to another. A change in:

- X indicates a major revision of the IP. If you update your Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 1. Release Information

Item	Description
IP Version	6.1.0
Quartus Prime	24.1
Release Date	2024.04.01

Note: This documentation is preliminary and subject to change.

Related Information

- External Memory Interfaces Agilex™ 7 M-Series FPGA IP Design Example User Guide
  Design example and quick-start information for the Agilex™ 7 M-Series EMIF IP, which provides external memory interface support for the DDR4, DDR5, and LPDDR5 memory protocols.
- External Memory Interfaces Agilex™ 7 M-Series FPGA IP Release Notes
  Release notes for the Agilex™ 7 M-Series EMIF IP.

2. Agilex™ 7 M-Series FPGA EMIF IP – Introduction

Intel's fast, efficient, and low-latency external memory interface (EMIF) intellectual property (IP) cores interface with today's higher speed memory devices.

You can implement the EMIF IP core functions through the Quartus Prime software.

The *External Memory Interfaces Agilex™ 7 M-Series FPGA IP* (referred to hereafter as the *Agilex 7 M-Series EMIF IP*) provides the following components:

- A physical layer interface (PHY) which builds the data path and manages timing transfers between the FPGA and the memory device.
- A memory controller which implements all the memory commands and protocol-level requirements.

For information on the maximum speeds supported by the external memory interface IP, refer to the *External Memory Interface Spec Estimator*, available here: https://www.intel.com/content/www/us/en/programmable/support/support-resources/support-centers/external-memory-interfaces-support/emif.html.

2.1. Agilex 7 M-Series EMIF IP Protocol and Feature Support

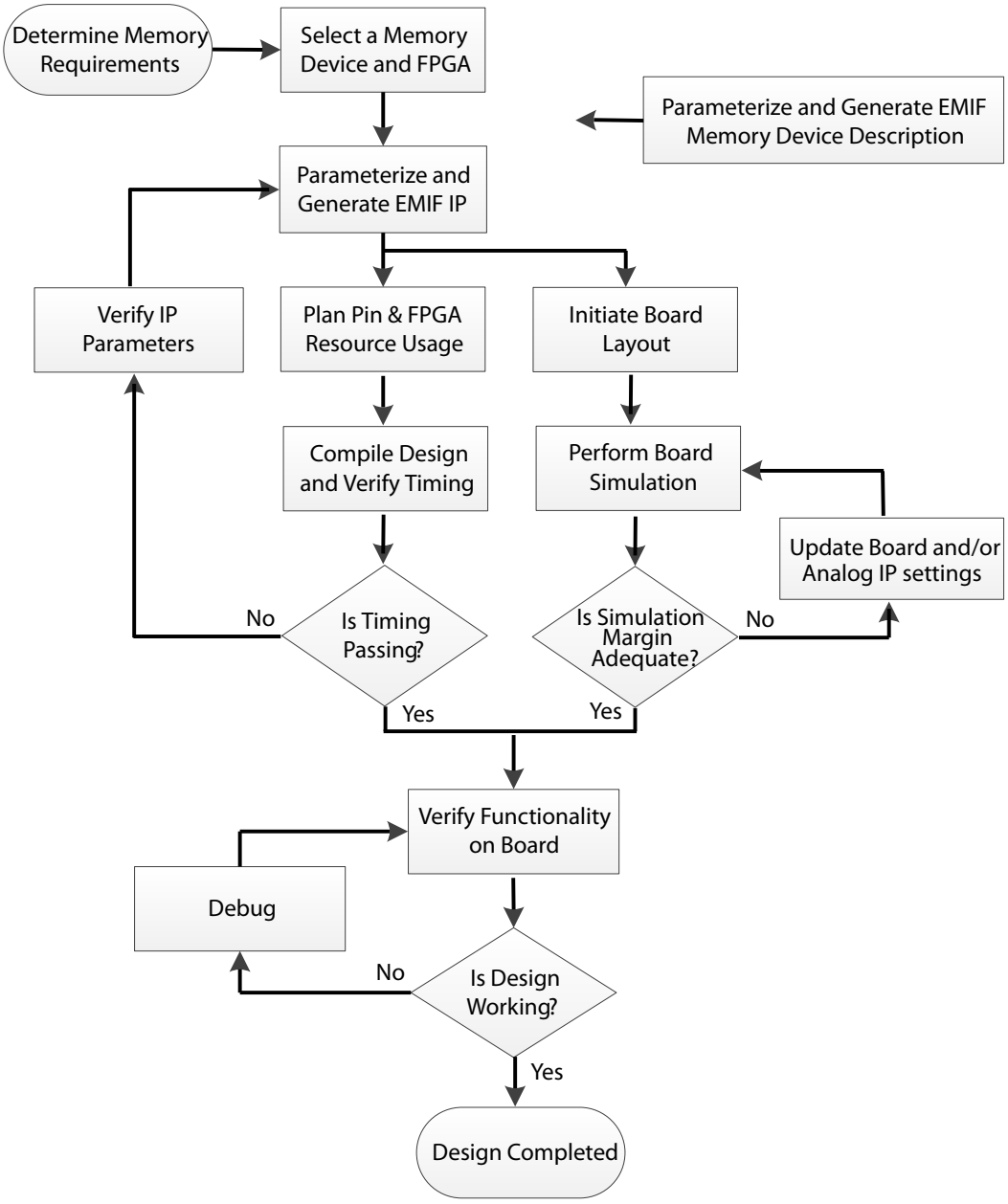
- The Agilex 7 M-Series FPGA EMIF IP supports DDR4 with hard memory controller and hard PHY.
- The Agilex 7 M-Series FPGA EMIF IP supports DDR5 with hard memory controller and hard PHY.
- The Agilex 7 M-Series FPGA EMIF IP supports LPDDR5 with hard memory controller and hard PHY.

2.2. Agilex 7 M-Series EMIF IP Design Flow

Intel recommends creating an example top-level file with the desired pin outs and all interface IPs instantiated. This enables the Quartus Prime software to validate the design and resource allocation before PCB and schematic sign off.

The following figure shows the design flow to provide the fastest out-of-the-box experience with the EMIF IP.

Figure 1. EMIF IP Design Flow



2.3. Agilex 7 M-Series EMIF IP Design Checklist

Refer to the following checklist as a quick reference for information about steps in the EMIF design flow.

Table 2. EMIF Design Checklist

Design Step	Description	Resources
Select an FPGA	Not all Intel FPGAs support all memory types and configurations. To help with the FPGA selection process, refer to the resources listed in the right column.	• External Memory Interfaces Support Center • External Memory Interface Spec Estimator
Parameterize the IP	Correct IP parameterization is important for good EMIF IP operation. The resources listed in the right column define the memory parameters during IP generation.	• DDR4 Parameter Descriptions • DDR5 Parameter Descriptions • LPDDR5 Parameter Descriptions
Generate initial IP and example design	After you have parameterized the EMIF IP, you can generate the IP, along with an optional example design. Refer to the Quick-Start Guide for a walkthrough of this process.	• External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide
Perform functional simulation	Simulation of the EMIF design helps to determine correct operation. The resources listed in the right column explain how to perform simulation and what differences exist between simulation and hardware implementation.	• External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide • Simulating Memory IP
Make pin assignments	For guidance on pin placement, refer to the resources listed in the right column.	• DDR4 Parameter Descriptions • DDR5 Parameter Descriptions • LPDDR5 Parameter Descriptions • Device Pin Tables
Perform board simulation	Board simulation helps determine optimal settings for signal integrity, drive strength, as well as sufficient timing margins and eye openings. For guidance on board simulation, refer to the resources listed in the right column.	• Board Design Guidelines • Timing Closure
Verify timing closure	For information regarding compilation, system-level timing closure and timing reports refer to the Timing Closure section of this User Guide.	• Timing Closure
Run the design on hardware	For instructions on how to program a FPGA refer to the Quick-Start section of the Design Example User Guide.	• External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide
Debug issues with preceding steps	Operational problems can generally be attributed to one of the following: interface configuration, pin/resource planning, signal integrity, or timing. The resources listed in the right column contain information on typical debug procedures and available tools to help diagnose hardware issues.	• Debugging • External Memory Interfaces Support Center

3. Agilex 7 M-Series FPGA EMIF IP – Product Architecture

This chapter describes the Agilex 7 M-Series FPGA EMIF IP product architecture.

3.1. Agilex 7 M-Series EMIF Architecture: Introduction

The Agilex 7 M-Series EMIF architecture contains many new hardware features designed to meet the high-speed requirements of emerging memory protocols, while consuming the smallest amount of core logic area and power.

Note: The current version of the External Memory Interfaces Agilex 7 M-Series FPGA IP supports the DDR4, DDR5, and LPDDR5 memory protocols.

The following are key hardware features of the Agilex 7 M-Series EMIF architecture:

Hard Sequencer

The sequencer employs a hardened processor, and can perform memory calibration for a wide range of protocols. For Agilex 7 M-Series devices, the sequencer and calibration are localized to each I/O bank.

Note: You cannot use the hardened processor for any user applications after calibration is complete.

Hard PHY

The PHY circuitry in Agilex 7 M-Series devices is hardened in the silicon, which simplifies the challenges of achieving timing closure and minimizing power consumption.

Hard Memory Controller

The hard memory controller reduces latency and minimizes core logic consumption in the external memory interface. The hard memory controller supports the DDR4, DDR5, and LPDDR5 memory protocols.

High-Speed PHY Clock Tree

Dedicated high speed PHY clock networks clock the I/O buffers in Agilex 7 M-Series EMIF IP. The PHY clock trees exhibit low jitter and low duty cycle distortion, maximizing the data valid window.

Automatic Clock Phase Alignment

Automatic clock phase alignment circuitry dynamically adjusts the clock phase of core clock networks to match the clock phase of the PHY clock networks. The clock phase alignment circuitry minimizes clock skew that can complicate timing closure in transfers between the FPGA core and the periphery.

Network-on-Chip (NoC) Interface

The Agilex 7 M-Series EMIF IP supports a new Network-on-Chip (NoC) interface. Each IO96 bank contains two 256-bit AXI4 targets and one 32-bit AXI4-Lite target that connect to the NoC. NoC segments span one FPGA clock sector and consists of three AXI4 initiators on the FPGA fabric side. A network of switches transfer packets horizontally across the high-speed interconnect NoC and connect the initiators and targets. Refer to the *NoC User Guide* for additional information.

3.1.1. Agilex 7 M-Series EMIF Architecture: I/O Subsystem

In Agilex 7 M-Series devices, the I/O subsystem consists of two rows at the edge of the core.

The I/O subsystem provides the following features:

- General-purpose I/O registers and I/O buffers
- Compensation Block (Comp block)
  - On-chip termination control (OCT)
- I/O PLLs
  - I/O Bank I/O PLL for external memory interfaces and user logic
  - Fabric-feeding for non-EMIF/non-LVDS SERDES IP applications
- True differential signaling
- External memory interface components, as follows:
  - A Primary hard memory controller, which has connectivity to 8 lanes (up to 4 byte lanes for data, and optionally one additional lane for out-of-band ECC data)
  - A Secondary hard memory controller, which has connectivity to 4 lanes (up to 2 byte lanes for data)
  - Hard PHY
  - Hardened processor and calibration logic
  - DLL

3.1.2. Agilex 7 M-Series EMIF Architecture: I/O SSM

Each I/O bank includes one I/O subsystem manager (I/O SSM), which contains a hardened processor with dedicated memory. The I/O SSM is responsible for calibration of all the EMIFs in the I/O bank.

The I/O SSM includes dedicated memory which stores both the calibration algorithm and calibration run-time data. The hardened processor and the dedicated memory can be used only by an external memory interface, and cannot be employed for any other use.

The on-chip configuration network clocks the I/O SSM, and therefore the I/O SSM does not consume a PLL.

Each EMIF instance must be connected to the I/O SSM through the External Memory Interfaces Calibration IP. The Calibration IP exposes a calibration bus master port, which must be connected to the slave calibration bus port on every EMIF instance.

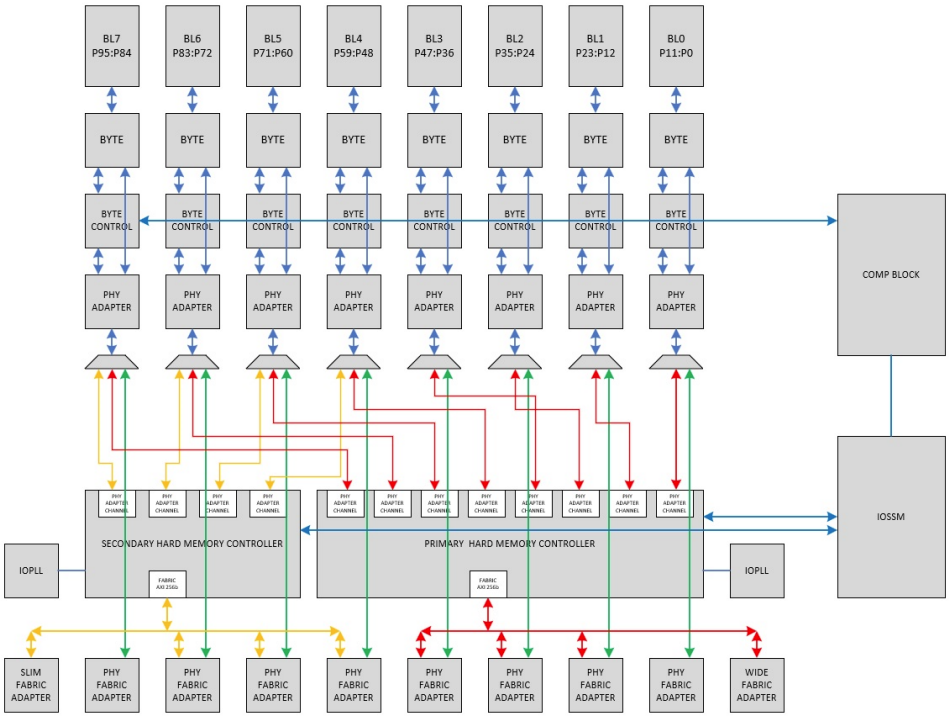
3.1.3. Agilex 7 M-Series EMIF Architecture: I/O Bank

Each I/O row contains up to four I/O banks; the exact number of banks depends on device size and pin package.

Each I/O bank consists of two sub-banks, and each sub-bank contains the following components:

- I/O PLL and PHY clock trees
- DLL
- Input DQS clock trees
- 48 pins, organized into four I/O lanes of 12 pins each

Figure 3. I/O Bank Architecture in Agilex 7 M-Series Devices



Within an I/O bank, the top sub-bank is pin indexes P95:P48, and the bottom sub-bank is pin indexes P47:P0.

Agilex 7 M-Series devices have two hard memory controllers: primary and secondary. The primary hard memory controller has access to all 96 pins in an I/O bank. The secondary hard memory controller has access only to the top sub-bank. In the above figure, the yellow signals highlight the connections for the secondary hard memory controller, while the red signals show the connections for the primary hard memory controller. The green signals show where both hard memory controllers are bypassed to provide access to the PHY from the core logic.

3.1.3.1. Lockstep Configuration

To support user data widths greater than 32 bits in DDR4, the external memory interface (EMIF) IP instantiates multiple memory controllers driven in lockstep.

The primary controller drives the address and command bus and 32-bits of the DQ bus, while the other controllers drive the remainder of the DQ bus.

The following table summarizes the supported lockstep configurations:

Table 3. Supported Lockstep Controller Configurations

Memory Protocol	Configuration	DQ Width	AXI Interface
DDR4	x40	40	256 b + 64 b USER DATA
DDR4	x64	64	512 b
DDR4	x64 with ECC [1]	72	512 b
DDR4	x72	72	512 b + 64 b USER DATA

To ensure that the controllers remain coordinated in lockstep, the following points apply:

- Lockstep configurations support only synchronous fabric clocking mode. NoC Mode is not supported.
- Some of the controller scheduling and optimization features are disabled.
- There are limitations on the types of AXI transactions supported. Refer to the following table for details.

Table 4. Limitations for Lockstep Configuration in ES Devices

Memory Protocol	Configuration	Limitation
DDR4	x72	Supports AXI transfer size= 6 only.
DDR4	x64	AXI transfer size min = 1* (2 bytes).
DDR4	x64	Supports 2-byte aligned transfers only.
DDR4	x40	Supports AXI transfer size= 5 only.
DDR4	x40	Support 32-bit aligned transfers only.

Two Controllers in Lockstep Within One IO96 Bank

The EMIF IP instantiates 2 controllers within one IO96 bank to support x40 configurations. The AXI bus is configured as 256-bits wide plus 64-bits of user data (WUSER/RUSER), to generate the required 320-bits of data to transfer a burst-of-8 of 40-bit DQ. The following table illustrates how the AXI WDATA/RDATA and WUSER/RUSER can be mapped to the DQ lanes.

Table 5. Mapping of WDATA/RDATA & WUSER/RUSER in x40 Configuration

Transfer	0	1	2	3	4	5	6	7
WDATA/ RDATA	31:0	63:32	95:64	127:96	159:128	191:160	223:192	255:224
DQ	[31:0]							
WUSER/ RUSER	7:0	15:8	23:16	31:24	39:32	47:40	55:48	63:56
DQ	[39:32]							

This configuration can support only 3 address/command lane configurations, because there are only 8 byte lanes in one IO96 bank. The WUSER/RUSER signal is mapped to byte lane 7 (DQ lane with prefix *s*) in x40 configuration.

Table 6. Supported Byte Lane Placement for x40 configuration

Scheme	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
DDR4_AC_TOP	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]
DDR4_AC_BOT	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	sDQ[4]

Two/Three Controllers in Lockstep Within One IO96 Bank

This configuration is supported only in DDR4. The EMIF IP instantiates two or three controllers across two adjacent IO96 banks for the configurations listed in the table below.

Table 7. Supported 2/3 Controller Configurations in Lockstep

Configuration	DQ Width	AXI Interface	Notes
x64	64	512b	2 controllers used.
x64 with ECC	72	512b	3 controllers used. ECC will be generated and checked by a soft IP block within the EMIF IP.
x72	72	512b + 64b USER DATA	3 controllers used.

For the x72 configuration, the AXI bus is configured as 512-bits wide, plus 64-bits of user data (WUSER/RUSER). The following example illustrates how you can map the AXI WDATA/RDATA and WUSER/RUSER to the DQ lanes. In this illustration, the WUSER/RUSER is mapped to the byte lane used for DQ [71:64].

The actual DQ lane to which the WUSER/RUSER is mapped depends on the address and command placement used. In each supported address and command placement scheme, the WUSER/RUSER is mapped to the DQ lane that has a prefix *s* (for example, sDQ0, sDQ4 or sDQ8). Refer to the following tables in the *DDR4 Data Width Mapping* topic, to identify the actual DQ lane used for WUSER/RUSER:

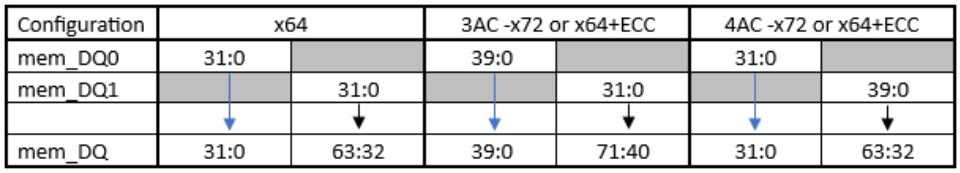
- *Supported Lockstep configuration for DDR4 x64*
- *Supported Lockstep configuration for DDR4 x72 or x64 (with ECC)*

Table 8. Example Mapping of WDATA/RDATA & WUSER/RUSER in x72 Configuration

Transfer	0	1	2	3	4	5	6	7
WDATA/RDATA	63:0	127:64	191:128	255:192	319:256	383:320	447:384	511:448
DQ	[63:0]							
WUSER/RUSER	7:0	15:8	23:16	31:24	39:32	47:40	55:48	63:56
DQ	[71:64]							

The generated IP has 2 mem DQ ports, collectively labeled as `mem_DQ0` and `mem_DQ1`. The mapping below shows how these 2 ports are grouped into one single wide DQ port:

Figure 4.



3.1.3.2. DDR4 Pin Placement

Table 9. DDR4 Pin Placement

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
BL7	95	MEM_DQ[39]*			
	94	MEM_DQ[38] *			
	93	MEM_DQ[37] *			
	92	MEM_DQ[36] *			
	91				
	90	MEM_DM_N[4]			
	89	MEM_DQS_C[4]			
	88	MEM_DQS_T[4]			
	87	MEM_DQ[35] *			
	86	MEM_DQ[34] *			
	85	MEM_DQ[33] *			
	84	MEM_DQ[32] *			
BL6	83	MEM_DQ[31]	MEM_DQ[31]		
	82	MEM_DQ[30]	MEM_DQ[30]		
	81	MEM_DQ[29]	MEM_DQ[29]		
	80	MEM_DQ[28]	MEM_DQ[28]		
	79				
	78	MEM_DM_N[3]	MEM_DM_N[3]		
	77	MEM_DQS_C[3]	MEM_DQS_C[3]		
	76	MEM_DQS_T[3]	MEM_DQS_T[3]		
	75	MEM_DQ[27]	MEM_DQ[27]		
	74	MEM_DQ[26]	MEM_DQ[26]		
	73	MEM_DQ[25]	MEM_DQ[25]		
	72	MEM_DQ[24]	MEM_DQ[24]		

continued...

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
BL5	71	MEM_DQ[23]	MEM_DQ[23]	MEM_DQ[23] *	
	70	MEM_DQ[22]	MEM_DQ[22]	MEM_DQ[22] *	
	69	MEM_DQ[21]	MEM_DQ[21]	MEM_DQ[21] *	
	68	MEM_DQ[20]	MEM_DQ[20]	MEM_DQ[20] *	
	67				
	66	MEM_DM_N[2]	MEM_DM_N[2]	MEM_DM_N[2]	
	65	MEM_DQS_C[2]	MEM_DQS_C[2]	MEM_DQS_C[2]	
	64	MEM_DQS_T[2]	MEM_DQS_T[2]	MEM_DQS_T[2]	
	63	MEM_DQ[19]	MEM_DQ[19]	MEM_DQ[19] *	
	62	MEM_DQ[18]	MEM_DQ[18]	MEM_DQ[18] *	
	61	MEM_DQ[17]	MEM_DQ[17]	MEM_DQ[17] *	
	60	MEM_DQ[16]	MEM_DQ[16]	MEM_DQ[16] *	
BL4	59	MEM_DQ[15]	MEM_DQ[15]	MEM_DQ[15]	MEM_DQ[15]
	58	MEM_DQ[14]	MEM_DQ[14]	MEM_DQ[14]	MEM_DQ[14]
	57	MEM_DQ[13]	MEM_DQ[13]	MEM_DQ[13]	MEM_DQ[13]
	56	MEM_DQ[12]	MEM_DQ[12]	MEM_DQ[12]	MEM_DQ[12]
	55				
	54	MEM_DM_N[1]	MEM_DM_N[1]	MEM_DM_N[1]	MEM_DM_N[1]
	53	MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_DQS_C[1]
	52	MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_DQS_T[1]
	51	MEM_DQ[11]	MEM_DQ[11]	MEM_DQ[11]	MEM_DQ[11]
	50	MEM_DQ[10]	MEM_DQ[10]	MEM_DQ[10]	MEM_DQ[10]
	49	MEM_DQ[9]	MEM_DQ[9]	MEM_DQ[9]	MEM_DQ[9]
	48	MEM_DQ[8]	MEM_DQ[8]	MEM_DQ[8]	MEM_DQ[8]
BL3	47	MEM_BG[0]	MEM_BG[0]	MEM_BG[0]	MEM_BG[0]
	46	MEM_BA[1]	MEM_BA[1]	MEM_BA[1]	MEM_BA[1]
	45	MEM_BA[0]	MEM_BA[0]	MEM_BA[0]	MEM_BA[0]
	44	MEM_ALERT_N[0]	MEM_ALERT_N[0]	MEM_ALERT_N[0]	MEM_ALERT_N[0]
	43	MEM_A[16]	MEM_A[16]	MEM_A[16]	MEM_A[16]
	42	MEM_A[15]	MEM_A[15]	MEM_A[15]	MEM_A[15]
	41	MEM_A[14]	MEM_A[14]	MEM_A[14]	MEM_A[14]
	40	MEM_A[13]	MEM_A[13]	MEM_A[13]	MEM_A[13]
	39	MEM_A[12]	MEM_A[12]	MEM_A[12]	MEM_A[12]
	38	RZQ Site	RZQ Site	RZQ Site	RZQ Site
	37	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site

continued...

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
	36	Differential "P-Side" Reference Clock Input Site	Differential "P-Side" Reference Clock Input Site	Differential "P-Side" Reference Clock Input Site	Differential "P-Side" Reference Clock Input Site
BL2	35	MEM_A[11]	MEM_A[11]	MEM_A[11]	MEM_A[11]
	34	MEM_A[10]	MEM_A[10]	MEM_A[10]	MEM_A[10]
	33	MEM_A[9]	MEM_A[9]	MEM_A[9]	MEM_A[9]
	32	MEM_A[8]	MEM_A[8]	MEM_A[8]	MEM_A[8]
	31	MEM_A[7]	MEM_A[7]	MEM_A[7]	MEM_A[7]
	30	MEM_A[6]	MEM_A[6]	MEM_A[6]	MEM_A[6]
	29	MEM_A[5]	MEM_A[5]	MEM_A[5]	MEM_A[5]
	28	MEM_A[4]	MEM_A[4]	MEM_A[4]	MEM_A[4]
	27	MEM_A[3]	MEM_A[3]	MEM_A[3]	MEM_A[3]
	26	MEM_A[2]	MEM_A[2]	MEM_A[2]	MEM_A[2]
	25	MEM_A[1]	MEM_A[1]	MEM_A[1]	MEM_A[1]
	24	MEM_A[0]	MEM_A[0]	MEM_A[0]	MEM_A[0]
BL1	23	MEM_PAR[0]	MEM_PAR[0]	MEM_PAR[0]	MEM_PAR[0]
	22	MEM_CS_N[1]	MEM_CS_N[1]	MEM_CS_N[1]	MEM_CS_N[1]
	21	MEM_CK_C[0]	MEM_CK_C[0]	MEM_CK_C[0]	MEM_CK_C[0]
	20	MEM_CK_T[0]	MEM_CK_T[0]	MEM_CK_T[0]	MEM_CK_T[0]
	19	MEM_CKE[1]	MEM_CKE[1]	MEM_CKE[1]	MEM_CKE[1]
	18	MEM_CKE[0]	MEM_CKE[0]	MEM_CKE[0]	MEM_CKE[0]
	17	MEM_ODT[1]	MEM_ODT[1]	MEM_ODT[1]	MEM_ODT[1]
	16	MEM_ODT[0]	MEM_ODT[0]	MEM_ODT[0]	MEM_ODT[0]
	15	MEM_ACT_N[0]	MEM_ACT_N[0]	MEM_ACT_N[0]	MEM_ACT_N[0]
	14	MEN_CS_N[0]	MEN_CS_N[0]	MEN_CS_N[0]	MEN_CS_N[0]
	13	MEM_RESET_N[0]	MEM_RESET_N[0]	MEM_RESET_N[0]	MEM_RESET_N[0]
	12	MEM_BG[1]	MEM_BG[1]	MEM_BG[1]	MEM_BG[1]
BL0	11	MEM_DQ[7]	MEM_DQ[7]	MEM_DQ[7]	MEM_DQ[7]
	10	MEM_DQ[6]	MEM_DQ[6]	MEM_DQ[6]	MEM_DQ[6]
	9	MEM_DQ[5]	MEM_DQ[5]	MEM_DQ[5]	MEM_DQ[5]
	8	MEM_DQ[4]	MEM_DQ[4]	MEM_DQ[4]	MEM_DQ[4]
	7				
	6	MEM_DM_N[0]	MEM_DM_N[0]	MEM_DM_N[0]	MEM_DM_N[0]
	5	MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_DQS_C[0]
	4	MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_DQS_T[0]
	3	MEM_DQ[3]	MEM_DQ[3]	MEM_DQ[3]	MEM_DQ[3]

continued...

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
	2	MEM_DQ[2]	MEM_DQ[2]	MEM_DQ[2]	MEM_DQ[2]
	1	MEM_DQ[1]	MEM_DQ[1]	MEM_DQ[1]	MEM_DQ[1]
	0	MEM_DQ[0]	MEM_DQ[0]	MEM_DQ[0]	MEM_DQ[0]

Note: The presence of an asterisk (*) in the above table indicates an ECC byte location.

3.1.3.3. DDR5 Pin Placement

Table 10. DDR5 Pin Placement

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
BL7	95			MEM_1_MEM_DQ[15]		
	94			MEM_1_MEM_DQ[14]		
	93			MEM_1_MEM_DQ[13]		
	92			MEM_1_MEM_DQ[12]		
	91					
	90			MEM_1_MEM_DM_N[1]		
	89			MEM_1_MEM_DQS_C[1]		
	88			MEM_1_MEM_DQS_T[1]		
	87			MEM_1_MEM_DQ[11]		
	86			MEM_1_MEM_DQ[10]		
	85			MEM_1_MEM_DQ[9]		
	84			MEM_1_MEM_DQ[8]		
BL6	83	MEM_DQ[39]*		MEM_1_MEM_DQ[7]		
	82	MEM_DQ[38]*		MEM_1_MEM_DQ[6]		
	81	MEM_DQ[37]*		MEM_1_MEM_DQ[5]		
	80	MEM_DQ[36]*		MEM_1_MEM_DQ[4]		
	79					
	78	MEM_DM_N[4]		MEM_1_MEM_DM_N[0]		

continued...

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	77	MEM_DQS_C[4]		MEM_1_MEM_DQS_C[0]		
	76	MEM_DQS_T[4]		MEM_1_MEM_DQS_T[0]		
	75	MEM_DQ[35]*		MEM_1_MEM_DQ[3]		
	74	MEM_DQ[34]*		MEM_1_MEM_DQ[2]		
	73	MEM_DQ[33]*		MEM_1_MEM_DQ[1]		
	72	MEM_DQ[32]*		MEM_1_MEM_DQ[0]		
BL5	71	MEM_DQ[31]	MEM_DQ[31]	MEM_1_CK_C[1]		
	70	MEM_DQ[30]	MEM_DQ[30]	MEM_1_CK_T[1]		
	69	MEM_DQ[29]	MEM_DQ[29]	MEM_1_MEM_CS_N[0]		
	68	MEM_DQ[28]	MEM_DQ[28]	MEM_1_MEM_CS_N[1]		
	67			MEM_1_CK_C[0]		
	66	MEM_DM_N[3]	MEM_DM_N[3]	MEM_1_CK_T[0]		
	65	MEM_DQS_C[3]	MEM_DQS_C[3]	MEM_1_MEM_CA[12]		
	64	MEM_DQS_T[3]	MEM_DQS_T[3]	MEM_1_MEM_CA[11]		
	63	MEM_DQ[27]	MEM_DQ[27]	MEM_1_RESET_N		
	62	MEM_DQ[26]	MEM_DQ[26]	OCT_1_OCT_RZQIN		
	61	MEM_DQ[25]	MEM_DQ[25]	MEM_1_ALERT_N		
	60	MEM_DQ[24]	MEM_DQ[24]	MEM_1_MEM_CA[10]		
BL4	59	MEM_DQ[23]	MEM_DQ[23]	Differential "NSide" Reference Clock Input Site	MEM_DQ[23]*	
	58	MEM_DQ[22]	MEM_DQ[22]	Differential "PSide" Reference Clock Input Site	MEM_DQ[22]*	
	57	MEM_DQ[21]	MEM_DQ[21]	MEM_1_MEM_CA[9]	MEM_DQ[21]*	

continued...

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	56	MEM_DQ[20]	MEM_DQ[20]	MEM_1_MEM_CA[8]	MEM_DQ[20]*	
	55			MEM_1_MEM_CA[7]		
	54	MEM_DM_N[2]	MEM_DM_N[2]	MEM_1_MEM_CA[6]	MEM_DM_N[2]	
	53	MEM_DQS_C[2]	MEM_DQS_C[2]	MEM_1_MEM_CA[5]	MEM_DQS_C[2]	
	52	MEM_DQS_T[2]	MEM_DQS_T[2]	MEM_1_MEM_CA[4]	MEM_DQS_T[2]	
	51	MEM_DQ[19]	MEM_DQ[19]	MEM_1_MEM_CA[3]	MEM_DQ[19]*	
	50	MEM_DQ[18]	MEM_DQ[18]	MEM_1_MEM_CA[2]	MEM_DQ[18]*	
	49	MEM_DQ[17]	MEM_DQ[17]	MEM_1_MEM_CA[1]	MEM_DQ[17]*	
	48	MEM_DQ[16]	MEM_DQ[16]	MEM_1_MEM_CA[0]	MEM_DQ[16]*	
BL3	47	MEM_CK_C[1]	MEM_CK_C[1]	MEM_0_CK_C[1]	MEM_CK_C[1]	MEM_CK_C[1]
	46	MEM_CK_T[1]	MEM_CK_T[1]	MEM_0_CK_T[1]	MEM_CK_T[1]	MEM_CK_T[1]
	45	MEM_CS_N[0]	MEM_CS_N[0]	MEM_0_MEM_CS_N[0]	MEM_CS_N[0]	MEM_CS_N[0]
	44	MEM_CS_N[1]	MEM_CS_N[1]	MEM_0_MEM_CS_N[1]	MEM_CS_N[1]	MEM_CS_N[1]
	43	MEM_CK_C[0]	MEM_CK_C[0]	MEM_0_CK_C[0]	MEM_CK_C[0]	MEM_CK_C[0]
	42	MEM_CK_T[0]	MEM_CK_T[0]	MEM_0_CK_T[0]	MEM_CK_T[0]	MEM_CK_T[0]
	41	MEM_CA[12]	MEM_CA[12]	MEM_0_MEM_CA[12]	MEM_CA[12]	MEM_CA[12]
	40	MEM_CA[11]	MEM_CA[11]	MEM_0_MEM_CA[11]	MEM_CA[11]	MEM_CA[11]
	39	MEM_RESET_N[0]	MEM_RESET_N[0]	MEM_0_RESET_N	MEM_RESET_N[0]	MEM_RESET_N[0]
	38	RZQ Site	RZQ Site	OCT_0_OCT_RZQIN	RZQ Site	RZQ Site
	37	MEM_ALERT_N[0]	MEM_ALERT_N[0]	MEM_0_ALERT_N	MEM_ALERT_N[0]	MEM_ALERT_N[0]
	36	MEM_CA[10]	MEM_CA[10]	MEM_0_MEM_CA[10]	MEM_CA[10]	MEM_CA[10]
BL2	35	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "NSide" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site

continued...

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	34	Differential "P-Side" Reference Clock Input Site	Differential "P-Side" Reference Clock Input Site	Differential "PSide" Reference Clock Input Site	Differential "P-Side" Reference Clock Input Site	Differential "P-Side" Reference Clock Input Site
	33	MEM_CA[9]	MEM_CA[9]	MEM_0_MEM_CA[9]	MEM_CA[9]	MEM_CA[9]
	32	MEM_CA[8]	MEM_CA[8]	MEM_0_MEM_CA[8]	MEM_CA[8]	MEM_CA[8]
	31	MEM_CA[7]	MEM_CA[7]	MEM_0_MEM_CA[7]	MEM_CA[7]	MEM_CA[7]
	30	MEM_CA[6]	MEM_CA[6]	MEM_0_MEM_CA[6]	MEM_CA[6]	MEM_CA[6]
	29	MEM_CA[5]	MEM_CA[5]	MEM_0_MEM_CA[5]	MEM_CA[5]	MEM_CA[5]
	28	MEM_CA[4]	MEM_CA[4]	MEM_0_MEM_CA[4]	MEM_CA[4]	MEM_CA[4]
	27	MEM_CA[3]	MEM_CA[3]	MEM_0_MEM_CA[3]	MEM_CA[3]	MEM_CA[3]
	26	MEM_CA[2]	MEM_CA[2]	MEM_0_MEM_CA[2]	MEM_CA[2]	MEM_CA[2]
	25	MEM_CA[1]	MEM_CA[1]	MEM_0_MEM_CA[1]	MEM_CA[1]	MEM_CA[1]
	24	MEM_CA[0]	MEM_CA[0]	MEM_0_MEM_CA[0]	MEM_CA[0]	MEM_CA[0]
BL1	23	MEM_DQ[7]	MEM_DQ[7]	MEM_0_MEM_DQ[7]	MEM_DQ[7]	MEM_DQ[7]
	22	MEM_DQ[6]	MEM_DQ[6]	MEM_0_MEM_DQ[6]	MEM_DQ[6]	MEM_DQ[6]
	21	MEM_DQ[5]	MEM_DQ[5]	MEM_0_MEM_DQ[5]	MEM_DQ[5]	MEM_DQ[5]
	20	MEM_DQ[4]	MEM_DQ[4]	MEM_0_MEM_DQ[4]	MEM_DQ[4]	MEM_DQ[4]
	19					
	18	MEM_DM_N[0]	MEM_DM_N[0]	MEM_0_MEM_DM_N[0]	MEM_DM_N[0]	MEM_DM_N[0]
	17	MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_0_MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_DQS_C[0]
	16	MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_0_MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_DQS_T[0]
	15	MEM_DQ[3]	MEM_DQ[3]	MEM_0_MEM_DQ[3]	MEM_DQ[3]	MEM_DQ[3]
	14	MEM_DQ[2]	MEM_DQ[2]	MEM_0_MEM_DQ[2]	MEM_DQ[2]	MEM_DQ[2]
	13	MEM_DQ[1]	MEM_DQ[1]	MEM_0_MEM_DQ[1]	MEM_DQ[1]	MEM_DQ[1]
	12	MEM_DQ[0]	MEM_DQ[0]	MEM_0_MEM_DQ[0]	MEM_DQ[0]	MEM_DQ[0]

continued...

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
BL0	11	MEM_DQ[15]	MEM_DQ[15]	MEM_0_MEM_DQ[15]	MEM_DQ[15]	MEM_DQ[15]
	10	MEM_DQ[14]	MEM_DQ[14]	MEM_0_MEM_DQ[14]	MEM_DQ[14]	MEM_DQ[14]
	9	MEM_DQ[13]	MEM_DQ[13]	MEM_0_MEM_DQ[13]	MEM_DQ[13]	MEM_DQ[13]
	8	MEM_DQ[12]	MEM_DQ[12]	MEM_0_MEM_DQ[12]	MEM_DQ[12]	MEM_DQ[12]
	7					
	6	MEM_DM_N[1]	MEM_DM_N[1]	MEM_0_MEM_DM_N[1]	MEM_DM_N[1]	MEM_DM_N[1]
	5	MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_0_MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_DQS_C[1]
	4	MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_0_MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_DQS_T[1]
	3	MEM_DQ[11]	MEM_DQ[11]	MEM_0_MEM_DQ[11]	MEM_DQ[11]	MEM_DQ[11]
	2	MEM_DQ[10]	MEM_DQ[10]	MEM_0_MEM_DQ[10]	MEM_DQ[10]	MEM_DQ[10]
	1	MEM_DQ[9]	MEM_DQ[9]	MEM_0_MEM_DQ[9]	MEM_DQ[9]	MEM_DQ[9]
	0	MEM_DQ[8]	MEM_DQ[8]	MEM_0_MEM_DQ[8]	MEM_DQ[8]	MEM_DQ[8]

Note: The presence of an asterisk (*) in the above table indicates an ECC byte location.

3.1.3.4. LPDDR5 Pin Placement

Table 12. LPDDR5 Pin Placement

Lane Number	Pin Index	x32	2 Channel x16
BL7	95	MEM_DQ[31]	MEM_1_MEM_DQ[15]
	94	MEM_DQ[30]	MEM_1_MEM_DQ[14]
	93	MEM_DQ[29]	MEM_1_MEM_DQ[13]
	92	MEM_DQ[28]	MEM_1_MEM_DQ[12]
	91		
	90	MEM_DMI[3]	MEM_1_MEM_DMI[1]
	89	MEM_RDQS_C[3]	MEM_1_MEM_RDQS_C[1]
	88	MEM_RDQS_T[3]	MEM_1_MEM_RDQS_T[1]
	87	MEM_DQ[27]	MEM_1_MEM_DQ[11]
	86	MEM_DQ[26]	MEM_1_MEM_DQ[10]
	85	MEM_DQ[25]	MEM_1_MEM_DQ[9]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	84	MEM_DQ[24]	MEM_1_MEM_DQ[8]
BL6	83	MEM_DQ[23]	MEM_1_MEM_DQ[7]
	82	MEM_DQ[22]	MEM_1_MEM_DQ[6]
	81	MEM_DQ[21]	MEM_1_MEM_DQ[5]
	80	MEM_DQ[20]	MEM_1_MEM_DQ[4]
	79		
	78	MEM_DMI[2]	MEM_1_MEM_DMI[0]
	77	MEM_RDQS_C[2]	MEM_1_MEM_RDQS_C[0]
	76	MEM_RDQS_T[2]	MEM_1_MEM_RDQS_T[0]
	75	MEM_DQ[19]	MEM_1_MEM_DQ[3]
	74	MEM_DQ[18]	MEM_1_MEM_DQ[2]
	73	MEM_DQ[17]	MEM_1_MEM_DQ[1]
	72	MEM_DQ[16]	MEM_1_MEM_DQ[0]
BL5	71		
	70		
	69		
	68		MEM_1_MEM_CS[1]
	67		MEM_1_CK_C
	66		MEM_1_CK_T
	65		MEM_1_MEM_CS[0]
	64		MEM_1_MEM_CA[6]
	63		MEM_1_RESET_N
	62		OCT_1_OCT_RZQIN
	61		
	60		
BL4	59		Differential "NSide" Reference Clock Input Site
	58		Differential "PSide" Reference Clock Input Site
	57		MEM_1_MEM_CA[5]
	56		MEM_1_MEM_CA[4]
	55		MEM_1_MEM_WCK_C[1]
	54		MEM_1_MEM_WCK_T[1]
	53		MEM_1_MEM_WCK_C[0]
	52		MEM_1_MEM_WCK_T[0]
	51		MEM_1_MEM_CA[3]
	50		MEM_1_MEM_CA[2]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	49		MEM_1_MEM_CA[1]
	48		MEM_1_MEM_CA[0]
BL3	47		
	46		
	45		
	44	MEM_CS[1]	MEM_0_MEM_CS[1]
	43	MEM_CK_C	MEM_0_CK_C
	42	MEM_CK_T	MEM_0_CK_T
	41	MEM_CS[0]	MEM_0_MEM_CS[0]
	40	MEM_CA[6]	MEM_0_MEM_CA[6]
	39	MEM_RESET_N	MEM_0_RESET_N
	38	RZQ Site	OCT_0_OCT_RZQIN
	37		
	36		
BL2	35	Differential "N-Side" Reference Clock Input Site	Differential "NSide" Reference Clock Input Site
	34	Differential "P-Side" Reference Clock Input Site	Differential "PSide" Reference Clock Input Site
	33	MEM_CA[5]	MEM_0_MEM_CA[5]
	32	MEM_CA[4]	MEM_0_MEM_CA[4]
	31	MEM_WCK_C[1]	MEM_0_MEM_WCK_C[1]
	30	MEM_WCK_T[1]	MEM_0_MEM_WCK_T[1]
	29	MEM_WCK_C[0]	MEM_0_MEM_WCK_C[0]
	28	MEM_WCK_T[0]	MEM_0_MEM_WCK_T[0]
	27	MEM_CA[3]	MEM_0_MEM_CA[3]
	26	MEM_CA[2]	MEM_0_MEM_CA[2]
	25	MEM_CA[1]	MEM_0_MEM_CA[1]
	24	MEM_CA[0]	MEM_0_MEM_CA[0]
BL1	23	MEM_DQ[15]	MEM_0_MEM_DQ[15]
	22	MEM_DQ[14]	MEM_0_MEM_DQ[14]
	21	MEM_DQ[13]	MEM_0_MEM_DQ[13]
	20	MEM_DQ[12]	MEM_0_MEM_DQ[12]
	19		
	18	MEM_DMI[1]	MEM_0_MEM_DMI[1]
	17	MEM_RDQS_C[1]	MEM_0_MEM_RDQS_C[1]
	16	MEM_RDQS_T[1]	MEM_0_MEM_RDQS_T[1]
	15	MEM_DQ[11]	MEM_0_MEM_DQ[11]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	14	MEM_DQ[10]	MEM_0_MEM_DQ[10]
	13	MEM_DQ[9]	MEM_0_MEM_DQ[9]
	12	MEM_DQ[8]	MEM_0_MEM_DQ[8]
BL0	11	MEM_DQ[7]	MEM_0_MEM_DQ[7]
	10	MEM_DQ[6]	MEM_0_MEM_DQ[6]
	9	MEM_DQ[5]	MEM_0_MEM_DQ[5]
	8	MEM_DQ[4]	MEM_0_MEM_DQ[4]
	7		
	6	MEM_DMI[0]	MEM_0_MEM_DMI[0]
	5	MEM_RDQS_C[0]	MEM_0_MEM_RDQS_C[0]
	4	MEM_RDQS_T[0]	MEM_0_MEM_RDQS_T[0]
	3	MEM_DQ[3]	MEM_0_MEM_DQ[3]
	2	MEM_DQ[2]	MEM_0_MEM_DQ[2]
	1	MEM_DQ[1]	MEM_0_MEM_DQ[1]
	0	MEM_DQ[0]	MEM_0_MEM_DQ[0]

Note: It is important to strictly follow the pin placement for a given memory topology when assigning pin locations for your EMIF IP.

The recommended approach is to manually constrain some interface signals and allow the Quartus Prime Fitter to place the pins. For this method of I/O placement, you must constrain the following signals:

- PLL reference clock
- RZQ pin
- MEM_RESET_N

Do not change the location for the EMIF pin using a `.qsf` assignment or the Pin Planner if you need to swap the DQ pins within a DQS group or the DQS group to simplify board design.

Refer to the *Configuring DQ Pin Swizzling* topic in the *External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide* for more information on how to swap the DQ pin and DQS group.

For dual-rank component interfaces, you cannot have different swizzling specifications for rank 0 and rank 1.

3.1.3.5. I/O Sub-Bank Usage

The pins in an I/O bank can serve as address and command pins, data pins, or clock and strobe pins for an external memory interface.

A given sub-bank cannot be shared between multiple EMIFs.

All the sub-banks are capable of functioning as the address and command bank.

3.1.4. Agilex 7 M-Series EMIF Architecture: I/O Lane

An I/O bank contains two sub-banks. Each sub-bank contains 48 I/O pins, organized into four I/O lanes of 12 pins each. You can identify where a pin is located within an I/O bank based on its `Index within I/O Bank` in the device pinout.

Table 13. Pin Index Mapping

Pin Index	Lane	Sub-bank Location
0-11	0	Bottom
12-23	1	
24-35	2	
36-47	3	
48-59	4	Top
60-71	5	
72-83	6	
84-95	7	

Each I/O lane can implement one x8/x9 read capture group (DQS group), with two pins functioning as the read capture clock/strobe pair (DQS/DQS#), and up to 10 pins functioning as data pins (DQ and DM pins). To implement a x18 group, you can use multiple lanes within the same sub-bank.

It is also possible to implement a pair of x4 groups in a lane. In this case, four pins function as clock/strobe pair, and 8 pins function as data pins. DM is not available for x4 groups. There must be an even number of x4 groups for each interface.

For x4 groups, you must place DQS0 and DQS1 in the same I/O lane as a pair. Similarly, DQS2 and DQS3 must be paired. In general, DQS(x) and DQS(x+1) must be paired in the same I/O lane.

For DQ and DQS pin assignments for various configurations, refer to the Agilex 7 M-Series device pin tables.

Table 14. Lanes Used Per DQS Group

Group Size	Number of Lanes Used	Maximum Number of Data Pins per Group
x8 / x9	1	10
x18	2	22
pair of x4	1	4 per group, 8 per lane

Figure 5. x4 Group

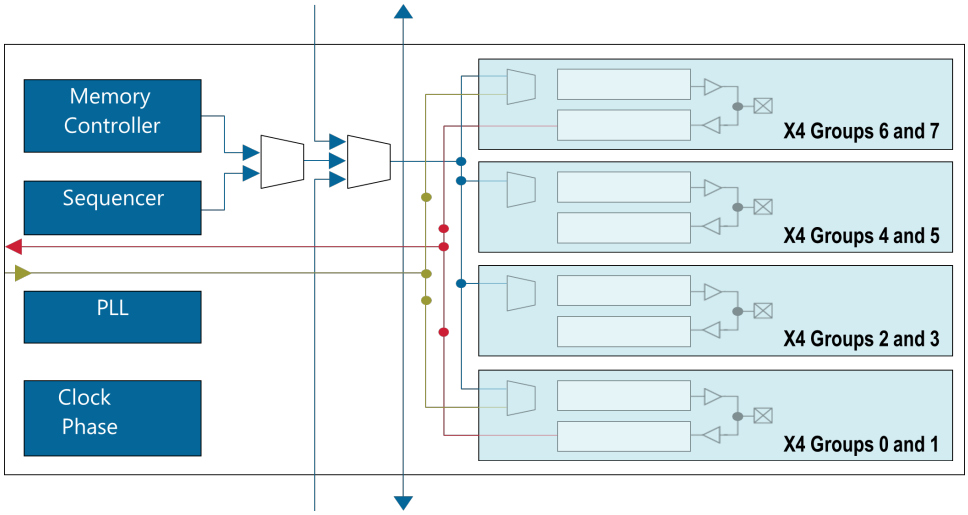


Figure 6. x8 Group

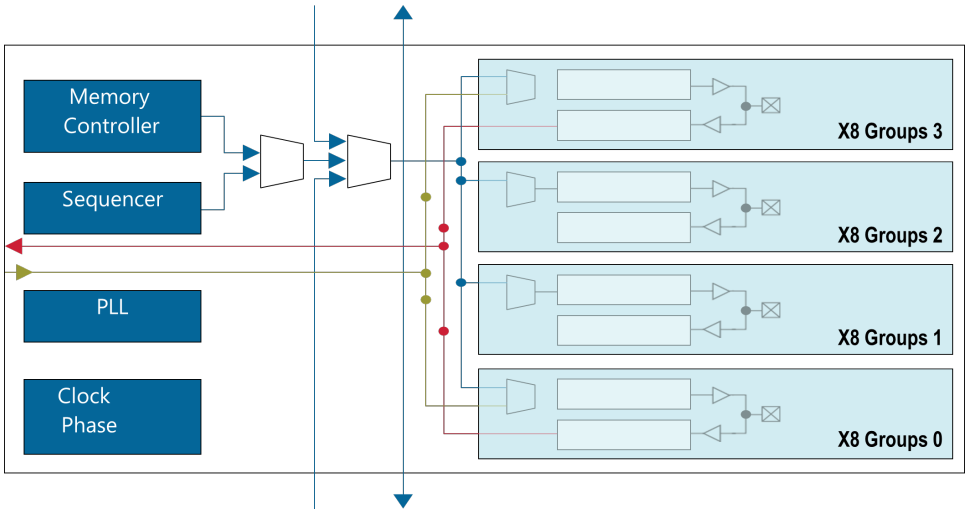
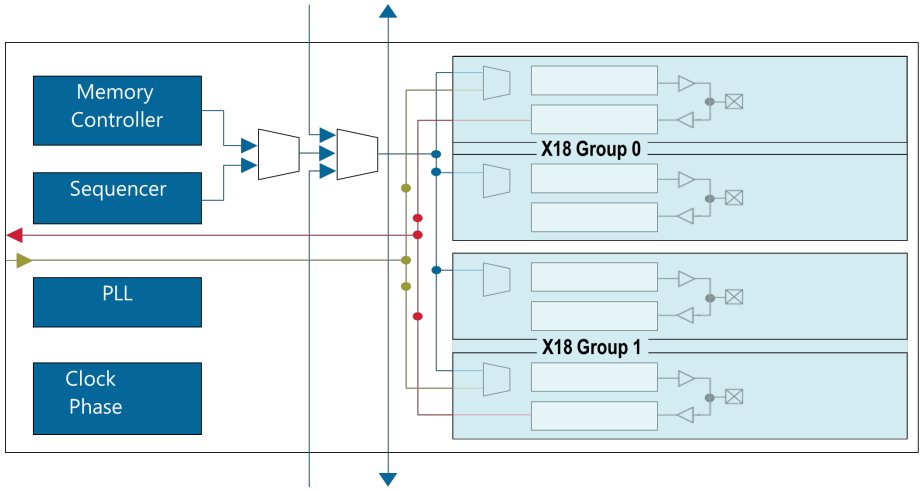


Figure 7. x18 Group



3.1.5. Agilex 7 M-Series EMIF Architecture: Input DQS Clock Tree

The input DQS clock tree is a balanced clock network that distributes the read capture clock (such as QK/QK# which are free-running read clocks) and strobe (such as DQS_T/DQS_C) from the external memory device to the read capture registers inside the I/Os.

You can configure an input DQS clock tree in x4 mode, x8/x9 mode, or x18 mode.

Within every bank, only certain physical pins at specific locations can drive the input DQS clock trees. The pin locations that can drive the input DQS clock trees vary, depending on the size of the group.

Table 15. Pins Usable as Read Capture Clock / Strobe Pair

Group Size	Index of Lanes Spanned by Clock Tree [1]	Sub-Bank	Index of Pins Usable as Read Capture Clock / Strobe Pair	
			DQS_T	DQS_C
x4	0A	Bottom	4	5
x4	0B		6	7
x4	1A		16	17
x4	1B		18	19
x4	2A		28	29
x4	2B		30	31
x4	3A		40	41
x4	3B		42	43
x8 / x9	0		4	5
x8 / x9	1		16	17
x8 / x9	2		28	29
x8 / x9	3		40	41

continued...

Group Size	Index of Lanes Spanned by Clock Tree [1]	Sub-Bank	Index of Pins Usable as Read Capture Clock / Strobe Pair	
			DQS_T	DQS_C
x18	0, 1		4	5
x18	2, 3		28	29
x4	0A	Top	52	53
x4	0B		54	55
x4	1A		64	65
x4	1B		66	67
x4	2A		76	77
x4	2B		78	79
x4	3A		88	89
x4	3B		90	91
x8 / x9	0		52	53
x8 / x9	1		64	65
x8 / x9	2		76	77
x8 / x9	3		88	89
x18	0,1		52	53
x18	2,3		76	77
Note: [1] *A* and *B* refer to the two nibbles within the lane.				

3.1.6. Agilex 7 M-Series EMIF Architecture: PHY Clock Tree

Dedicated high-speed clock networks drive I/Os in the Agilex 7 M-Series EMIF.

The relatively short span of the PHY clock trees results in low jitter and low duty-cycle distortion, maximizing the data valid window.

The PHY clock tree in Agilex 7 M-Series devices can run as fast as 1.6 GHz. All Agilex 7 M-Series external memory interfaces use the PHY clock trees.

3.1.7. Agilex 7 M-Series EMIF Architecture: PLL Reference Clock Networks

Each sub-bank includes an I/O bank I/O PLL that can drive the PHY clock trees of that bank, through dedicated connections. In addition to supporting EMIF-specific functions, the I/O bank I/O PLLs can also serve as general-purpose PLLs for user logic.

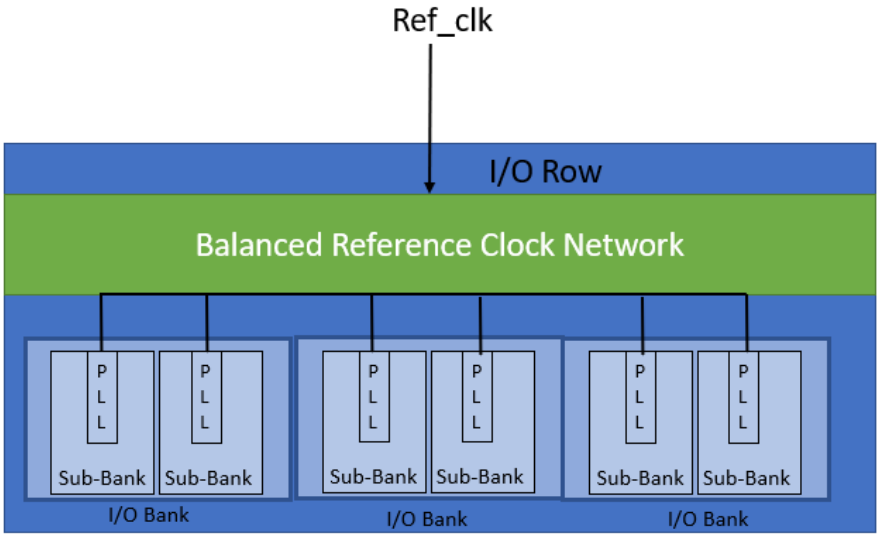
The PLL reference clock must be constrained to the address and command sub-bank only.

Agilex 7 M-Series external memory interfaces that span multiple banks use the PLL in each bank. The Agilex 7 M-Series architecture allows for relatively short PHY clock networks, reducing jitter and duty-cycle distortion.

The following mechanisms ensure that the clock outputs of individual I/O bank I/O PLLs in a multi-bank interface remain in phase:

- A single PLL reference clock source feeds all I/O bank I/O PLLs. The reference clock signal reaches the PLLs by a balanced PLL reference clock tree. The Quartus Prime software automatically configures the PLL reference clock tree so that it spans the correct number of banks. This clock must be free-running and stable prior to FPGA configuration.
- The EMIF IP sets the PLL configuration (counter settings, bandwidth settings, compensation and feedback mode setting) values appropriately to maintain synchronization among the clock dividers across the PLLs. This requirement restricts the legal PLL reference clock frequencies for a given memory interface frequency and clock rate. If you plan to use an on-board oscillator, you must ensure that its frequency matches the PLL reference clock frequency that you select from the displayed list.

Figure 8. PLL Balanced Reference Clock Tree



3.1.8. Agilex 7 M-Series EMIF Architecture: Clock Phase Alignment

In Agilex 7 M-Series external memory interfaces, a global clock network clocks registers inside the FPGA core, and the PHY clock network clocks registers inside the FPGA periphery. Clock phase alignment circuitry employs negative feedback to dynamically adjust the phase of the core clock signal to match the phase of the PHY clock signal.

The clock phase alignment feature effectively eliminates the clock skew effect in all transfers between the core and the periphery, facilitating timing closure. All Agilex 7 M-Series external memory interfaces employ clock phase alignment circuitry.

Figure 9. Clock Phase Alignment Illustration

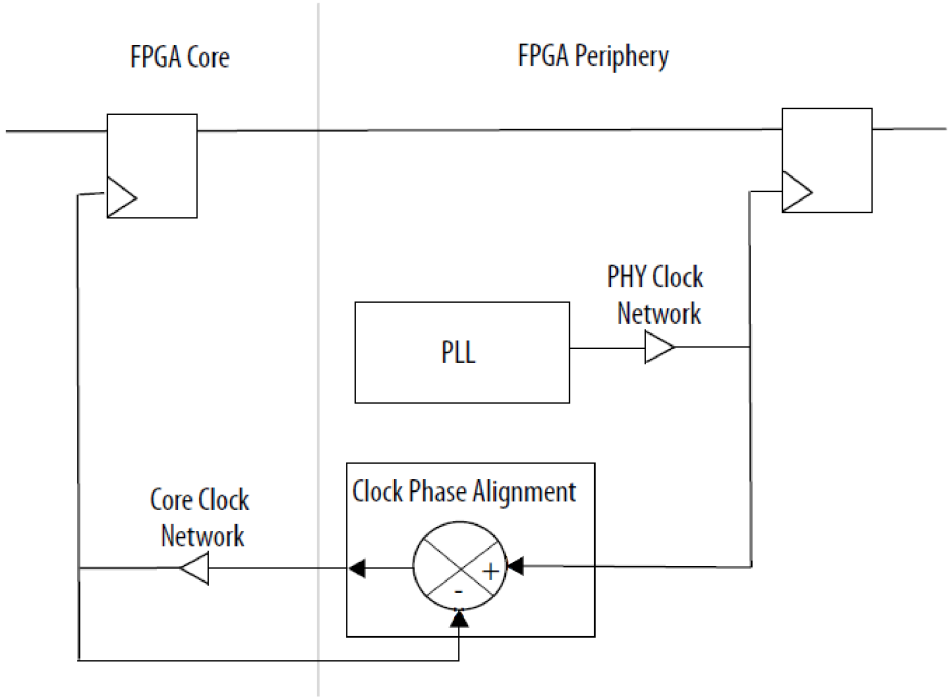
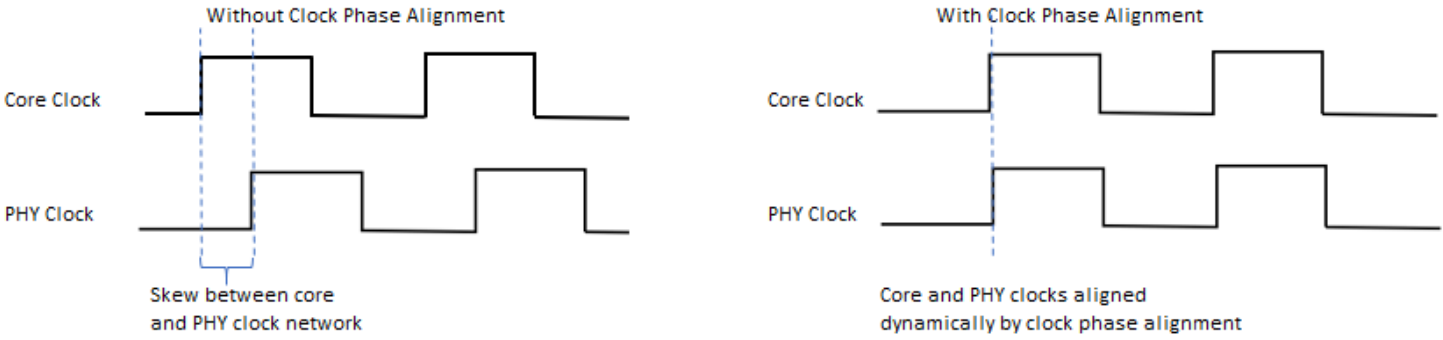


Figure 10. Effect of Clock Phase Alignment



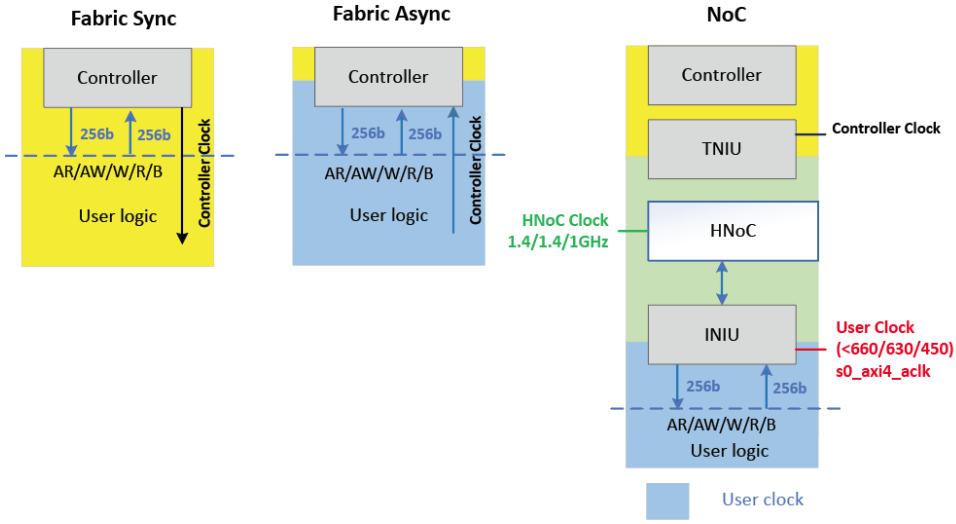
3.1.9. User Clock in Different Core Access Modes

The EMIF IP for Agilex 7 M-Series devices supports three user access modes.

- Synchronous fabric clocking, where the EMIF IP provides a user clock.
  - The user clock frequency is limited by the maximum core-to-periphery (C2P) and periphery-to-core (P2C) frequency of 400 MHz.
  - In DDR4, the user clock frequency will be one-quarter of the memory clock frequency ((mem_CK)/4).
  - In DDR5 and LPDDR5, the user clock frequency will be one-eighth of the memory clock frequency ((mem CK)/8).
- Asynchronous fabric clocking, where you provide the clock to the EMIF IP.
  - The asynchronous user clock can come from any user clock source on the device.
  - The user clock frequency has no dependency on the memory clock (mem_CK).
- NoC Mode.
  - The initiator clock can come from any user clock source on the device.
  - For additional information, refer to the *Agilex 7 M-Series FPGA Network-on-Chip (NoC) User Guide*.

The following figures illustrate the different clocking styles available for the Agilex 7 M-Series EMIF IP. The NoC mode shown is the simplest NoC mode.

Figure 11. Access Modes



Benefits of Each Access Mode

- Synchronous fabric clocking is required for DDR4 DIMM.
- Asynchronous fabric access mode has the lowest latency.
- NoC mode can achieve the highest bandwidth and efficiency, and has the lowest initiator blockage.

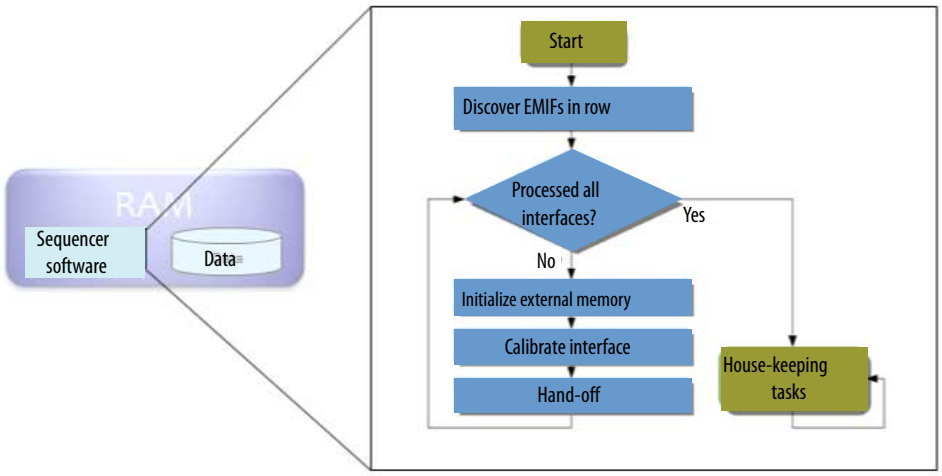
3.2. Agilex 7 M-Series EMIF Sequencer

The Agilex 7 M-Series EMIF sequencer is fully hardened in silicon, with executable code to handle protocols and topologies. Hardened RAM contains the calibration algorithm.

The Agilex 7 M-Series EMIF sequencer is responsible for the following operations:

- Initializes memory devices.
- Calibrates the external memory interface.
- Governs the hand-off of control to the memory controller.
- Handles recalibration requests and debug requests.
- Handles all supported protocols and configurations.

Figure 12. Agilex 7 M-Series EMIF Sequencer Operation



3.2.1. Agilex 7 M-Series Mailbox Structure and Register Definitions

The mailbox is a software structure that the calibration subsystem manager (SSM) polls periodically.

All accesses to the mailbox should align to 32-bit boundaries, with no byte masking support. The following tables show the mailbox structure and the calibration status register definitions.

Table 16. Mailbox Structure

Register Name	Byte Offset	Width (bits)	Access	Description
STATUS	1024	32	RO	*[Output]* Status Register "At a Glance" status register. This field is automatically updated by the Calibration I/O SSM and no explicit operation is required to trigger an update.

continued...

Register Name	Byte Offset	Width (bits)	Access	Description
				For details on the calibration status register, refer to the following table, *Calibration Status Register*.
Reserved				
CMD_PARAM_6	1056	32	RW	*[Input]* This register specifies the seventh parameter (if applicable) for the requested command.
CMD_PARAM_5	1060	32	RW	*[Input]* This register specifies the sixth parameter (if applicable) for the requested command.
CMD_PARAM_4	1064	32	RW	*[Input]* This register specifies the fifth parameter (if applicable) for the requested command.
CMD_PARAM_3	1068	32	RW	*[Input]* This register specifies the fourth parameter (if applicable) for the requested command.
CMD_PARAM_2	1072	32	RW	*[Input]* This register specifies the third parameter (if applicable) for the requested command.
CMD_PARAM_1	1076	32	RW	*[Input]* This register specifies the second parameter (if applicable) for the requested command.
CMD_PARAM_0	1080	32	RW	*[Input]* This register specifies the first parameter (if applicable) for the requested command.
CMD_REQ	1084	32	RW	*[Input]* This register specifies the command to be performed and the target IP type and identifier.
Reserved				
CMD_RESPONSE_DATA_2	1104	32	RO	*[Output]* For commands that return up to 3.5x 32-bit values, CMD_RESPONSE_DATA 0/1/2, and/or CMD_RESPONSE_DATA_SHORT fields contain the requested response data.
CMD_RESPONSE_DATA_1	1108	32	RO	*[Output]* This register can contain two types of values depending on the requested operation. For commands that return up to 3.5x 32-bit values, CMD_RESPONSE_DATA 0/1/2, and/or CMD_RESPONSE_DATA_SHORT fields contain the requested response data. For commands that return more data, this register specifies a pointer to a data buffer within the 4K User. It is a byte offset relative to the start of the 4Kbyte RAM. Users should not assume that this offset value remains static as the value of this pointer offset may change depending on the requested operation.
CMD_RESPONSE_DATA_0	1112	32	RO	*[Output]* This register can contain two types of values depending on the requested operation. For commands that return up to 3.5x 32-bit values, CMD_RESPONSE_DATA 0/1/2, and/or CMD_RESPONSE_DATA_SHORT fields contain the requested response data.

continued...

Register Name	Byte Offset	Width (bits)	Access	Description
				For commands that return more data, this value contains the size of the returned data structure in bytes. The location of the data buffer is specified in CMD_RESPONSE_DATA_1.
CMD_RESPONSE_STATUS	1116	32	RW	*[Output]* Command Interface status Captures the current state of the Mailbox's Command Interface (e.g., Is the response data ready for the user?). CMD_RESPONSE_DATA_SHORT field in this register can be used for 16-bit response data.

Table 17. Calibration Status Register

Bit	Name	Description	Access	Reset
[31:3]	Reserved			
[2]	STATUS_CAL_BUSY	Indicates calibration busy status of any external memory interfaces in the IO96: '1' - One or more EMIF instances are busy with calibration. '0' - No EMIF instances are busy with calibration.	RO	0x0
[1]	STATUS_CAL_FAIL	Indicates calibration failure status of any external memory interfaces in the IO96: '1' - One or more EMIF instances have failed to calibrate successfully. '0' - No calibration failures have been reported for any of the EMIFs.	RO	0x0
[0]	STATUS_CAL_SUCCESS	Indicates final calibration status of all interfaces in the IO96: '1' All EMIF instances within the IO96 have calibrated successfully. '0' One or more EMIF instances in the IO96 have either failed to calibrate or have not completed calibration yet.	RO	0x0

3.2.1.1. Mailbox Supported Commands

Table 18. Supported Commands

CMD_TYPE Enum	CMD_OPCODE Enum	Value
CMD_GET_SYS_INFO	GET_MEM_INTF_INFO	0x001
CMD_GET_MEM_INFO	GET_MEM_TECHNOLOGY	0x002
	GET_MEMCLK_FREQ_KHZ	0x003
	GET_MEM_WIDTH_INFO	0x004
CMD_TRIG_CONTROLLER_OP	ECC_ENABLE_SET	0x0101
	ECC_ENABLE_STATUS	0x0102
	ECC_INTERRUPT_STATUS	0x0103
	ECC_INTERRUPT_ACK	0x0104

continued...

CMD_TYPE Enum	CMD_OPCODE Enum	Value
	ECC_INTERRUPT_MASK	0x0105
	ECC_WRITEBACK_ENABLE	0x0106
	ECC_GET_SBE_INFO	0x0107
	ECC_GET_DBE_INFO	0x0108
	ECC_INJECT_ERROR	0x0109
	ECC_SCRUB_IN_PROGRESS_STATUS	0x0201
	ECC_SCRUB_MODE_0_START	0x0202
	ECC_SCRUB_MODE_1_START	0x0203
	BIST_STANDARD_MODE_START	0x0301
	BIST_RESULTS_STATUS	0x0302
	BIST_MEM_INIT_START	0x0303
	BIST_MEM_INIT_STATUS	0x0304
	BIST_SET_DATA_PATTERN_UPPER	0x0305
	BIST_SET_DATA_PATTERN_LOWER	0x0306
	LP_MODE_ENTER	0x0d01
	LP_MODE_EXIT	0x0d02
	LP_MODE_STATUS	0x0d03

3.2.1.2. Mailbox Command Definitions

Table 19. CMD_TYPE Definition

CMD_TYPE	Value	Description
CMD_NOP	0x00	No operation command.
CMD_GET_SYS_INFO	0x01	Retrieving information about the IO96B configuration.
CMD_GET_MEM_INFO	0x02	Retrieving information about the memory interface operation.
CMD_TRIG_CONTROLLER_OP	0x04	Triggering memory controller-related operations.
CMD_TRIG_MEM_CAL_OP	0x05	Triggering calibration events.

Table 20. CMD_REQ Definition

Bit	Name	Description	Access	Reset
[31:29]	CMD_TARGET_IP_TYPE	Indicates the type of IP, as follows:	Read-Write	0x0

continued...

Bit	Name	Description	Access	Reset
		• 0x0 – Not used. • 0x1 – Primary MC of primary IO96B. • 0x2 – Secondary MC of primary IO96B. • 0x3 – Primary MC of secondary IO96B. • 0x4 – Secondary MC of secondary IO96B.		
[28:24]	CMD_TARGET_IP_INSTANCE_ID	IP identifier.	Read-Write	0x00
[23:16]	CMD_TYPE	The type of command that the user wants the firmware to perform.	Read-Write	0x00
[15:0]	CMD_OPCODE	The opcode of the command that the user wants the firmware to perform.	Read-Write	0x00

Table 21. CMD_TARGET_IP_TYPE Definition

Multi-channel/ Lockstep Configurations	CMD_TARGET_IP_TYPE			
	1 – Primary MC, Primary IO96B	2 – Secondary MC, Primary IO96B	3 – Primary MC, Secondary IO96B	4 – Secondary MC, Secondary IO96B
LPDDR4/5 2CHx16	CH1	CH2		
LPDDR4/5 4CHx16	CH1	CH2	CH3	CH4
DDR5 2CHx16	CH1	CH2		
DDR5 2CHx32	CH1		CH2	
DDR5 x40 lockstep	CH1	*		
DDR4 x40 lockstep	CH1	*		
DDR4 x64, x72 lockstep	CH1	*	*	*
Note: * These controllers are used but have no (or limited) mailbox features due to limited lockstep capabilities.				

Table 22. Command Definitions

CMD_REQ	Description
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <UNUSED> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <UNUSED> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_GET_SYS_INFO** *CMD_REQ [15:0]:* CMD_OPCODE = **GET_MEM_INTF_INFO**	Command to get the memory interface IP type and instance ID of all the IPs in the IO96B. [Inputs] N/A [Outputs[KSH1] [VCV2] [KSH3]] CMD_RESPONSE_DATA_SHORT [1:0]: NUM_USED_MEM_INTF Number of memory interfaces instantiated. CMD_RESPONSE_DATA_0 [31:29]: INTF_0_IP_TYPE Indicates the type of IP for Interface 0: 0x0 – Not used 0x1 – EMIF CMD_RESPONSE_DATA_0 [28:24]: INTF_0_INSTANCE_ID IP identifier for Interface 0. CMD_RESPONSE_DATA_1 [31:29]: INTF_1_IP_TYPE Indicates the type of IP for Interface 1: 0x0 – Not used

continued...

CMD_REQ	Description
	0x1 – EMIF CMD_RESPONSE_DATA_1 [28:24]: INTF_1_INSTANCE_ID IP identifier for Interface 1.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_GET_MEM_INFO** *CMD_REQ [15:0]:* CMD_OPCODE = **GET_MEM_TECHNOLOGY**	Command to get the memory technology of the memory interface specified using the instance ID. [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_SHORT [2:0]: MEM_TECHNOLOGY Reports the memory type as below: 0x0 = DDR4, 0x1 = DDR5, 0x2 = DDR5_RDIMM, 0x3 = LPDDR4, 0x4 = LPDDR5, 0x5 = QDRIV
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_GET_MEM_INFO** *CMD_REQ [15:0]:* CMD_OPCODE = **GET_MEMCLK_FREQ_KHZ**	Command to get the memory clock frequency of the memory interface specified using the instance ID. [Inputs] CMD_PARAM_0 [1:0]: FREQUENCY_SET_POINT Get clock frequency for the specified frequency set point. 0x0 = Frequency set point 0 0x1 = Frequency set point 1 0x2 = Frequency set point 2 CMD_PARAM_0 [2:2]: USE_CURRENT_FSP Get clock frequency for the current frequency set point. 0x0 = Use FSP specified using FREQUENCY_SET_POINT. 0x1 = Use current FSP [Outputs] CMD_RESPONSE_DATA_0: DRAM_CLK_FREQ_KHZ Reports the memory clock frequency in KHz for the input frequency set point. [Errors] CMD_RESPONSE_STATUS - STATUS_CMD_RESPONSE_ERROR: 000 – No errors 111 – The FSP specified using FREQUENCY_SET_POINT is not defined.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_GET_MEM_INFO** *CMD_REQ [15:0]:* CMD_OPCODE = **GET_MEM_WIDTH_INFO**	Command to get the memory width information of the memory interface specified using the instance ID. [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_0 [7:0]: DQ_WIDTH CMD_RESPONSE_DATA_0 [15:8]: CS_WIDTH CMD_RESPONSE_DATA_0 [23:16]: C_WIDTH CMD_RESPONSE_DATA_1 [7:0]: TOTAL_MEM_CAPACITY Memory device capacity in Gb (gigabits) calculated as: CAPACITY = (DQ_WIDTH / DEVICE_WIDTH) * DEVICE_DENSITY
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_ENABLE_SET**	Command to enable different ECC modes for the memory interface specified using the instance ID. [Inputs] CMD_PARAM_0 [1:0]: ECC_ENABLE Set the current ECC error reporting (single-bit and double-bit errors) and correcting (single-bit errors) that is enabled. 'b00 = ECC is disabled. Data is written to the memory without ECC values, and data is returned to the user interface without being verified for accuracy. 'b01 = ECC is enabled, but without detection or correction.

continued...

CMD_REQ	Description
	'b10 = ECC is enabled with detection, but correction is not supported. When an error is found on a read operation, ECC reporting parameters are updated for read commands. Erroneous data is returned to the user on read commands and written to the memory on write commands. 'b11 = ECC is enabled with detection and correction. When an error is found on a read operation, the ECC reporting parameters are updated for read commands. Single bit errors are corrected automatically by the controller in both read and write commands. CMD_PARAM_0[2:2]: ECC_TYPE 'b0 = Out-of-Band ECC 'b1 =In-line ECC [Outputs] N/A
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_ENABLE_STATUS**	Command to get the ECC enable status of the memory interface specified using the instance ID. [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_SHORT [1:0]: ECC_ENABLE Reports the current ECC error reporting (single-bit and double-bit errors) and correcting (single-bit errors) that is enabled. 'b00 = ECC is disabled. Data is written to the memory without ECC values, and data is returned to the user interface without being verified for accuracy. 'b01 = ECC is enabled, but without detection or correction. 'b10 = ECC is enabled with detection, but correction is not supported. When an error is found on a read operation, ECC reporting parameters are updated for read commands. Erroneous data is returned to the user on read commands and written to the memory on write commands. 'b11 = ECC is enabled with detection and correction. When an error is found on a read operation, the ECC reporting parameters are updated for read commands. Single bit errors are corrected automatically by the controller in both read and write commands. CMD_RESPONSE_DATA_SHORT[2:2]: ECC_TYPE 'b0 = Out-of-Band ECC 'b1 =In-line ECC
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_INTERRUPT_STATUS**	Command to get status of ECC interrupts for the memory interface specified using the instance ID. [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_0 [16:0]: ECC_INTERRUPT_STATUS Reports the interrupts related to the ECC logic. Bit [8] = An ECC correctable error has been detected in a scrubbing read operation Bit [7] = The triggered scrub operation has completed. Bit [6] = One or more ECC writeback commands could not be executed. Bit [3] = Another un-correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged. Bit [2] = An un-correctable ECC event has been detected on a read operation.

continued...

CMD_REQ	Description
	Bit [1] = Another correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged. Bit [0] = A correctable ECC event has been detected on a read operation
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_INTERRUPT_ACK**	Command to acknowledge and clear the ECC interrupts for the memory interface specified using the instance ID. [Inputs] CMD_PARAM_0 [16:0]: ECC_INTERRUPT_ACK Used to acknowledge and clear the interrupts related to the ECC logic. Bit [8] = An ECC correctable error has been detected in a scrubbing read operation Bit [7] = The triggered scrub operation has completed. Bit [6] = One or more ECC writeback commands could not be executed. Bit [3] = Another un-correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged. Bit [2] = An un-correctable ECC event has been detected on a read operation. Bit [1] = Another correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged. Bit [0] = A correctable ECC event has been detected on a read operation [Outputs] N/A
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_INTERRUPT_MASK**	Command to set mask for ECC interrupts for the memory interface specified using the instance ID, in order to disable specific ECC interrupts. [Inputs] CMD_PARAM_0 [16:0]: ECC_INTERRUPT_MASK If any bit is set to 'b1 in this parameter, the corresponding interrupt does NOT trigger an interrupt on the top-level EMIF interrupt signal. Bit [13] = A RMW Read Link ECC double-bit error has been detected Bit [12] = A Read Link ECC double-bit error has been detected. Bit [11] = A Read Link ECC single-bit error has been detected. Bit [10] = A Write Link ECC double-bit error has been detected by the periodic MRR to MR43. Bit [9] = A Write Link ECC single-bit error has been detected by the periodic MRR to MR43. Bit [8] = An ECC correctable error has been detected in a scrubbing read operation Bit [7] = The triggered scrub operation has completed. Bit [6] = One or more ECC writeback commands could not be executed. Bit [3] = Another un-correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged. Bit [2] = An un-correctable ECC event has been detected on a read operation. Bit [1] = Another correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged.

continued...

CMD_REQ	Description
	Bit [0] = A correctable ECC event has been detected on a read operation [Outputs] N/A
CMD_REQ column is: *CMD_REQ [31:29]*: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]*: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]*: CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]*: CMD_OPCODE = **ECC_WRITEBACK_ENABLE**	Command to set automatic writing of corrected errors on read operation for the memory interface specified using the instance ID CMD_PARAM_0 [0:0]: ECC_WRITEBACK_EN Enables automatic writing of corrected data on single bit correctable errors on read operations. This parameter is only relevant if ECC is enabled with detection and correction (ECC_ENABLE = 'b11). Note: No writebacks will be issued during BIST. 'b0 = Disable. 'b1 = Enable [Outputs] N/A [Command-Specific Errors] 'b000 – No errors 'b001 – ECC detection and correction not enabled
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_GET_SBE_INFO**	Command to get the details on the single-bit (SBE) or correctable errors detected by ECC for the memory interface specified using the instance ID. This command should be called only if all of the following conditions are true: 1. A correctable ECC event occurs. 2. ECC is enabled in the ECC_ENABLE parameter ('b01, 'b10 or 'b11). [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_0 [31:0]: ECC_SBE_INFO_SIZE Holds the size of the single bit error details. The value of this is 192 bits. CMD_RESPONSE_DATA_1 [31:0]: ECC_SBE_INFO_PTR Holds the offset pointer of the single bit error details. The data at the pointer location is as shown below: OFFSET [0] to OFFSET [1]: ECC_SBE_ADDR [37:0] Holds the address of the read data that caused a single-bit correctable ECC event. The Controller pads this parameter with zeros for any address bits not used by the controller. Here, the 5th bit of OFFSET [0] has the 37th bit, and the lowest bit of OFFSET [1] has the 0th bit of ECC_SBE_ADDR. OFFSET [2] to OFFSET [3]: ECC_SBE_DATA [63:0] Holds the pre-corrected data associated with a single-bit correctable ECC event. OFFSET [4]: ECC_SBE_ID [6:0] Holds the source ID associated with a single-bit correctable ECC event. For AXI ports, the source ID is comprised of the Port ID (upper bit/s) and the Requestor ID, where the Requestor ID is the axi0_AWID for write commands or the axi0_ARID for read commands. OFFSET [5]: ECC_SBE_SYND [7:0] Holds the syndrome value associated with a single-bit correctable ECC error event. This value indicates which bit of the check code or data was erroneous. Table 7 shows the syndrome corresponding to the single bit errors. [Command-Specific Errors] 'b000 – No errors 'b001 – ECC not enabled

continued...

CMD_REQ	Description
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_GET_DBE_INFO**	Command to get the details on the double-bit (DBE) or uncorrectable errors detected by ECC for the memory interface specified using the instance ID. This command should be called only if all of the following conditions are true: 1. An uncorrectable ECC event occurs. 2. ECC is enabled in the ECC_ENABLE parameter (’b01, ’b10 or ’b11. [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_0 [31:0]: ECC_DBE_INFO_SIZE Holds the size of the double bit error details. The value of this will be 192 bits. CMD_RESPONSE_DATA_1 [31:0]: ECC_DBE_INFO_PTR Holds the offset pointer of the double bit error details. The data at the pointer location is as shown below: OFFSET [0] to OFFSET [1]: ECC_DBE_ADDR [37:0] Holds the address of the read data that caused a double-bit uncorrectable ECC event. The Controller pads this parameter with zeros for any address bits not used by the controller. Here, the 5th bit of OFFSET [0] has the 37th bit, and the lowest bit of OFFSET [1] has the 0th bit of ECC_SBE_ADDR. OFFSET [2] to OFFSET [3]: ECC_DBE_DATA [63:0] Holds the data associated with a double-bit uncorrectable ECC event. OFFSET [4]: ECC_DBE_ID [6:0] Holds the source ID associated with a double-bit uncorrectable ECC event. For AXI ports, the source ID is comprised of the Port ID (upper bit/s) and the Requestor ID, where the Requestor ID is the axi0_AWID for write commands or the axi0_ARID for read commands. OFFSET [5]: ECC_DBE_SYND [7:0] Holds the syndrome bits associated with a double-bit un-correctable ECC error event. This controller can indicate that only 2 bits of the data and/or check code are erroneous but can not identify which bits. Table 7 shows the syndrome corresponding to the single bit errors. [Command-Specific Errors] ‘b000 – No errors ‘b001 – ECC not enabled
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_SCRUB_IN_PROGRESS_STATUS**	Command to check if the ECC scrub is in-progress for the memory interface specified using the instance ID. [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: ECC_SCRUB_IN_PROGRESS Reports the scrubbing operation status. This parameter is read-only. ’b0 = Not actively performing a scrubbing operation. ’b1 = The Controller is in the process of performing a scrubbing operation.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_SCRUB_MODE_0_START**	Command to start ECC scrub in mode 0 where scrub is performed at regular intervals for the memory interface specified using the instance ID. [Inputs] CMD_PARAM_0 [15:0]: ECC_SCRUB_INTERVAL

continued...

CMD_REQ	Description
	Sets the minimum interval between two ECC scrubbing commands, in number of controller clock cycles. The controller clock is based on the Controller's operating frequency. Clearing this parameter to 0x0000 disables interval operation. CMD_PARAM_1 [11:0]: ECC_SCRUB_LEN Defines the length (in bytes) of the ECC scrubbing read command that the controller will issue. This value must be an integer multiple of the memory burst length, and the lowest 3 bits of this parameter must be cleared to 'b0. CMD_PARAM_2 [0:0]: ECC_SCRUB_FULL_MEM Defines whether to perform ECC scrub on full memory or on the specified address range. 'b0 – ECC scrub performed on address range specified using ECC_SCRUB_START_ADDR and ECC_SCRUB_END_ADDR 'b1 – ECC scrub performed on full memory address range CMD_PARAM_3 [31:0]: ECC_SCRUB_START_ADDR [31:0] CMD_PARAM_4 [5:0]: ECC_SCRUB_START_ADDR [36:32] Defines the starting address from where scrubbing operations begin. This value must be less than or equal to the value programmed into the ECC_SCRUB_END_ADDR parameter. Only used when ECC_SCRUB_FULL_MEM is 'b0. CMD_PARAM_5 [31:0]: ECC_SCRUB_END_ADDR [31:0] CMD_PARAM_6 [5:0]: ECC_SCRUB_END_ADDR [36:32] Defines the ending address at which scrubbing operations wrap around to the start address. This parameter must be programmed to a non-zero value for the scrubbing logic to operate. Only used when ECC_SCRUB_FULL_MEM is 'b0. [Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: ECC_SCRUB_INITIATED 'b1 – ECC scrub initiated successfully 'b0 – ECC scrub initiation failed [Command-Specific Errors] 'b000 – No errors 'b001 – ECC not enabled
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **ECC_SCRUB_MODE_1_START**	Command to start ECC scrub in mode 1 where scrub is performed when the controller is idle for the memory interface specified using the instance ID. [Inputs] CMD_PARAM_0 [15:0]: ECC_SCRUB_IDLE_CNT Defines the number of controller clock cycles that the scrubbing engine waits in the Controller's idle state before starting scrubbing operations. The Controller is considered idle when the command queue is empty. When this condition is detected, an internal counter loads with the value programmed in this parameter and count down on each controller clock. When the counter expires, either the scrubbing operation begins or the next address is tested. The controller clock is based on the Controller's operating frequency. Clearing this parameter to 0x0000 disables idle operation. CMD_PARAM_1 [11:0]: ECC_SCRUB_LEN Defines the length (in bytes) of the ECC scrubbing read command that the controller issues. This value must be an integer multiple of the memory burst length, and the lowest 3 bits of this parameter must be cleared to 'b0. CMD_PARAM_2 [0:0]: ECC_SCRUB_FULL_MEM

continued...

CMD_REQ	Description
	Defines whether to perform ECC scrub on full memory or on the specified address range. 'b0 – ECC scrub performed on address range specified using ECC_SCRUB_START_ADDR and ECC_SCRUB_END_ADDR 'b1 – ECC scrub performed on full memory address range CMD_PARAM_3 [31:0]: ECC_SCRUB_START_ADDR [31:0] CMD_PARAM_4 [5:0]: ECC_SCRUB_START_ADDR [36:32] Defines the starting address from where scrubbing operations begin. This value must be less than or equal to the value programmed into the ECC_SCRUB_END_ADDR parameter. Only used when ECC_SCRUB_FULL_MEM is 'b0. CMD_PARAM_5 [31:0]: ECC_SCRUB_END_ADDR [31:0] CMD_PARAM_6 [5:0]: ECC_SCRUB_END_ADDR [36:32] Defines the ending address at which scrubbing operations wrap around to the start address. This parameter must be programmed to a non-zero value for the scrubbing logic to operate. Only used when ECC_SCRUB_FULL_MEM is 'b0. [Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: ECC_SCRUB_INITIATED 'b1 – ECC scrub initiated successfully 'b0 – ECC scrub initiation failed [Command-Specific Errors] 'b000 – No errors 'b001 – ECC not enabled
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]*: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]*: CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]*: CMD_OPCODE = **ECC_INJECT_ERROR**	Command to force a specific check code to be written into memory interface specified using the instance ID for diagnostic purposes or for flagging a particular memory address as erroneous for future accesses. The procedure is as follows: 1. Set the ECC_ENABLE parameter to enable detection ('b1x). 2. Ensure that no writes to the controller are pending. 3. Send ECC_INJECT_ERROR command through mailbox setting the ECC_XOR_CHECK_BITS input parameter. Use the syndromes to program the ECC_XOR_CHECK_BITS parameter. Each byte of the ECC_XOR_CHECK_BITS parameter controls the ECC event forcing for a separate user-word space. For example, setting a value of 0xF4 as ECC_XOR_CHECK_BITS will result in the check bits to be updated and written to the memory such that it flags a single-bit correctable error on bit [0] of the user-word on subsequent access of the same address. 4. Execute a write command for an aligned user word. The controller will XOR the ECC_XOR_CHECK_BITS parameter with the generated checksum bits from the word written to the memory. The next read command to the same address will force the ECC error. 5. Depending on the programming of the ECC_XOR_CHECK_BITS parameter, a single bit, double bit or multi-bit ECC error will occur. For single bit and double bit errors, the appropriate bit in the ECC_INTERRUPT_STATUS parameter will be set to 'b1 and the ECC error signature parameters will be filled with the relevant information. [Inputs] CMD_PARAM_0 [31:0]: ECC_XOR_CHECK_BITS

continued...

CMD_REQ	Description
	The check bits generated by the next write operation will be XOR'ed with this parameter. The result will be written into memory as the new check value. ECC_XOR_CHECK_BITS [31:24] maps to user word [255:192] ECC_XOR_CHECK_BITS [23:16] maps to user word [191:128] ECC_XOR_CHECK_BITS [15:8] maps to user word [127:64] ECC_XOR_CHECK_BITS [7:0] maps to user word [63:0] [Outputs] N/A [Command-Specific Errors] 'b000 – No errors 'b001 – ECC detection not enabled
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_STANDARD_MODE_START**	Command to initiate Original MOVI1 3N BIST algorithm for data checking for the memory interface specified using the instance ID. This command MUST be followed by BIST_RESULTS_STATUS to get the results of the BIST operation and resume normal operation of the memory controller. [Inputs] CMD_PARAM_0 [5:0]: BIST_ADDR_SPACE [5:0] Used in BIST data checking to define the address space in bytes from 0 to 2addr_space that the BIST logic checks. As an example, if the addr_space parameter was programmed to 0x1c, then the BIST logic would check 228 bytes = 256 MBytes. Note: A BIST test must cover a minimum of 2 bursts. Therefore, the user must program this parameter to a value such that the start address and end address of the BIST test will encompass a minimum of 2 bursts. Only used if BIST_FULL_MEM is 'b0. CMD_PARAM_0 [6:6]: BIST_FULL_MEM Defines whether to perform BIST on full memory or on the specified address range. 'b0 – BIST performed on address range specified using BIST_START_ADDR and BIST_ADDR_SPACE 'b1 – BIST performed on full memory address range CMD_PARAM_1 [31:0]: BIST_START_ADDR [31:0] CMD_PARAM_2 [5:0]: BIST_START_ADDR [36:32] Used in BIST data checking and memory initialization programming to define the starting address for BIST checking in bytes. Only used if BIST_FULL_MEM is 'b0. [Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: BIST_INITIATED 'b1 – BIST initiated successfully 'b0 – BIST initiation failed [Command-Specific Errors] 'b00 – No errors 'b01 – A previous command's saved state not restored. For example, BIST_STANDARD_MODE_START should be followed by BIST_RESULTS_STATUS command to restored saved state and resume normal memory controller operation.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP**	Command to get BIST results for the previously initiated BIST operation for memory interface specified using the instance ID [Inputs] N/A [Outputs]

continued...

CMD_REQ	Description
CMD_REQ [15:0]: CMD_OPCODE = **BIST_RESULTS_STATUS**	CMD_RESPONSE_DATA_SHORT [0:0]: BIST_STATUS Holds the status of the BIST operation. 'b0 = BIST operation still in progress if previously initiated. 'b1 = BIST operation has been completed. [Command-Specific Error] 'b00 – No errors 'b01 – Could not restore saved state. BIST_STANDARD_MODE_START command MUST be immediately followed by BIST_RESULTS_STATUS to get the results of the BIST operation, restore saved state, and resume normal operation of the memory controller. The populated results, if any, may be invalid. CMD_RESPONSE_DATA_SHORT [3:3]: BIST_RESULT Holds the result of the BIST operation. For this BIST mode, the test ends at the first failure, or completely checks the specified data range if no failures are found. This value is valid when BIST_STATUS indicates that the BIST operation has completed. 'b0 = Data check failed. 'b1 = Data check passed. CMD_RESPONSE_DATA_0 [31:0]: BIST_FAIL_RESULT_SIZE Holds the size of the BIST failure results. The value of this will be 640 bits. CMD_RESPONSE_DATA_1 [31:0]: BIST_FAIL_RESULT_PTR Holds the offset pointer of the BIST failure results. The data at the pointer location is as shown below: OFFSET [0] to OFFSET [8]: BIST_EXP_DATA [287:0] Holds the expected read data for a BIST data check failure. Here, the highest bit of OFFSET [0] has the 287th bit, and the lowest bit of OFFSET [8] has the 0th bit of BIST_EXP_DATA. OFFSET [9] to OFFSET [10]: BIST_FAIL_ADDR [37:0] Holds the actual failing address for a BIST data check failure. OFFSET [11] to OFFSET [20]: BIST_FAIL_DATA [287:0] Holds the actual failing data for a BIST data check failure.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_MEM_INIT_START**	Command to initiate memory initialization BIST for the memory interface specified using the instance ID. Memory initialization programming allows a selectable range of memory to be initialized with a programmable data value. This command MUST be followed by BIST_MEM_INIT_STATUS to get the results of the BIST memory initialization operation and resume normal operation of the memory controller. [Inputs] N/A CMD_PARAM_0 [5:0]: BIST_ADDR_SPACE [5:0] Used in BIST data checking to define the address space in bytes from 0 to 2addr_space that the BIST logic checks. As an example, if the BIST_ADDR_SPACE parameter was programmed to 0x1c, then the BIST logic would check 228 bytes = 256 MBytes. Note: A BIST test must cover a minimum of 2 bursts. Therefore, you must program this parameter to a value such that the start address and end address of the BIST test encompass a minimum of 2 bursts. Only used if BIST_FULL_MEM is 'b0. CMD_PARAM_0 [6:6]: BIST_FULL_MEM Defines whether to perform BIST on full memory or on the specified address range. 'b0 – BIST performed on address range specified using BIST_START_ADDR and BIST_ADDR_SPACE

continued...

CMD_REQ	Description
	'b1 – BIST performed on full memory address range CMD_PARAM_1 [31:0]: BIST_START_ADDR [31:0] CMD_PARAM_2 [5:0]: BIST_START_ADDR [37:32] Used in BIST data checking and memory initialization programming to define the starting address for BIST checking in bytes. Only used if BIST_FULL_MEM is 'b0. CMD_PARAM_3: BIST_DATA_PATTERN Specifies the data pattern to use for the memory initialization. 'b00 – Initialize memory to all zeros. 'b10 – Use data pattern specified using the values set using commands BIST_SET_DATA_PATTERN_UPPER and BIST_SET_DATA_PATTERN_LOWER before issuing BIST_MEM_INITIAL_START command. [Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: BIST_INITIATED 'b1 – BIST memory initialization initiated successfully 'b0 – BIST memory initialization initiation failed [Command-Specific Errors] 'b00 – No errors 'b01 – A previous command's saved state not restored. For example, BIST_STANDARD_MODE_START should be followed by BIST_RESULTS_STATUS command to restored saved state and resume normal memory controller operation.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_MEM_INIT_STATUS**	Command to get BIST memory initialization status for the previously initiated BIST operation for memory interface specified using the instance ID [Inputs] N/A [Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: BIST_STATUS Holds the status of the BIST operation. 'b0 = BIST operation still in progress if previously initiated. 'b1 = BIST operation has been completed. [Command-Specific Errors] 'b00 – No errors 'b01 – Could not restore saved state. BIST_MEM_INIT_START command MUST be immediately followed by BIST_MEM_INIT_STATUS to get the results of the BIST operation, restore saved state, and resume normal operation of the memory controller. The populated results, if any, may be invalid.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_SET_DATA_PATTERN_UPPER**	CMD_PARAM_3 [31:0]: BIST_DATA_PATTERN [287:256] CMD_PARAM_2 [31:0]: BIST_DATA_PATTERN [255:224] CMD_PARAM_1 [31:0]: BIST_DATA_PATTERN [223:192] CMD_PARAM_0 [31:0]: BIST_DATA_PATTERN [191:160] Defines the data pattern bits [287:160] to be used. Only data corresponding to active portion of core word is used while the inactive portion is ignored. [Outputs] N/A [Command-Specific Errors] 'b00 – No errors 'b01 – Cannot set upper data pattern for slim interface.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE>	CMD_PARAM_4 [31:0]: BIST_DATA_PATTERN [159:128] CMD_PARAM_3 [31:0]: BIST_DATA_PATTERN [127:96] CMD_PARAM_2 [31:0]: BIST_DATA_PATTERN [95:64]

continued...

CMD_REQ	Description
CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_SET_DATA_PATTERN_LOWER**	CMD_PARAM_1 [31:0]: BIST_DATA_PATTERN [63:32] CMD_PARAM_0 [31:0]: BIST_DATA_PATTERN [31:0] Defines the data pattern bits [223:0] to be used. Only data corresponding to active portion of the core word is used while inactive portion is ignored.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **LP_MODE_ENTER**	Command to cause the Interface to enter a low power state. Note that other interface operations, including recalibration and mode register reads and writes, can cause automatic exits from some low-power states. [Inputs] CMD_PARAM_0[3:0]: The low power state the interface will enter 'b0001 – Active Power Down (All Protocols) 'b0010 – Active Power Down with Memory Clock Gating (LPDDR4/LPDDR5 Only) 'b0011 – Pre-Charge Power Down (All Protocols) 'b0100 – Pre-Charge Power Down with Memory Clock Gating (LPDDR4/LPDDR5 Only) 'b0101 – Self-Refresh Short (DDR4/DDR5 and LPDDR4 Only) 'b0110 – Self-Refresh Short with Memory Clock Gating (DDR4/DDR5 Only) 'b1000 – Self-Refresh Long (DDR4/DDR5 Only) 'b1001 – Self-Refresh Long with Memory Clock Gating (DDR4/DDR5 Only) 'b1010 – Self-Refresh Long with Memory Clock and Controller Clock Gating (DDR4/DDR5 Only) 'b1011 – Self-Refresh Power Down Short (LPDDR4/LPDDR5 Only) 'b1100 – Self-Refresh Power Down Short with Memory Clock Gating (LPDDR4/LPDDR5 Only) 'b1101 – Self-Refresh Power Down Long (LPDDR4/LPDDR5 Only) 'b1110 – Self-Refresh Power Down Long with Memory Clock Gating (LPDDR4/LPDDR5 Only) 'b1111 – Self-Refresh Power Down Long with Memory and Controller Clock Gating (LPDDR4/LPDDR5 Only) [Outputs] N/A [Error Codes] 'b000 – No errors 'b001 – The Selected Low Power State is Not Available for the Current Protocol 'b010 – The Selected Low Power State is invalid/Does not Exist.
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **LP_MODE_EXIT**	Command to exit any low power state. [Inputs] N/A [Outputs] N/A [Error Codes] 'b000 – No errors
CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> *CMD_REQ [28:24]:* CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> *CMD_REQ [23:16]:* CMD_TYPE = **CMD_TRIG_CONTROLLER_OP**	Command to get the Current Low Power State of the Interface. [Inputs] N/A [Outputs]

CMD_REQ	Description
CMD_REQ [15:0]: CMD_OPCODE = **LP_MODE_STATUS**	CMD_RESPONSE_DATA_SHORT[0:0]: Valid Bit. The Data in CMD_RESPONSE_DATA_0[5:0] is only valid if this bit is 'b1. 'b0 – Invalid, the Interface is currently transitioning into or out of a low power state 'b1 – Valid. CMD_RESPONSE_DATA_0[5:0]: Current Interface Low Power State 'b000000 - Idle 'b000001 – Active Power Down 'b000010 – Active Power Down with Memory Clock Gating 'b000011 – Pre-Charge Power Down 'b000100 – Pre-Charge Power Down with Memory Clock Gating 'b000101 – Self-Refresh Short 'b000110 – Self-Refresh Short with Memory Clock Gating 'b001000 – Self-Refresh Long 'b001001 – Self-Refresh Long with Memory Clock Gating 'b001010 – Self-Refresh Long with Memory Clock and Controller Clock Gating 'b001011 – Self-Refresh Power Down Short 'b001100 – Self-Refresh Power Down Short with Memory Clock Gating 'b001101 – Self-Refresh Power Down 'b001110 – Self-Refresh Power Down Long with Memory Clock Gating 'b001111 – Self-Refresh Power Down Long with Memory and Controller Clock Gating [Error Codes] 'b000 – No errors

3.3. Agilex 7 M-Series EMIF Controller

3.3.1. Hard Memory Controller

The Agilex 7 M-Series hard memory controller is designed for high speed, high performance, high flexibility, and area efficiency. The Agilex 7M-Series hard memory controller supports the DDR4, DDR5, and LPDDR5 memory standards.

The hard memory controller implements efficient pipelining techniques and advanced dynamic command and data reordering algorithms to improve bandwidth usage and reduce latency, providing a high-performance solution.

The hard memory controller consists of the following logic blocks:

- Core and PHY interfaces
- Main control path
- Data buffer controller
- Read and write data buffers

The controller user interface uses the AXI4 protocol. The controller communicates to the PHY using the DDR PHY Interface (DFI).

3.3.1.1. Hard Memory Controller Features

Table 23. Features of the Agilex 7 M-Series Hard Memory Controller

Feature	Description
Memory standards support	Supports DDR4, DDR5, and LPDDR5 SDRAM.
Memory devices support	Supports the following memory devices: • Discrete (DDR4, DDR5, LPDDR5) • UDIMM (DDR5) • SODIMM (DDR5) • RDIMM (DDR5)
Interface protocols support	• Supports the AXI4 interface.
Configurable memory interface width	• DDR4 supports DQ widths: 16, 32, 40 — Discrete component : 16, 16+ECC, 24, 32, 32+ECC, 40 — DIMM : 64, 64 + ECC, 72 • DDR5 supports DQ widths: 16, 16+ECC (1ch/2ch), 32, 32+ECC (1ch/2ch) • LPDDR5 supports DQ widths: 16 (1ch/2ch), 32 (1ch)
Maximum rank support	2 ranks with single slot.
Burst length support	• DDR4: BL8 • DDR5: BL16 • LPDDR5: BL16
Efficiency optimization features	• Open-page policy—by default, opens page on every access. However, the controller intelligently closes a row based on incoming traffic, which improves the efficiency of the controller especially for random traffic. • Pre-emptive bank management—the controller issues bank management commands early, which ensures that the required row is open when the read or write occurs. • Data reordering—the controller reorders read/write commands. • Additive latency—the controller can issue a READ/WRITE command after the ACTIVATE command to the memory bank prior to t_{RCD}, which increases the command efficiency.
Starvation counter	Ensures all requests are served before a predefined time-out period, which ensures that low priority access are not left behind while reordering data for efficiency.
Bank interleaving	Able to issue read or write commands continuously to "random" addresses. You must correctly cycle the bank addresses.
On-die termination	In DDR4, the controller controls the on-die termination signal for the memory. This feature improves signal integrity and simplifies your board design.
Refresh features	• User-controlled refresh timing—optionally, you can control when refreshes occur and this allows you to prevent important read or write operations from clashing with the refresh lock-out time. • Per-rank refresh—allows refresh for each individual rank. • Controller-controlled refresh.

continued...

Feature	Description
Power saving features	• Low power modes (power down and self-refresh)—optionally, you can request the controller to put the memory into one of the two low power states. • Automatic power down—puts the memory device in power down mode when the controller is idle. You can configure the idle waiting time. • Memory clock gating.
Memory features	• Bank group support—supports different timing parameters for between bank groups. • Command/Address parity—command and address bus parity check.
User ZQ calibration	Long or short ZQ calibration request for DDR4.

3.4. Agilex 7 M-Series EMIF IP for Hard Processor Subsystem (HPS)

The Agilex 7 M-Series FPGA EMIF IP can access external DRAM memory devices using the External Memory Interfaces for HPS Intel FPGA IP.

To enable connectivity between the HPS and the Agilex 7 M-Series EMIF IP, you must create and configure an instance of the EMIF for HPS IP, and connect it to the Agilex 7 FPGA hard processor subsystem instance in your system.

Restrictions on I/O Bank Usage for Agilex 7 M-Series EMIF IP with HPS

- Only the two IO96 banks adjacent to the HPS MPFE can be used for HPS-EMIF (bank 3C and bank 3D).
- If only one IO96 bank is to be used by HPS-EMIF, it must be the one adjacent to the HPS MPFE (bank 3D).
- No protocol's data width usage may span multiple IO96 banks. For example, a single DDR4 x64, which requires 8 byte lanes for data and 3 byte lanes for address and control, may not span two IO96 banks. However, a single DDR4 x32, which requires 4 byte lanes of data and 3 byte lanes of address and control, may be placed in one IO96 bank, and another single DDR4 x32, may be placed in another IO96 bank.
- Unused pins in an HPS-EMIF occupied IO96 bank must be left unused; you cannot use them as general-purpose I/O pins.
- Unused lanes in an HPS-EMIF occupied IO96 bank should be left unconnected; you cannot use them as general-purpose I/O pins.

- Reference clock sharing is not allowed between HPS-EMIF IP and other IPs.
- For multi-channel EMIFs or when multiple EMIFs are used inside HPS-EMIF IP, they must have identical IP parameters.
- Initiators and targets must only be connected according to the following table:

	HPS Initiator	Non-HPS Initiator	HPS Initiator-lite	Non-HPS Initiator-lite
HPS EMIF Target	Yes	No	—	—
Non-HPS EMIF Target	No	Yes	—	—
HPS EMIF Target-lite	—	—	Yes *	No
Non-HPS EMIF Target-lite	—	—	No	Yes
Note: * The Quartus Prime software may make this connection automatically.				

Table 24. IO96 Bank and Lane Usage for HPS EMIF

Data Width Usage	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR4								
DDR4x16	—	—	—	DQ[1]	AC2	AC1	AC0	DQ[0]
DDR4x16+ ECC	—	—	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
DDR4x32	—	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
DDR4x32+ ECC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
DDR4x64	Not Supported							
DDR4x64+ ECC	Not Supported							
DDR5								
DDR5x16	—	—	—	—	AC1	AC0	DQ[0]	DQ[1]
DDR5x16	DQ[1]	DQ[0]	AC1	AC0	—	—	—	—
DDR5 2chx16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[0]	DQ[1]
DDR5x16+ ECC	—	—	—	DQ[ECC]	AC1	AC0	DQ[0]	DQ[1]
DDR5x32	—	—	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
DDR5x32+ ECC	—	DQ[ECC]	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
LPDDR5								
LPDDR5x16	—	—	—	—	AC1	AC0	DQ[1]	DQ[0]

continued...

Data Width Usage	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
LPDDR5x16	DQ[1]	DQ[0]	AC1	AC0	—	—	—	—
LPDDR5 2chx16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
LPDDR5x32	DQ[3]	DQ[2]	—	—	AC1	AC0	DQ[1]	DQ[0]

4. Agilex 7 M-Series FPGA EMIF IP – End-User Signals

4.1. Agilex 7 M-Series FPGA EMIF IP Interfaces for DDR4

The interfaces in the Agilex 7 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 25. Interfaces for EMIF Architecture Component

Interface Name	Interface Type	Description
ref_clk	clock	PLL reference clock input
core_init_n	reset	An input to indicate that core configuration is complete
usr_async_clk	clock	User clock interface
usr_clk	clock	User clock interface
usr_rst_n	reset	User clock domain reset interface
s0_axi4	axi4	Fabric (i.e. NOC-bypass) interface to controller
mem	conduit	Interface between FPGA and external memory
oct	conduit	On-Chip Termination (OCT) interface

4.1.1. ref_clk for EMIF

PLL reference clock input

Table 26. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	input	PLL reference clock input. **PLL reference clock jitter** specifies the peak-to-peak jitter on the PLL reference clock source. The clock source of the PLL reference clock must meet or exceed the following jitter requirements: 10ps peak to peak, or 1.42ps RMS at 1e-12 BER, 1.22ps at 1e-16 BER.

4.1.2. core_init_n for EMIF

An input to indicate that core configuration is complete

Table 27. Interface: core_init_n

Interface type: reset

Port Name	Direction	Description
core_init_n	input	Core init signal going into EMIF. Used to generate the reset signal on the core-EMIF interface in fabric modes.

4.1.3. usr_async_clk for EMIF

User clock interface

Table 28. Interface: usr_async_clk

Interface type: clock

Port Name	Direction	Description
usr_async_clk	input	User clock

4.1.4. usr_clk for EMIF

User clock interface

Table 29. Interface: usr_clk

Interface type: clock

Port Name	Direction	Description
usr_clk	output	User clock

4.1.5. usr_rst_n for EMIF

User clock domain reset interface

Table 30. Interface: usr_rst_n

Interface type: reset

Port Name	Direction	Description
usr_rst_n	output	User reset

4.1.6. s0_axi4 for EMIF

Fabric AXI interface to controller.

Table 31. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
Write Address (Command) Channel		
s0_axi4_awaddr	input	Write address
s0_axi4_awburst	input	Write burst type.

continued...

Port Name	Direction	Description
		• 'b00 = Reserved (FIXED is not supported). • 'b01 = INCR. • 'b10 = WRAP. • 'b11 = Reserved.
s0_axi4_awid	input	Write address ID
s0_axi4_awlen	input	Write burst length. Any value between 0 and 255 is valid, representing a transfer of 1 to 256 beats.
s0_axi4_awlock	input	Write lock type. This 2-bit signal is used to control exclusive accesses and locking. • 'b0 = Normal Access. • 'b1 = Exclusive Access.
s0_axi4_awqos	input	Write quality of service. Supported priority values range from 0 to 3, with 0 as the lowest priority.
s0_axi4_awsize	input	Write burst size. AWSIZE = 5 (32 bytes) is supported by the memory controller when the AXI port is 256 bits, and only AWSIZE = 4 (16 bytes) is supported by the memory controller when the AXI port is 128 bits.
s0_axi4_awvalid	input	Write address valid
s0_axi4_awuser	input	Write address user signal. • [0]: Enable/Disable Auto-precharge.Drive 1 to enable Auto-precharge.An auto-precharge will be issued after the write command is completed. • [1]: ALLSTRB: When all write strobes are driven (no byte enable signals not asserted), this signal can be enabled to improve controller performance. • [13:2]: Not connected. Drive 0.
s0_axi4_awprot	input	Write protection type. This 2-bit signal is used to control privileged and secure accesses. • 'b00 = Non-Privileged & Secure Access. • 'b01 = Privileged & Secure Access. • 'b10 = Non-Privileged & Non-Secure. • 'b11 = Privileged & Non-Secure.
s0_axi4_awready	output	Write address ready
Write Data Channel		
s0_axi4_wdata	input	Write data
s0_axi4_wlast	input	Write last. This signal indicates the last transfer in a write burst.
s0_axi4_wready	output	Write ready. Indicates that the AXI port is ready to accept write data.
s0_axi4_wstrb	input	Write strobes
s0_axi4_wuser	input	Write user signal. Only applicable to the x40/x72 lockstep cases. The additional user bits to be written are sent on this interface. If a x36 interface is used, then only the lowest 32-bits are connected.
s0_axi4_wvalid	input	Write valid
Write Response Channel		
s0_axi4_bready	input	Response ready
s0_axi4_bid	output	Write response ID

continued...

Port Name	Direction	Description
s0_axi4_bresp	output	Write response. A response is sent for the entire burst. • 'b00 = OKAY. Write command was successfully processed, or exclusive write command was not processed as exclusive. • 'b01 = EXOKAY. Exclusive write command was successfully processed. • 'b10 = SLVERR. Slave has received the read write command but there is an error in the transaction. • 'b11 = DECERR. Slave does not exist and/or there is an error with the transaction.
s0_axi4_bvalid	output	Write response valid.
Read Address (Command) Channel		
s0_axi4_araddr	input	Read address.
s0_axi4_arburst	input	Read burst type. • 'b00 = Reserved (FIXED is not supported). • 'b01 = INCR. • 'b10 = WRAP. • 'b11 = Reserved.
s0_axi4_arid	input	Read write address ID
s0_axi4_arlen	input	Read burst length. Any value between 0 and 128 255 is valid, representing a transfer of 1 to 256 beats.
s0_axi4_arlock	input	Read lock type. This 2-bit signal is used to control exclusive accesses and locking. • 'b0 = Normal Access. • 'b1 = Exclusive Access.
s0_axi4_arqos	input	Read quality of service Supported priority values range from 0 to 3, with 0 as the lowest priority
s0_axi4_arsize	input	Read burst size. AWSIZE = 5 (32 bytes) is supported by the memory controller when the AXI port is 256 bits, and only AWSIZE = 4 (16 bytes) is supported by the memory controller when the AXI port is 128 bits.
s0_axi4_arvalid	input	Read address valid.
s0_axi4_aruser	inout	Read address user signal. • [0]: Enable/Disable Auto-precharge. Drive 1 to enable auto-precharge.An auto-precharge will be issued after the read command is completed. • [13:1]: Not connected.Drive 0.
s0_axi4_arprot	input	Read protection type. This 2-bit signal is used to control privileged ad secure accesses. • 'b00 = Non-Privileged & Secure Access. • 'b01 = Privileged & Secure Access. • 'b10 = Non-Privileged & Non-Secure. • 'b11 = Privileged & Non-Secure.
s0_axi4_arready	output	Read address ready
Ready Data Channel		
s0_axi4_rdata	output	Read data
s0_axi4_rid	output	Read ID
s0_axi4_rlast	output	Read last. This signal indicates the last transfer in a read burst.

continued...

Port Name	Direction	Description
s0_axi4_rready	input	Read ready
s0_axi4_rresp	output	read response. A response is sent with each burst, indicating the status of that burst. • 'b00 = OKAY. Read command was successfully processed, or exclusive read command was not processed as exclusive. • 'b01 = EXOKAY. Exclusive read command was successfully processed. • 'b10 = SLVERR. Slave has received the read command but there is an error in the transaction. • 'b11 = DECERR. Slave does not exist and/or there is an error with the transaction.
s0_axi4_ruser	output	Read user signal. Only applicable to the x40/x72 lockstep cases. These are the additional user bits received on this interface. If a x36 interface is used, then only the lowest 32-bits are connected.
s0_axi4_rvalid	output	Read valid.

4.1.7. mem for EMIF

Interface between FPGA and external memory

Table 32. Interface: mem

Interface type: conduit

Port Name	Direction	Description
mem_ck_t	output	CK Clock (true)
mem_ck_c	output	CK Clock (complement)
mem_cke	output	Clock Enable
mem_odt	output	On-Die Termination
mem_cs_n	output	Chip Select
mem_c	output	Chip ID
mem_a	output	Address
mem_ba	output	Bank Address
mem_bg	output	Bank Group
mem_act_n	output	Activation Command
mem_par	output	Command/Address Parity (to DDR4 device)
mem_alert_n	input	Indicates an Address Parity and/or Write CRC Error
mem_reset_n	output	Asynchronous Reset
mem_dq	bidir	Data (read/write)
mem_dqs_t	bidir	Data Strobe (true)
mem_dqs_c	bidir	Data Strobe (complement)
mem_dbi_n	bidir	Acts as either the data bus inversion pin, or the data mask pin, depending on the configuration and whether it's a read or write transaction

4.1.8. oct for EMIF

On-Chip Termination (OCT) interface

Table 33. Interface: oct

Interface type: conduit

Port Name	Direction	Description
oct_rzqin	input	Calibrated On-Chip Termination (OCT) input pin

4.2. Agilex 7 M-Series FPGA EMIF IP Interfaces for DDR5

The interfaces in the Agilex 7 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 34. Interfaces for EMIF Architecture Component

Interface Name	Interface Type	Description
ref_clk	clock	PLL reference clock input
core_init_n	reset	An input to indicate that core configuration is complete
usr_async_clk	clock	User clock interface
usr_clk	clock	User clock interface
usr_rst_n	reset	User clock domain reset interface
s0_axi4	axi4	Fabric (i.e. NOC-bypass) interface to controller
mem	conduit	Interface between FPGA and external memory
i3c	conduit	Sideband bus interface
mem_lbd	conduit	Loopback data interface
mem_lbs	conduit	Loopback strobe interface
oct	conduit	On-Chip Termination (OCT) interface

4.2.1. ref_clk for EMIF

PLL reference clock input

Table 35. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	input	PLL reference clock input. **PLL reference clock jitter** specifies the peak-to-peak jitter on the PLL reference clock source. The clock source of the PLL reference clock must meet or exceed the following jitter requirements: 10ps peak to peak, or 1.42ps RMS at 1e-12 BER, 1.22ps at 1e-16 BER.

4.2.2. core_init_n for EMIF

An input to indicate that core configuration is complete

Table 36. Interface: core_init_n

Interface type: reset

Port Name	Direction	Description
core_init_n	input	Core init signal going into EMIF. Used to generate the reset signal on the core-EMIF interface in fabric modes.

4.2.3. usr_async_clk for EMIF

User clock interface

Table 37. Interface: usr_async_clk

Interface type: clock

Port Name	Direction	Description
usr_async_clk	input	User clock

4.2.4. usr_clk for EMIF

User clock interface

Table 38. Interface: usr_clk

Interface type: clock

Port Name	Direction	Description
usr_clk	output	User clock

4.2.5. usr_rst_n for EMIF

User clock domain reset interface

Table 39. Interface: usr_rst_n

Interface type: reset

Port Name	Direction	Description
usr_rst_n	output	User reset

4.2.6. s0_axi4 for EMIF

Fabric AXI interface to controller.

Table 40. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
Write Address (Command) Channel		
s0_axi4_awaddr	input	Write address
s0_axi4_awburst	input	Write burst type. • 'b00 = Reserved (FIXED is not supported). • 'b01 = INCR. • 'b10 = WRAP. • 'b11 = Reserved.
s0_axi4_awid	input	Write address ID
s0_axi4_awlen	input	Write burst length. Any value between 0 and 255 is valid, representing a transfer of 1 to 256 beats.
s0_axi4_awlock	input	Write lock type. This 2-bit signal is used to control exclusive accesses and locking. • 'b0 = Normal Access. • 'b1 = Exclusive Access.
s0_axi4_awqos	input	Write quality of service. Supported priority values range from 0 to 3, with 0 as the lowest priority.
s0_axi4_awsize	input	Write burst size. AWSIZE = 5 (32 bytes) is supported by the memory controller when the AXI port is 256 bits, and only AWSIZE = 4 (16 bytes) is supported by the memory controller when the AXI port is 128 bits.
s0_axi4_awvalid	input	Write address valid
s0_axi4_awuser	input	Write address user signal. • [0]: Enable/Disable Auto-precharge.Drive 1 to enable Auto-precharge.An auto-precharge will be issued after the write command is completed. • [1]: ALLSTRB: When all write strobes are driven (no byte enable signals not asserted), this signal can be enabled to improve controller performance. • [13:2]: Not connected. Drive 0.
s0_axi4_awprot	input	Write protection type. This 2-bit signal is used to control privileged and secure accesses. • 'b00 = Non-Privileged & Secure Access. • 'b01 = Privileged & Secure Access. • 'b10 = Non-Privileged & Non-Secure. • 'b11 = Privileged & Non-Secure.
s0_axi4_awready	output	Write address ready
Write Data Channel		
s0_axi4_wdata	input	Write data
s0_axi4_wlast	input	Write last. This signal indicates the last transfer in a write burst.
s0_axi4_wready	output	Write ready. Indicates that the AXI port is ready to accept write data.
s0_axi4_wstrb	input	Write strobes

continued...

Port Name	Direction	Description
s0_axi4_wuser	input	Write user signal. Only applicable to the x40/x72 lockstep cases. The additional user bits to be written are sent on this interface. If a x36 interface is used, then only the lowest 32-bits are connected.
s0_axi4_wvalid	input	Write valid
Write Response Channel		
s0_axi4_bready	input	Response ready
s0_axi4_bid	output	Write response ID
s0_axi4_bresp	output	Write response. A response is sent for the entire burst. • 'b00 = OKAY. Write command was successfully processed, or exclusive write command was not processed as exclusive. • 'b01 = EXOKAY. Exclusive write command was successfully processed. • 'b10 = SLVERR. Slave has received the read write command but there is an error in the transaction. • 'b11 = DECERR. Slave does not exist and/or there is an error with the transaction.
s0_axi4_bvalid	output	Write response valid.
Read Address (Command) Channel		
s0_axi4_araddr	input	Read address.
s0_axi4_arburst	input	Read burst type. • 'b00 = Reserved (FIXED is not supported). • 'b01 = INCR. • 'b10 = WRAP. • 'b11 = Reserved.
s0_axi4_arid	input	Read write address ID
s0_axi4_arlen	input	Read burst length. Any value between 0 and 128 255 is valid, representing a transfer of 1 to 256 beats.
s0_axi4_arlock	input	Read lock type. This 2-bit signal is used to control exclusive accesses and locking. • 'b0 = Normal Access. • 'b1 = Exclusive Access.
s0_axi4_arqos	input	Read quality of service Supported priority values range from 0 to 3, with 0 as the lowest priority
s0_axi4_arsize	input	Read burst size. AWSIZE = 5 (32 bytes) is supported by the memory controller when the AXI port is 256 bits, and only AWSIZE = 4 (16 bytes) is supported by the memory controller when the AXI port is 128 bits.
s0_axi4_arvalid	input	Read address valid.
s0_axi4_aruser	inout	Read address user signal. • [0]: Enable/Disable Auto-precharge. Drive 1 to enable auto-precharge.An auto-precharge will be issued after the read command is completed. • [13:1]: Not connected.Drive 0.
s0_axi4_arprot	input	Read protection type. This 2-bit signal is used to control privileged ad secure accesses.

continued...

Port Name	Direction	Description
		• 'b00 = Non-Privileged & Secure Access. • 'b01 = Privileged & Secure Access. • 'b10 = Non-Privileged & Non-Secure. • 'b11 = Privileged & Non-Secure.
s0_axi4_arready	output	Read address ready
Ready Data Channel		
s0_axi4_rdata	output	Read data
s0_axi4_rid	output	Read ID
s0_axi4_rlast	output	Read last. This signal indicates the last transfer in a read burst.
s0_axi4_rready	input	Read ready
s0_axi4_rresp	output	read response. A response is sent with each burst, indicating the status of that burst. • 'b00 = OKAY. Read command was successfully processed, or exclusive read command was not processed as exclusive. • 'b01 = EXOKAY. Exclusive read command was successfully processed. • 'b10 = SLVERR. Slave has received the read command but there is an error in the transaction. • 'b11 = DECERR. Slave does not exist and/or there is an error with the transaction.
s0_axi4_ruser	output	Read user signal. Only applicable to the x40/x72 lockstep cases. These are the additional user bits received on this interface. If a x36 interface is used, then only the lowest 32-bits are connected.
s0_axi4_rvalid	output	Read valid.

4.2.7. mem for EMIF

Interface between FPGA and external memory

Table 41. Interface: mem

Interface type: conduit

Port Name	Direction	Description
mem_ck_t	output	CK Clock (true)
mem_ck_c	output	CK Clock (complement)
mem_reset_n	output	Asynchronous Reset
mem_cs_n	output	Chip Select
mem_ca	output	Command/Address Bus
mem_par	output	Command/Address Parity
mem_dq	bidir	Data (read/write)
mem_dqs_t	bidir	Data Strobe (true)

continued...

Port Name	Direction	Description
mem_dqs_c	bidir	Data Strobe (complement)
mem_dm_n	output	Data Mask
mem_alert_n	input	Indicates Write CRC Error

4.2.8. i3c for EMIF

Sideband bus interface

Table 42. Interface: i3c

Interface type: conduit

Port Name	Direction	Description
i3c_scl	output	Serial Clock
i3c_sda	bidir	Serial Data

4.2.9. mem_lbd for EMIF

Loopback data interface

Table 43. Interface: mem_lbd

Interface type: conduit

Port Name	Direction	Description
mem_lbd	input	Loopback data input pin

4.2.10. mem_lbs for EMIF

Loopback strobe interface

Table 44. Interface: mem_lbs

Interface type: conduit

Port Name	Direction	Description
mem_lbs	input	Loopback strobe input pin

4.2.11. oct for EMIF

On-Chip Termination (OCT) interface

Table 45. Interface: oct

Interface type: conduit

Port Name	Direction	Description
oct_rzqin	input	Calibrated On-Chip Termination (OCT) input pin

4.3. Agilex 7 M-Series FPGA EMIF IP Interfaces for LPDDR5

The interfaces in the Agilex 7 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 46. Interfaces for EMIF Architecture Component

Interface Name	Interface Type	Description
ref_clk	clock	PLL reference clock input
core_init_n	reset	An input to indicate that core configuration is complete
usr_async_clk	clock	User clock interface
usr_clk	clock	User clock interface
usr_rst_n	reset	User clock domain reset interface
s0_axi4	axi4	Fabric (i.e. NOC-bypass) interface to controller
mem	conduit	Interface between FPGA and external memory
oct	conduit	On-Chip Termination (OCT) interface

4.3.1. ref_clk for EMIF

PLL reference clock input

Table 47. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	input	PLL reference clock input. **PLL reference clock jitter** specifies the peak-to-peak jitter on the PLL reference clock source. The clock source of the PLL reference clock must meet or exceed the following jitter requirements: 10ps peak to peak, or 1.42ps RMS at 1e-12 BER, 1.22ps at 1e-16 BER.

4.3.2. core_init_n for EMIF

An input to indicate that core configuration is complete

Table 48. Interface: core_init_n

Interface type: reset

Port Name	Direction	Description
core_init_n	input	Core init signal going into EMIF. Used to generate the reset signal on the core-EMIF interface in fabric modes.

4.3.3. usr_async_clk for EMIF

User clock interface

Table 49. Interface: usr_async_clk

Interface type: clock

Port Name	Direction	Description
usr_async_clk	input	User clock

4.3.4. usr_clk for EMIF

User clock interface

Table 50. Interface: usr_clk

Interface type: clock

Port Name	Direction	Description
usr_clk	output	User clock

4.3.5. usr_rst_n for EMIF

User clock domain reset interface

Table 51. Interface: usr_rst_n

Interface type: reset

Port Name	Direction	Description
usr_rst_n	output	User reset

4.3.6. s0_axi4 for EMIF

Fabric AXI interface to controller.

Table 52. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
Write Address (Command) Channel		
s0_axi4_awaddr	input	Write address
s0_axi4_awburst	input	Write burst type. • 'b00 = Reserved (FIXED is not supported). • 'b01 = INCR. • 'b10 = WRAP. • 'b11 = Reserved.
s0_axi4_awid	input	Write address ID
s0_axi4_awlen	input	Write burst length. Any value between 0 and 255 is valid, representing a transfer of 1 to 256 beats.
s0_axi4_awlock	input	Write lock type. This 2-bit signal is used to control exclusive accesses and locking. • 'b0 = Normal Access. • 'b1 = Exclusive Access.

continued...

Port Name	Direction	Description
s0_axi4_awqos	input	Write quality of service. Supported priority values range from 0 to 3, with 0 as the lowest priority.
s0_axi4_awsize	input	Write burst size. AWSIZE = 5 (32 bytes) is supported by the memory controller when the AXI port is 256 bits, and only AWSIZE = 4 (16 bytes) is supported by the memory controller when the AXI port is 128 bits.
s0_axi4_awvalid	input	Write address valid
s0_axi4_awuser	input	Write address user signal. • [0]: Enable/Disable Auto-precharge.Drive 1 to enable Auto-precharge.An auto-precharge will be issued after the write command is completed. • [1]: ALLSTRB: When all write strobes are driven (no byte enable signals not asserted), this signal can be enabled to improve controller performance. • [13:2]: Not connected. Drive 0.
s0_axi4_awprot	input	Write protection type. This 2-bit signal is used to control privileged and secure accesses. • 'b00 = Non-Privileged & Secure Access. • 'b01 = Privileged & Secure Access. • 'b10 = Non-Privileged & Non-Secure. • 'b11 = Privileged & Non-Secure.
s0_axi4_awready	output	Write address ready
Write Data Channel		
s0_axi4_wdata	input	Write data
s0_axi4_wlast	input	Write last. This signal indicates the last transfer in a write burst.
s0_axi4_wready	output	Write ready. Indicates that the AXI port is ready to accept write data.
s0_axi4_wstrb	input	Write strobes
s0_axi4_wuser	input	Write user signal. Only applicable to the x40/x72 lockstep cases. The additional user bits to be written are sent on this interface. If a x36 interface is used, then only the lowest 32-bits are connected.
s0_axi4_wvalid	input	Write valid
Write Response Channel		
s0_axi4_bready	input	Response ready
s0_axi4_bid	output	Write response ID
s0_axi4_bresp	output	Write response. A response is sent for the entire burst. • 'b00 = OKAY. Write command was successfully processed, or exclusive write command was not processed as exclusive. • 'b01 = EXOKAY. Exclusive write command was successfully processed. • 'b10 = SLVERR. Slave has received the read write command but there is an error in the transaction. • 'b11 = DECERR. Slave does not exist and/or there is an error with the transaction.
s0_axi4_bvalid	output	Write response valid.
Read Address (Command) Channel		

continued...

Port Name	Direction	Description
s0_axi4_araddr	input	Read address.
s0_axi4_arburst	input	Read burst type. • 'b00 = Reserved (FIXED is not supported). • 'b01 = INCR. • 'b10 = WRAP. • 'b11 = Reserved.
s0_axi4_arid	input	Read write address ID
s0_axi4_arlen	input	Read burst length. Any value between 0 and 128 255 is valid, representing a transfer of 1 to 256 beats.
s0_axi4_arlock	input	Read lock type. This 2-bit signal is used to control exclusive accesses and locking. • 'b0 = Normal Access. • 'b1 = Exclusive Access.
s0_axi4_arqos	input	Read quality of service Supported priority values range from 0 to 3, with 0 as the lowest priority
s0_axi4_arsize	input	Read burst size. AWSIZE = 5 (32 bytes) is supported by the memory controller when the AXI port is 256 bits, and only AWSIZE = 4 (16 bytes) is supported by the memory controller when the AXI port is 128 bits.
s0_axi4_arvalid	input	Read address valid.
s0_axi4_aruser	inout	Read address user signal. • [0]: Enable/Disable Auto-precharge. Drive 1 to enable auto-precharge.An auto-precharge will be issued after the read command is completed. • [13:1]: Not connected.Drive 0.
s0_axi4_arprot	input	Read protection type. This 2-bit signal is used to control privileged ad secure accesses. • 'b00 = Non-Privileged & Secure Access. • 'b01 = Privileged & Secure Access. • 'b10 = Non-Privileged & Non-Secure. • 'b11 = Privileged & Non-Secure.
s0_axi4_arready	output	Read address ready
Ready Data Channel		
s0_axi4_rdata	output	Read data
s0_axi4_rid	output	Read ID
s0_axi4_rlast	output	Read last. This signal indicates the last transfer in a read burst.
s0_axi4_rready	input	Read ready
s0_axi4_rresp	output	read response. A response is sent with each burst, indicating the status of that burst.

continued...

Port Name	Direction	Description
		• 'b00 = OKAY. Read command was successfully processed, or exclusive read command was not processed as exclusive. • 'b01 = EXOKAY. Exclusive read command was successfully processed. • 'b10 = SLVERR. Slave has received the read command but there is an error in the transaction. • 'b11 = DECERR. Slave does not exist and/or there is an error with the transaction.
s0_axi4_ruser	output	Read user signal. Only applicable to the x40/x72 lockstep cases. These are the additional user bits received on this interface. If a x36 interface is used, then only the lowest 32-bits are connected.
s0_axi4_rvalid	output	Read valid.

4.3.7. mem for EMIF

Interface between FPGA and external memory

Table 53. Interface: mem

Interface type: conduit

Port Name	Direction	Description
mem_ck_t	output	CK Clock (true)
mem_ck_c	output	CK Clock (complement)
mem_reset_n	output	Asynchronous Reset
mem_cs	output	Chip Select
mem_ca	output	Command/Address Bus
mem_dq	bidir	Data (read/write)
mem_wck_t	output	Write Clock (true)
mem_wck_c	output	Write Clock (complement)
mem_rdqs_t	bidir	Read Data Strobe (true)
mem_rdqs_c	bidir	Read Data Strobe (complement)
mem_dmi	bidir	Data Mask/Data Inversion

4.3.8. oct for EMIF

On-Chip Termination (OCT) interface

Table 54. Interface: oct

Interface type: conduit

Port Name	Direction	Description
oct_rzqin	input	Calibrated On-Chip Termination (OCT) input pin

4.4. Agilex 7 M-Series FPGA EMIF IP Interfaces for EMIF Calibration Component

The interfaces in the Agilex 7 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 55. Interfaces for EMIF Calibration Inner Component

Interface Name	Interface Type	Description
s0_axi4lite_clk	clock	Axilite clock interface
s0_axi4lite_rst_n	reset	Axilite reset interface
s0_axi4lite	axi4lite	Fabric (i.e. NOC-bypass) axilite interface to the IOSSM, including the EMIF mailbox and the calbus bridge

4.4.1. s0_axi4lite_clk for EMIF

Axilite clock interface

Table 56. Interface: s0_axi4lite_clk

Interface type: clock

Port Name	Direction	Description
s0_axi4lite_clk	input	Axilite clock

4.4.2. s0_axi4lite_rst_n for EMIF

Axilite reset interface

Table 57. Interface: s0_axi4lite_rst_n

Interface type: reset

Port Name	Direction	Description
s0_axi4lite_rst_n	input	Axilite reset

4.4.3. s0_axi4lite for EMIF

Fabric (i.e. NOC-bypass) axilite interface to the IOSSM, including the EMIF mailbox and the calbus bridge

Table 58. Interface: s0_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s0_axi4lite_awaddr	input	Write Address
s0_axi4lite_awvalid	input	Write Address Valid
s0_axi4lite_awready	output	Write Address Ready

continued...

Port Name	Direction	Description
s0_axi4lite_wdata	input	Write Data
s0_axi4lite_wstrb	input	Write Strobes
s0_axi4lite_wvalid	input	Write Valid
s0_axi4lite_wready	output	Write Ready
s0_axi4lite_bresp	output	Write Response
s0_axi4lite_bvalid	output	Write Response Valid
s0_axi4lite_bready	input	Response Ready
s0_axi4lite_araddr	input	Read Address
s0_axi4lite_arvalid	input	Read Address Valid
s0_axi4lite_arready	output	Read Address Ready
s0_axi4lite_rdata	output	Read Data
s0_axi4lite_rresp	output	Read Response
s0_axi4lite_rvalid	output	Read Valid
s0_axi4lite_rready	input	Read Ready
s0_axi4lite_awprot	input	Write Protection Type
s0_axi4lite_arprot	input	Read Protection Type

5. Agilex 7 M-Series FPGA EMIF IP – Simulating Memory IP

To simulate your design you require the following components:

- A simulator—The simulator must be an Intel-supported Verilog HDL simulator:
  - Siemens EDA* ModelSim
  - Siemens EDA QuestaSim*
  - Synopsys* VCS/VCS-MX
- A design using Intel's External Memory Interface (EMIF) IP
- An example driver or traffic generator (to initiate read and write transactions)
- A testbench and a suitable memory simulation model

The Intel External Memory Interface IP is not compatible with the Platform Designer Testbench System. Instead, use the simulation design example from your generated IP to validate memory interface operation, or as a reference for creating a full simulatable design. The provided simulation design example contains the generated memory interface, a memory model, and a traffic generator. For more information about the EMIF simulation design example, refer to the *External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide*.

Memory Simulation Models

There are two types of memory simulation models that you can use:

- Intel-provided generic memory model
- Vendor-specific memory model

The Quartus Prime software generates the generic memory simulation model with the simulation design example. The model adheres to all the memory protocol specifications, and can be parameterized.

Vendor-specific memory models are simulation models for specific memory components from memory vendors such as Micron and Samsung. You can obtain these simulation models from the memory vendor's website.

Note: Intel does not provide support for vendor-specific memory models.

5.1. Simulation Walkthrough

Simulation is a good way to determine the latency of your system. However, the latency reflected in simulation may be different than the latency found on the board because functional simulation does not take into account board trace delays and different process, voltage, and temperature scenarios.

A given design may display different latency values on different boards, due to differences in board implementation.

The Agilex 7 M-Series EMIF IP supports functional simulation through the design example using the traffic generator IP.

To perform functional simulation for an Agilex 7 M-Series EMIF IP design example, locate the design example files in the design example directory.

You can use the IP functional simulation model with any supported VHDL or Verilog HDL simulator.

After you have generated the memory IP, you can locate multiple file sets for various supported simulations in the `sim/ed_sim` subdirectory. For more information about the EMIF simulation design example, refer to the *External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide*.

5.1.1. Calibration

Calibration occurs shortly after the memory device is initialized, to compensate for uncertainties in the hardware system, including silicon PVT variation, circuit board trace delays, and skewed arrival times. The Agilex 7 M-Series FPGA EMIF IP provides skip calibration mode for simulating the design example.

Skip Calibration Mode

In Skip Calibration mode, the calibration processor assumes an ideal hardware environment, where PVT variations, board delays, and trace skews are all zero. Instead of running the actual calibration routine, the calibration processor calculates the expected arrival time of read data based on the memory latency values entered during EMIF IP generation, resulting in reduced simulation time. Skip calibration mode is recommended for use during system development, because it allows you to focus on interacting with the controller and optimizing your memory access patterns, thus facilitating rapid RTL development.

If you enable Skip Calibration Mode, the interface still performs some memory initialization, sending DRAM Mode Register Set (MRS) commands, or commands to program register code words for RDIMM/LRDIMM, before starting normal operation. These initialization commands are necessary to set up the memory model operation and latencies.

5.1.2. Simulation Scripts

The Quartus Prime software generates simulation scripts during project generation for several different third party simulation tools—Cadence, Synopsys, and Siemens EDA.

The simulation scripts are located under the `sim/ed_sim` directory, in separate folders named after each supported simulator.

5.1.3. Functional Simulation with Verilog HDL

Simulation scripts for the Synopsys and Siemens EDA simulators are provided for you to run the design example.

The simulation scripts are located in the following main folder locations:

Simulation scripts in the simulation folders are located as follows:

- `sim\ed_sim\mentor\msim_setup.tcl`
- `sim\ed_sim\synopsys\vcs\vcs_setup.sh`
- `sim\ed_sim\synopsys\vcsmx\vcsmx_setup.sh`

For more information about simulating Verilog HDL or VHDL designs using command lines, refer to the *Questa - Intel FPGA Edition, ModelSim, and QuestaSim Simulator Support* chapter in the Quartus Prime Pro Edition User Guide, Third-party Simulation.

5.1.4. Simulating the Design Example

This topic describes how to simulate the design example in Synopsys, and Siemens EDA simulators.

To simulate the example design in the Quartus Prime software using the Synopsys simulator, follow these steps:

1. At the Linux* shell command prompt, change directory to `sim\ed_sim\synopsys\vcsmx`
2. Run the simulation by typing the following command at the command prompt:

```
sh vcsmx_setup.sh
```

To simulate the example design in the Quartus Prime software using the Siemens EDA simulator, follow these steps:

1. At the Linux or Windows shell command prompt, change directory to `sim\ed_sim\mentor`
2. Execute the **msim_setup.tcl** script that automatically compiles and runs the simulation by typing the following command at the Linux or Windows command prompt:

```
vsim -do msim_setup.tcl
```

   or

   Type the following command at the ModelSim* command prompt:

```
do msim_setup.tcl
```

3. Type the command **ld_debug**. When this command completes, you can select the desired signal into the waveform.
4. Type **run-all** to run the simulation.

For more information about simulating the external memory interface using the Siemens EDA simulator, refer to the *Simulating External Memory Interface IP With ModelSim* chapter in the *External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide*.

Note: Intel does not provide the `run.do` file for the example design with the EMIF interface.

For more information about simulation, refer to the *Quartus Prime Pro Edition User Guide, Third-party Simulation*.

If your Quartus Prime project appears to be configured correctly but the example testbench still fails, check the known issues on the Intel FPGA Knowledge Base before filing a service request.

6. Agilex 7 M-Series FPGA EMIF IP – DDR4 Support

This chapter contains IP parameter descriptions, pin planning information, and board design guidelines for Agilex 7 M-Series FPGA external memory interface IP for DDR4.

6.1. Agilex 7 M-Series FPGA EMIF IP Parameters for DDR4

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP.

Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

6.1.1. Agilex 7 FPGA EMIF IP Parameter for DDR4

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 59. Group: General IP Parameters / High-Level Parameters

Display Name	Description
Technology Generation	Denotes the specific memory technology generation to be used Note: This parameter can be auto-computed. (Identifier: MEM_TECHNOLOGY)
Memory Format	Specifies the packaging format of the memory device (Identifier: MEM_FORMAT)
Memory Device Topology	Topology used by memory device (Identifier: MEM_TOPOLOGY)
Memory Ranks	Total number of physical ranks in the interface (Identifier: MEM_NUM_RANKS)
Number of Channels	Number of Channels (Identifier: MEM_NUM_CHANNELS)
Device DQ Width	If the device is a DIMM: Specifies the full DQ width of the DIMM. If the interface is composed of discrete components: Specifies the DQ width of each discrete component. (Identifier: MEM_DEVICE_DQ_WIDTH)
Number of Components Per Rank	Number of components per rank. If each component contains more than one rank, then set this parameter to 1. (Identifier: MEM_COMPS_PER_RANK)

continued...

Display Name	Description
Force Ranks to Share One Memory Interface Clock	Specifies whether all the ranks in the same channel should share one pair of memory interface differential clock. Applicable to DDR4 only. (Identifier: MEM_RANKS_SHARE_CLOCKS)
Command-Address Mirroring	Enable command-address mirroring for multi-rank DDR4 interfaces per JEDEC Standard No. 21C. Applicable to DDR4 only. The default value might not be suitable for DIMM with dual-die package DDR4 components. Note: This parameter can be auto-computed. (Identifier: MEM_AC_MIRRORING)
ECC Mode	Specifies the type of ECC (if any) and the required number of side-band bits per channel that will be used by this EMIF instance. While not all required side-band bits necessarily carry ECC bits, all need to be connected to the memory device. If enabling ECC requires more side-band bits than necessary ECC bits, then ECC bits are transmitted on the least significant side-band bits. Note: This parameter can be auto-computed. (Identifier: CTRL_ECC_MODE)
Enable Extra DQ Byte Lane	Augment a given memory interface with 8 extra DQ bits. These extra bits are accessed via the WUSER and RUSER ports on the PHY's AXI4 interface. The AXI4 WUSER and RUSER ports are 64-bit wide. In this release, this option can only augment a given 32/64-bit DDR4 in interface configured in fabric-synchronous mode without controller generated ECC bits. Note: This parameter can be auto-computed. (Identifier: AXI4_USER_DATA_ENABLE)
Total DQ Width	(Derived Parameter) This will be the width (in bits) of the mem_dq port on the memory interface. For a component interface, it is calculated based on: (MEM_COMPS_PER_RANK * MEM_DEVICE_DQ_WIDTH + (8 bits if Side-band ECC is enabled, or 8 bits if AXI4 User Data is enabled in fabric modes, or 4 bits if AXI4 User Data is enabled in NoC mode)) * MEM_NUM_CHANNELS For a DIMM-based interface, it is just MEM_DEVICE_DQ_WIDTH + (8 bits if side-band ECC is enabled, or 8 bits if AXI4 User Data is enabled in fabric modes, or 4 bits if AXI4 User Data is enabled in NoC mode) * MEM_NUM_CHANNELS. (Identifier: MEM_TOTAL_DQ_WIDTH)
Alert_N Pin Placement	(DDR4 only) Specifies the AC lane index in which to place the ALERT_N pin. (Identifier: PHY_ALERT_N_PLACEMENT)
Minimum Number of AC Lanes for DDR4	Specifies the minimum number of AC lanes required for the memory interface. Only applicable for DDR4. (Identifier: USER_MIN_NUM_AC_LANES)
Memory Clock Frequency	Specifies the operating frequency of the memory interface in MHz. If you change the memory frequency, you must select a matching Preset from the dropdown (or create a custom one), to update all the timing parameters. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FREQ_MHZ)
Instance ID	Instance ID of the EMIF IP. EMIF in the same bank, or connected to related user logic (e.g. to the same INIU), should have unique IDs in order to distinguish them when using the side-band interface. Valid values are 0-6. (Identifier: INSTANCE_ID)

Table 60. Group: General IP Parameters / Memory Device Preset Selection

Display Name	Description
Use Memory Device Preset from file	Specifies whether MEM_PRESET_ID will be a value from Quartus (if false), or a value from a custom preset file path (if true)

continued...

Display Name	Description
	(Identifier: MEM_PRESET_FILE_EN)
Memory Preset custom file path	Path to a .qprs file on the users disk (Identifier: MEM_PRESET_FILE_QPRS)
Memory Preset	The name of a preset that the user would like to load, describing the memory device that this emif will be targeting. Note: This parameter can be auto-computed. (Identifier: MEM_PRESET_ID)

Table 61. Group: General IP Parameters / Advanced Parameters / PHY / Topology

Display Name	Description
Use NOC	Specifies whether we are using the NOC or bypassing it Note: This parameter can be auto-computed. (Identifier: PHY_NOC_EN)
Asynchronous Enable	Specifies whether the user logic is clocked based on the clock provided by the IP (Sync), or by a separate user clock (Async). If true - async mode is used, if false - sync mode is used. (Identifier: PHY_ASYNC_EN)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms IO BANK and TOP vs BOT part of the IO BANK). Legal ranges are derived from device floorplan. By default (value=AUTO), the most optimal location is selected (to maximize available frequency and data width). Note: This parameter can be auto-computed. (Identifier: PHY_AC_PLACEMENT)
PLL Reference Clock Frequency	Specifies what PLL reference clock frequency the user will supply. It is recommended to use the fastest possible PLL reference clock frequency because it leads to better jitter performance. Note: This parameter can be auto-computed. (Identifier: PHY_REFCLK_FREQ_MHZ)

Table 62. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings

Display Name	Description
Voltage	The voltage level for the I/O pins driving the signals between the memory device and the FPGA memory interface. (Identifier: PHY_IO_VOLTAGE)

Table 63. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Address/Command

Display Name	Description
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the Address/Command Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_AC_X_R_S_AC_OUTPUT_OHM)

Table 64. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Memory Clock

Display Name	Description
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the CK Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_CLK_X_R_S_CK_OUTPUT_OHM)

Table 65. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Data Bus

Display Name	Description
I/O Standard	Specifies the I/O electrical standard for the data bus pins. The selected I/O standard configures the circuit within the I/O buffer to match the industry standard. (Identifier: GRP_PHY_DATA_X_DQ_IO_STD_TYPE)
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the DQ Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_DATA_X_R_S_DQ_OUTPUT_OHM)
Slew Rate	Specifies the slew rate of the data bus pins. The slew rate (or edge rate) describes how quickly the signal can transition, measured in voltage per unit time. *Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.* (Identifier: GRP_PHY_DATA_X_DQ_SLEW_RATE)
Input Termination	This parameter allows you to change the input on chip termination settings for the selected I/O standard on the DQ Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_DATA_X_R_T_DQ_INPUT_OHM)
Initial Vrefin	Specifies the **initial value for the reference voltage on the data pins(Vrefin)**. The specified value serves as a starting point and may be overridden by calibration to provide better timing margins. (Identifier: GRP_PHY_DATA_X_DQ_VREF)

Table 66. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / PHY Inputs

Display Name	Description
PLL Reference Clock Input Termination	This parameter allows you to change the input on chip termination settings for the selected I/O standard on the refclk input pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_IN_X_R_T_REFCLK_INPUT_OHM)

Table 67. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Data Bus On-Die Termination (ODT)

Display Name	Description
Target Write Termination	Specifies the target termination to be used during a write (Identifier: GRP_MEM_ODT_DQ_X_TGT_WR)
Non-Target Write Termination	Specifies the termination to be used for the non-target rank in a multi-rank configuration during a write (Identifier: GRP_MEM_ODT_DQ_X_NON_TGT_WR)
Non-Target Read Termination	Specifies the termination to be used for the non-target rank in a multi-rank configuration during a read

continued...

Display Name	Description
	(Identifier: GRP_MEM_ODT_DQ_X_NON_TGT_RD)
Drive Strength	Specifies the termination to be used when driving read data from memory (Identifier: GRP_MEM_ODT_DQ_X_RON)

Table 68. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Data Bus Reference Voltage (Vref)

Display Name	Description
VrefDQ Range	Specifies which of the memory protocol defined ranges will be used (Identifier: GRP_MEM_DQ_VREF_X_RANGE)
VrefDQ Value	Specifies the initial VrefDQ value to be used (Identifier: GRP_MEM_DQ_VREF_X_VALUE)

Table 69. Group: General IP Parameters / Advanced Parameters / AXI Settings / AXI Interface Settings

Display Name	Description
Enable Debug Tools	If enabled, the AXI-L port will be connected to SLD nodes, allowing for a system-console avalon manager interface to interact with this AXI-L subordinate interface. (Identifier: DEBUG_TOOLS_EN)
AXI-Lite Port Access Mode	Specifies whether the AXI-Lite port is connected to the fabric, the NOC, or disabled Note: This parameter can be auto-computed. (Identifier: AXI_SIDEBAND_ACCESS_MODE)

Table 70. Group: General IP Parameters / Advanced Parameters / Additional Parameters / Additional String Parameters

Display Name	Description
User Extra Parameters	Semi-colon separated list of key/value pairs of extra parameters (Identifier: USER_EXTRA_PARAMETERS)

Table 71. Group: Example Design / Example Design

Display Name	Description
HDL Selection	This option lets you choose the format of HDL in which generated simulation and synthesis files are created. You can select either Verilog or VHDL. (Identifier: EX_DESIGN_HDL_FORMAT)
Synthesis	Generate Synthesis Example Design (Identifier: EX_DESIGN_GEN_SYNTH)
Simulation	Generate Simulation Example Design (Identifier: EX_DESIGN_GEN_SIM)
Core Clock Freq	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode) Note: This parameter can be auto-computed. (Identifier: EX_DESIGN_CORE_CLK_FREQ_MHZ)
Core Refclk Freq	PLL reference clock frequency in MHz for PLL supplying the core clock

continued...

Display Name	Description
	(Identifier: EX_DESIGN_CORE_REFCLK_FREQ_MHZ)
NOC Refclk Freq	NOC Refclk Freq for the NOC control IP Note: This parameter can be auto-computed. (Identifier: EX_DESIGN_NOC_REFCLK_FREQ_MHZ)
Traffic Generator Remote Access	Specifies whether the traffic generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled (Identifier: EX_DESIGN_HYDRA_REMOTE)

Table 72. Group: Example Design / Performance Monitor

Display Name	Description
Enable performance monitoring	Enable performance monitor on all channels for measuring read/write transaction metrics (Identifier: EX_DESIGN_PMON_ENABLED)

Table 73. Group: Example Design / Traffic Generator Program

Display Name	Description
Traffic Generator Program	Specifies the traffic pattern to run. (Identifier: EX_DESIGN_HYDRA_PROG)

6.1.2. Agilex 7 FPGA EMIF Memory Device Description IP (DDR4) Parameter Descriptions

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 74. Group: Configuration Save

Display Name	Description
Configuration Filepath	Filepath to Save to (.qprs extension) (Identifier: MEM_CONFIG_FILE_QPRS)

Table 75. Group: High-Level Parameters

Display Name	Description
Memory Format	Specifies the packaging format of the memory device (Identifier: MEM_FORMAT)
Number of Ranks per DIMM	Number of ranks per DIMM Note: This parameter can be auto-computed. (Identifier: MEM_RANKS_PER_DIMM)
DRAM Component Package Type	Specifies the packaging type of each memory component used in the interface. (Identifier: DDR4_MEM_DEVICE_PACKAGE)
Density of Each Memory Die	Specifies the density of each memory die on the device in Gb. (Identifier: DDR4_MEM_DEVICE_DIE_DENSITY_GBITS)
Density of Each Memory Component	Specifies the density of each memory component in Gb.

continued...

Display Name	Description
	(Identifier: DDR4_MEM_DEVICE_COMPONENT_DENSITY_GBITS)
Enable Read DBI	Specifies whether read DBI is enabled. Read DBI is only supported on DDR4 discrete components with x8 or x16 DQ width. (Identifier: DDR4_MEM_DEVICE_READ_DBI_EN)
Write DBI and Data Mask	Specify the write DBI and data mask setting. Neither write DBI nor data mask is supported on DDR4 components with x4 DQ width. (Identifier: DDR4_MEM_DEVICE_DM_WRITE_DBI)
Enable Address-Command Parity	Specifies whether address-command parity is enabled. If enabled then command latency is increased by the value of parameter "Address-Command Latency Mode". (Identifier: DDR4_MEM_DEVICE_AC_PARITY_EN)

Table 76. Group: Memory Interface Parameters / Data Bus

Display Name	Description
Device DQ Width	If the device is a DIMM: Specifies the full DQ width of the DIMM. If the interface is composed of discrete components: Specifies the DQ width of each discrete component. (Identifier: MEM_DEVICE_DQ_WIDTH)
Device Die DQ Width	The data width of each DDR4 SDRAM die. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_DIE_DQ_WIDTH)
DQ Pins per Component	Specifies the total number of DQ pins per memory component. Must be either 4, 8, or 16. (Identifier: DDR4_MEM_DEVICE_COMPONENT_DQ_WIDTH)
Burst Length	Specifies burst length of the device in transfers. (Identifier: DDR4_MEM_DEVICE_BURST_LENGTH)

Table 77. Group: Memory Interface Parameters / Device Topology

Display Name	Description
Device Bank Group Width	Specifies the number of bank group pins. Automatically derived from the number of data pins per component. (Identifier: DDR4_MEM_DEVICE_BANK_GROUP_ADDR_WIDTH)
Device Bank Address Width	Specifies the number of bank address pins. Automatically set to 2. (Identifier: DDR4_MEM_DEVICE_BANK_ADDR_WIDTH)
Device Row Address Width	Specifies the number of row address pins. Automatically derived from the device density and the number of data pins per component. (Identifier: DDR4_MEM_DEVICE_ROW_ADDR_WIDTH)
Device Column Address Width	Specifies the number of column address pins. Automatically set to 10. (Identifier: DDR4_MEM_DEVICE_COL_ADDR_WIDTH)
Number of Differential Memory Clock Pairs	Specifies the number of CK_t/CK_c clock pairs exposed by the memory interface. Usually more than one pair is required for RDIMM/LRDIMM formats. The value of this parameter depends on the memory device selected. Please refer to the datasheet for your memory device. (Identifier: DDR4_MEM_DEVICE_CK_WIDTH)

Table 78. Group: Memory Timing Parameters / Timing Parameters

Display Name	Description
Memory Clock Frequency	Specifies the **operating frequency** of the memory interface in MHz. If you change the memory frequency, you must select a matching Preset from the dropdown (or create a custom one), to update all the timing parameters. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FREQ_MHZ)
Memory Speed Bin	Specifies the memory speed bin using the bin names defined in JEDEC Standard No. 79-4D Chapter 10. (Identifier: DDR4_MEM_DEVICE_SPEEDBIN)
Memory Read Latency	Specifies the read latency of the memory interface in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_CL_CYC)
Memory Write Latency	Specifies the write latency of the memory interface in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_CWL_CYC)

Table 79. Group: Memory Timing Parameters / Advanced Timing Parameters

Display Name	Description
tREFI	Specifies the average refresh interval in microseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TREFI_US)
tRAS	Specifies the activation-to-precharge command period in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRAS_NS)
tRCD	Specifies the activation to internal read or write delay interval in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRCD_NS)
tRP	Specifies the precharge command period in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRP_NS)
tRC	Specifies the activate-to-activate or activate-to-refresh command period in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRC_NS)
tCCD_L	Specifies the CAS-to-CAS command delay for the same bank group in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCCD_L_CYC)
tCCD_S	Specifies the CAS-to-CAS command delay for different bank groups in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCCD_S_CYC)
tRRD_L	Specifies the activation-to-activation command delay for the same bank group in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRRD_L_CYC)
tRRD_S	Specifies the activation-to-activation command delay for different bank groups in cycles.

continued...

Display Name	Description
	Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRRD_S_CYC)
tFAW	Specifies the four-activate-window in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TFAW_NS)
tWTR_L	Specifies the minimum delay from the start of an internal write transaction to the immediately next internal read command for the same bank group in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TWTR_L_CYC)
tWTR_S	Specifies the minimum delay from the start of an internal write transaction to the immediately next internal read command for different bank groups in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TWTR_S_CYC)
tRTP	Specifies the internal read to precharge command delay in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRTP_CYC)
tWR	Specifies the write recovery time in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TWR_NS)
tMRD	Specifies the mode-register command cycle time in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TMRD_CYC)
tCKSRE	Specifies the number of required valid clock cycles after self-refresh entry or power-down entry. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCKSRE_CYC)
tCKSRX	Specifies the number of required valid clock cycles before self-refresh exit, power-down exit, or reset exit. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCKSRX_CYC)
tCKE	Specifies the minimum CKE pulse width in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCKE_CYC)
tCKESR	Specifies the minimum CKE low pulse width from self-refresh entry to self-refresh exit in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCKESR_CYC)
tMPRR	Specifies the multi-purpose register recovery time measured in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TMPRR_CYC)
tRFC	Specifies the refresh-to-activate or refresh-to-refresh command period. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRFC_NS)
tDIVW	Specifies the data pin receiving timing window in UI. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDIVW_TOTAL_UI)

continued...

Display Name	Description
tDQSCK	Specifies the minimum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in picoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDQSCK_PS)
tDQSQ	Specifies the latest valid transition of the associated DQ pins for a READ. tDQSQ specifically refers to the DQS_t/DQS_c to DQ skew. It is the length of time between the DQS_t/DQS_c crossing to the last valid transition of the slowest DQ pin in the DQ group associated with that DQS strobe. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDQSQ_UI)
tDQSS	Specifies the skew between the memory clock (CK) and the output data strobes used for writes in cycles. It is the time between the rising data strobe edge (DQS_t/DQS_c). Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDQSS_CYC)
tDSH	Specifies the write DQS hold time. This is the time difference between the rising CK edge and the falling edge of DQS, measured as a percentage of tCK. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDSH_CYC)
tDSS	Describes the time between the falling edge of DQS to the rising edge of the next CK transition. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDSS_CYC)
tDWVp	Specifies the data valid window per device per pin measured in terms of UI. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TDVWP_UI)
tIH (Base) DC Level	Refers to the voltage level which the address/command signal must not cross during the hold window in mV. The signal is considered stable only if it remains above this voltage level (for a logic 1) or below this voltage level (for a logic 0) for the entire hold period. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TIH_DC_MV)
tIH (Base)	Refers to the hold time for the Address/Command bus after the rising edge of CK in picoseconds. Depending on what AC level the user has chosen for a design, the hold margin can vary (this variance will be automatically determined when the user chooses the "tIH (base) AC level"). Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TIH_PS)
tIS (Base) AC Level	Refers to the voltage level which the address/command signal must cross and remain above during the setup margin window in mV. The signal is considered stable only if it remains above this voltage level (for a logic 1) or below this voltage level (for a logic 0) for the entire setup period. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TIS_AC_MV)
tIS (Base)	Refers to the setup time for the Address/Command/Control bus to the rising edge of CK in picoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TIS_PS)
tQH	Specifies the output hold time for the DQ in relation to DQS_t/DQS_c in UI. It is the length of time between the DQS_t/DQS_c pair crossing to the earliest invalid transition of the fastest DQ pin in the DQ group associated with that DQS strobe. Note: This parameter can be auto-computed.

continued...

Display Name	Description
	(Identifier: DDR4_MEM_DEVICE_TQH_UI)
tQSH	Specifies the write DQS hold time in cycles. This is the time difference between the rising CK edge and the falling edge of DQS, measured as a percentage of tCK. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TQSH_CYC)
tWLH	Describes the write leveling hold time in cycles. It is measured from the rising edge of DQS to the rising edge of CK. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TWLH_CYC)
tWLS	Describes the write leveling setup time. It is measured from the rising edge of CK to the rising edge of DQS. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TWLS_CYC)
tDiVW_total	Describes the minimum horizontal width of the DQ eye opening required by the receiver (memory device/DIMM). It is measured in UI. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_VDIVW_TOTAL_MV)
tRFC_DLR	Specifies the refresh cycle time across different logical rank in nanoseconds. Only applicable to 3DS devices. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRFC_DLR_NS)
tRRD_DLR	Specifies the activation-to-activation time across different logical rank in nanoseconds. Only applicable to 3DS devices. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TRRD_DLR_CYC)
tFAW_DLR	Specifies the four-activate-window across different logical ranks in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TFAW_DLR_NS)
tCCD_DLR	Specifies the CAS-to-CAS delay across different logical ranks in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCCD_DLR_NS)
tXP	Specifies the delay from power down exit with DLL on to any valid command, or from precharge power down with with DLL frozen to commands not requiring a locked DLL. Measured in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TXP_CYC)
tXS	Specifies the delay from self refresh exit to commands not requiring a locked DLL in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TXS_NS)
tXSDLL	Specifies the delay from self refresh exit to commands requiring a locked DLL in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TXS_DLL_CYC)
tCPDED	Specifies the command pass disable delay measured in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TCPDED_CYC)

continued...

Display Name	Description
tMOD	Specifies the mode register set command update delay in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TMOD_CYC)
tZQCS	Specifies the normal operation short calibration time in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TZQCS_CYC)
tZQINIT	Specifies the power-up and reset calibration time in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TZQINIT_CYC)
tZQOPER	Specifies the normal operation full calibration time in cycles. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_TZQOPER_CYC)

6.1.3. Agilex 7 FPGA EMIF Calibration IP Parameter Descriptions

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 80. Group: High-Level Parameters

Display Name	Description
Unique Instance ID for the Calibration IP	Instance ID (Identifier: INSTANCE_ID)
Calibration IP is part of Bank Adjacent Pair	Calibration IP is part of Bank Adjacent Pair (Identifier: IS_PART_OF_BANK_ADJACENT_PAIR)
Number of Peripheral IPs	Number of Peripheral IPs (EMIFs, PHYLites) to be calibrated (Identifier: NUM_CALBUS_PERIPHS)
Number of Standalone I/O PLLs	Number of Standalone I/O PLLs to calibrate (Identifier: NUM_CALBUS_PLLS)
AXI-L Subordinate Port Mode	AXI-L subordinate port can be disabled, or can be used in one of two modes: directly exported to fabric, or connect to the NoC (i.e. to a TNIU) (Identifier: PORT_S_AXIL_MODE)

6.2. Agilex 7 M-Series FPGA EMIF IP Pin and Resource Planning

The following topics provide guidelines on pin placement for external memory interfaces.

Typically, all external memory interfaces require the following FPGA resources:

- Interface pins
- PLL and clock network
- Other FPGA resources—for example, core fabric logic and debug interfaces

Once all the requirements are known for your external memory interface, you can begin planning your system.

6.2.1. Agilex 7 M-Series FPGA EMIF IP Interface Pins

Any I/O banks that do not support transceiver operations in Agilex 7 M-Series FPGAs support external memory interfaces. However, DQS (data strobe or data clock) and DQ (data) pins are listed in the device pin tables and are fixed at specific locations in the device. You must adhere to these pin locations to optimize routing, minimize skew, and maximize margins. Always check the pin table for the actual locations of the DQS and DQ pins.

Note: Maximum interface width varies from device to device depending on the number of I/O pins and DQS or DQ groups available. Achievable interface width also depends on the number of address and command pins that the design requires. To ensure adequate PLL, clock, and device routing resources are available, you should always test fit any IP in the Quartus Prime software before PCB sign-off.

Note: The greater the number of banks, the greater the skew, hence Intel® recommends that you always generate a test project of your desired configuration and confirm that it meets timing.

6.2.1.1. Estimating Pin Requirements

You should use the Quartus Prime software for final pin fitting. However, you can estimate whether you have enough pins for your memory interface by performing the following steps:

1. Determine how many read/write data pins are associated per data strobe or clock pair.
2. Calculate the number of other memory interface pins needed, including any other clocks (write clock or memory system clock), address, command, and RZQ. Refer to the External Memory Interface Pin Table to determine necessary Address/Command/Clock pins based on your desired configuration.
3. Calculate the total number of I/O banks required to implement the memory interface, given that an I/O bank supports up to 96 pins.

You should test the proposed pin-outs with the rest of your design in the Quartus Prime software (with the correct I/O standard and OCT connections) before finalizing the pin-outs. There can be interactions between modules that are illegal in the Quartus Prime software that you might not know about unless you compile the design and use the Quartus Prime Pin Planner.

6.2.1.2. DIMM Options

Unbuffered DIMMs (UDIMMs) require one set of chip-select (CS#), on-die termination (ODT), clock-enable (CKE), and clock pair (CK/CKn) for every physical rank on the DIMM. Many registered DIMMs use only one pair of clocks; however, this is not a universal rule, so you should check your memory vendor's data sheet to be sure. DDR4 registered DIMMs require a minimum of one chip-select signal.

Table 81. UDIMM and RDIMM Pin Options for DDR4

Pins	UDIMM Pins (Single Rank)	UDIMM Pins (Dual Rank)	RDIMM Pins (Single Rank)	RDIMM Pins (Dual Rank)
Data	72 bit DQ[71:0]=	72 bit DQ[71:0]=	72 bit DQ[71:0]=	72 bit DQ[71:0]=

continued...

Pins	UDIMM Pins (Single Rank)	UDIMM Pins (Dual Rank)	RDIMM Pins (Single Rank)	RDIMM Pins (Dual Rank)
	{CB[7:0], DQ[63:0]}	{CB[7:0], DQ[63:0]}	{CB[7:0], DQ[63:0]}	{CB[7:0], DQ[63:0]}
Data Mask	DM#/DBI#[8:0] (1)	DM#/DBI#[8:0](1)	DM#/DBI#[8:0](1)	DM#/DBI#[8:0](1)
Data Strobe	x8: DQS[8:0] and DQS#[8:0]	x8: DQS[8:0] and DQS#[8:0]	x8: DQS[8:0] and DQS#[8:0] x4: DQS[17:0] and DQS#[17:0]	x8: DQS[8:0] and DQS#[8:0] x4: DQS[17:0] and DQS#[17:0]
Address	BA[1:0], BG[1:0], A[16:0] - 4GB: A[14:0] 8GB: A[15:0] 16GB: A[16:0] (2)	BA[1:0], BG[1:0], A[16:0] - 8GB: A[14:0] 16GB: A[15:0] 32GB: A[16:0] (2)	BA[1:0], BG[1:0], x8: A[16:0] - 4GB: A[14:0] 8GB: A[15:0] 16GB: A[16:0] (2) 32GB: A[17:0] (3)	BA[1:0], BG[1:0],x8: A[16:0] x4: A[17:0] - 8GB: A[14:0] 16GB: A[15:0] 32GB: A[16:0] (2) 64GB: A[17:0] (3)
Clock	CK0/CK0#	CK0/CK0#, CK1/CK1#	CK0/CK0#	CK0/CK0#, CK1/CK1#
Command	ODT, CS#, CKE, ACT#, RAS#/A16, CAS#/A15, WE#/A14	ODT[1:0], CS#[1:0], CKE[1:0], ACT#, RAS#/A16, CAS#/A15, WE#/A14	ODT, CS#, CKE, ACT#, RAS#/A16, CAS#/A15, WE#/A14	ODT[1:0], CS#[1:0], CKE, ACT#, RAS#/A16, CAS#/A15, WE#/A14
Parity	PAR, ALERT#	PAR, ALERT#	PAR, ALERT#	PAR, ALERT#
Other Pins	SA[2:0], SDA, SCL, EVENT#, RESET#	SA[2:0], SDA, SCL, EVENT#, RESET#	SA[2:0], SDA, SCL, EVENT#, RESET#	SA[2:0], SDA, SCL, EVENT#, RESET#

Notes to Table:
1. DM/DBI pins are available only for DIMMs constructed using x8 or greater components.
2. This density requires 4Gb x4 or 2Gb x8 DRAM components.
3. This density requires 8Gb x4 DRAM components.
4. The Agilex 7 M-Series memory controller can support up to two ranks per channel. Agilex 7 M-Series supports only one DIMM per channel (1DPC).

6.2.1.3. Maximum Number of Interfaces

The maximum number of interfaces supported for a given memory protocol varies, depending on the FPGA in use.

Unless otherwise noted, the calculation for the maximum number of interfaces is based on independent interfaces where the address or command pins are not shared.

Note: You may need to share PLL clock outputs depending on your clock network usage.

For interface information for Agilex 7 M-Series devices, consult the EMIF Device Selector on www.intel.com.

Timing closure depends on device resource and routing utilization. For more information about timing closure, refer to the *Area and Timing Optimization Techniques* chapter in the *Quartus Prime Handbook*.

6.2.2. Agilex 7 M-Series FPGA EMIF IP Resources

The Agilex 7 M-Series FPGA memory interface IP uses several FPGA resources to implement the memory interface.

6.2.2.1. OCT

You require an OCT calibration block if you are using an Agilex 7 M-Series FPGA OCT calibrated series, parallel, or dynamic termination for any I/O in your design. There are two OCT blocks in an I/O bank, one for each sub-bank.

- You must observe the following requirements when using OCT blocks:
  The I/O bank where you place the OCT calibration block must use the same V_{CCIO_PIO} voltage as the memory interface.
- The OCT calibration block uses a single fixed R_{ZQ}. You must ensure that an external termination resistor is connected to the correct pin for a given OCT block.

6.2.2.2. PLL

When using PLL for external memory interfaces, you must consider the following guidelines:

- For the clock source, use the clock input pin specifically dedicated to the PLL that you want to use with your external memory interface. The input and output pins are only fully compensated when you use the dedicated PLL clock input pin. Agilex 7 M-Series devices support only differential I/O standard on dedicated PLL clock input pin for EMIF IP.
- Intel recommends using the fastest possible PLL reference clock frequency available in the drop-down list in the EMIF IP Platform Designer, because doing so provides the best jitter performance.

6.2.3. Pin Guidelines for Agilex 7 M-Series FPGA EMIF IP

The Agilex 7 M-Series FPGA contains I/O banks on the top and bottom edges of the device, which can be used by external memory interfaces.

Agilex 7 M-Series FPGA I/O banks contain 96 I/O pins. Each bank is divided into two sub-banks with 48 I/O pins in each. Sub-banks are further divided into four byte-lanes, where each byte-lane is a group of twelve I/O ports.

Agilex 7 M-Series FPGAs do not support flexible DQ group assignments. Only specific byte-lanes can be used as Address/Command lanes or data lanes. As you increase the interface width, only specific byte-lanes can be used. Refer to *Pin Placement for Agilex M-Series FPGA DDR4 IP* for more information.

The I/O bank, byte-lane, and pairing pin for every physical I/O pin can be uniquely identified by the following naming convention in the device pin table:

- The I/O pins in a bank are represented as P#, where:
  - P# represents the pin number in a bank. It ranges from P0 to P95, for 96 pins in a bank. Because an IO96 bank comprises two IO48 sub-banks, all pins with P# value less than 48 (P# <48) belong to the bottom I/O sub-bank. All other pins belong to the top IO48 sub-bank.
- The *Index Within I/O Bank* value falls within one of the following ranges: 0 to 11, 12 to 23, 24 to 35, or 36 to 47, and represents one of byte-lanes 0, 1, 2, or 3, respectively.
- To determine whether I/O banks are adjacent, you can refer to *Architecture: I/O Bank* in the *Product Architecture* chapter. In general, the two sub-banks within an I/O bank are adjacent to each other when there is at least one byte-lane in each sub-bank that is bonded out and available for EMIF use.
- The pairing pin for an I/O pin is in the same I/O bank. You can identify the pairing pin by adding 1 to its *Index Within I/O Bank* number (if it is an even number), or by subtracting 1 from its *Index Within I/O Bank* number (if it is an odd number).

6.2.4. Pin Placements for Agilex 7 M-Series FPGA DDR4 EMIF IP

6.2.4.1. Address and Command Pin Placement for DDR4

Table 82. Address and Command Pin Placement for DDR4 IP

Address/ Command Lane	Index Within Byte Lane	DDR4				
		Scheme 1	Scheme 1A	Scheme 2	Scheme 3	Scheme 3A
AC3	11	CK_C[1]	CK_C[1]	Not used by Address/ Command pins in this scheme.	CK_C[1]	CK_C[1]
	10	CK_T[1]	CK_T[1]		CK_T[1]	CK_T[1]
	9					
	8		ALERT_N			ALERT_N
	7					
	6					
	5					
	4					
	3					
	2					
	1					
	0				C[0]	C[0]
AC2	11	BG[0]	BG[0]	BG[0]	BG[0]	BG[0]
	10	BA[1]	BA[1]	BA[1]	BA[1]	BA[1]

continued...

Address/ Command Lane	Index Within Byte Lane	DDR4				
		Scheme 1	Scheme 1A	Scheme 2	Scheme 3	Scheme 3A
	9	BA[0]	BA[0]	BA[0]	BA[0]	BA[0]
	8	ALERT_N	A[17]	ALERT_N	ALERT_N	A[17]
	7	A[16]	A[16]	A[16]	A[16]	A[16]
	6	A[15]	A[15]	A[15]	A[15]	A[15]
	5	A[14]	A[14]	A[14]	A[14]	A[14]
	4	A[13]	A[13]	A[13]	A[13]	A[13]
	3	A[12]	A[12]	A[12]	A[12]	A[12]
	2	RZQ site				
	1	Differential "N-side" reference clock input site.				
	0	Differential "P-side" reference clock input site.				
AC1	11	A[11]	A[11]	A[11]	A[11]	A[11]
	10	A[10]	A[10]	A[10]	A[10]	A[10]
	9	A[9]	A[9]	A[9]	A[9]	A[9]
	8	A[8]	A[8]	A[8]	A[8]	A[8]
	7	A[7]	A[7]	A[7]	A[7]	A[7]
	6	A[6]	A[6]	A[6]	A[6]	A[6]
	5	A[5]	A[5]	A[5]	A[5]	A[5]
	4	A[4]	A[4]	A[4]	A[4]	A[4]
	3	A[3]	A[3]	A[3]	A[3]	A[3]
	2	A[2]	A[2]	A[2]	A[2]	A[2]
	1	A[1]	A[1]	A[1]	A[1]	A[1]
	0	A[0]	A[0]	A[0]	A[0]	A[0]
AC0	11	PAR[0]	PAR[0]	PAR[0]	PAR[0]	PAR[0]
	10	CS_N[1]	CS_N[1]	CS_N[1]	CS_N[1]	CS_N[1]
	9	CK_C[0]	CK_C[0]	CK_C[0]	CK_C[0]	CK_C[0]
	8	CK_T[0]	CK_T[0]	CK_T[0]	CK_T[0]	CK_T[0]
	7	CKE[1]	CKE[1]	CKE[1]	CKE[1]	CKE[1]
	6	CKE[0]	CKE[0]	CKE[0]	CKE[0]	CKE[0]
	5	ODT[1]	ODT[1]	ODT[1]	ODT[1]	ODT[1]
	4	ODT[0]	ODT[0]	ODT[0]	ODT[0]	ODT[0]
	3	ACT_N[0]	ACT_N[0]	ACT_N[0]	ACT_N[0]	ACT_N[0]
	2	CS_N[0]	CS_N[0]	CS_N[0]	CS_N[0]	CS_N[0]
	1	RESET_N[0]	RESET_N[0]	RESET_N[0]	RESET_N[0]	RESET_N[0]
	0	BG[1]	BG[1]	BG[1]	BG[1]	BG[1]

Agilex 7 M-Series FPGA DDR4 IP supports fixed Address and Command pin placement as shown in the preceding table. The IP supports up to 2 ranks for the following schemes:

- Scheme 1 supports component, UDIMM, RDIMM, and SODIMM.
- Scheme 1A supports x4 component and RDIMM with A[17] (that is, with 16Gb, x4 DQ/DQS group base component).
- Scheme 2 supports component, UDIMM, RDIMM, and SODIMM. Scheme 2 is the only scheme for HPS DDR4 EMIF, available for fabric EMIF as well.
- Schemes 3 and 3A are similar to schemes 1 and 1A. Schemes 3 and 3A support 3DS for component, UDIMM, RDIMM, and SODIMM. The maximum supported 3DS height is 2.

6.2.4.2. DDR4 Data Width Mapping

Agilex 7 M-Series devices do not support flexible data lanes placement. Only fixed byte lanes within the I/O bank can be used as data lanes. The following table lists the supported address and command and data lane placements in an I/O bank.

Table 83. DDR4 Data Width Mapping

Controller	Address / Command Scheme	Data Width Usage	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Primary	Scheme 2	DDR4 x16	GPIO [2]	GPIO [2]	GPIO [2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	Scheme 1		GPIO [2]	GPIO [2]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 1a		GPIO [2]	GPIO [2]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 3		GPIO [2]	GPIO [2]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 3a		GPIO [2]	GPIO [2]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 2	DDR4 x16 + ECC	GPIO [2]	GPIO [2]	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
	Scheme 1		GPIO [2]	DQ[ECC]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 1a		GPIO [2]	DQ[ECC]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 3		GPIO [2]	DQ[ECC]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 3a		GPIO [2]	DQ[ECC]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 2	DDR4 x32	GPIO [2]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	Scheme 1		DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3
	Scheme 1a		DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC2	AC1	AC0	AC3

continued...

Controller	Address / Command Scheme	Data Width Usage	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
	Scheme 2	DDR4 x32 + ECC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
Primary + Secondary	Scheme 2	DDR4 x40 [1]	sDQ[0]	wDQ[3]	wDQ[2]	wDQ[1]	AC2	AC1	AC0	wDQ[0]
	Scheme 2 Flip	DDR4 x40	sDQ[0]	AC0	AC2	AC1	wDQ[1]	wDQ[2]	wDW[3]	wDQ[4]

Note: 1. DDR4 x40 requires both controllers within an I/O bank in a lockstep configuration, and AXI user data.
2. GPIO – available for GPIO/PHYLite.
3. DQ[ECC] – DQ/DQS group used as ECC.

Table 84. Supported DDR4 Mapping for Lockstep Configuration for x64

A/C	A/C Placement Option	Primary								Secondary							
		BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
3	AC Pri Top Sub-Bank / Sec DQ Bot	DQ[0]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	GPIO	DQ[7]	DQ[6]	DQ[5]	DQ[4]	GPIO	GPIO	GPIO	GPIO
	AC Pri Bot Sub-Bank / Sec DQ Bot	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	GPIO	DQ[7]	DQ[6]	DQ[5]	DQ[4]	GPIO	GPIO	GPIO	GPIO
	AC Pri Top Sub-Bank / Sec DQ Top(m)	DQ[0]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	GPIO	GPIO	X	X	X	DQ[4]	DQ[5]	DQ[6]	DQ[7]
4	AC Pri Top Sub-Bank / Sec DQ Bot(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	DQ[7]	DQ[6]	DQ[5]	DQ[4]	GPIO	GPIO	GPIO	GPIO
	AC Pri Top Sub-Bank / Sec DQ Top(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	GPIO	X	X	X	DQ[4]	DQ[5]	DQ[6]	DQ[7]
	AC Pri Bot Sub-Bank / Sec DQ Top(m)	AC3	AC0	AC1	AC2	DQ[0]	DQ[1]	DQ[2]	DQ[3]	GPIO	X	X	X	DQ[4]	DQ[5]	DQ[6]	DQ[7]

Note: • GPIO = available for GPIO/PHYLite.
• X = not available for GPIO/PHYLite.

Table 85. Supported Lockstep configuration for DDR4 x72 or x64 (with ECC)

A/C	A/C Placement Option	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
	Primary									Secondary							
3	AC Pri Top Sub-Bank / Sec DQ Bot	DQ[4]	DQ[3]	DQ{2]	DQ[1]	AC1	AC2	AC0	sDQ[0]	DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
	AC Pri Bot Sub-Bank / Sec DQ Bot	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	sDQ[4]	DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
	AC Pri Top Sub-Bank / Sec DQ Top(m)	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]	GPIO	X	X	X	DQ[5]	DQ[6]	DQ[7]	DQ[8]
4	AC Pri Top Sub-Bank / Sec DQ Bot(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	DQ[8]	DQ[7]	DQ[6]	DQ[5]	X	X	GPIO	sDQ[4]
	AC Pri Top Sub-Bank / Sec DQ Top(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	DQ[4]	X	X	X	DQ[5]	DQ[6]	DQ[7]	sDQ[8]
	AC Pri Top Sub-Bank / Sec DQ Top(m)	AC3	AC0	AC1	AC2	DQ[0]	DQ[1]	DQ[2]	DQ[3]	DQ[4]	X	X	X	DQ[5]	DQ[6]	DQ[7]	sDQ[8]

- GPIO = available for GPIO/PHYLite.
- X = not available for GPIO/PHYLite.
- DQ lane with an *s* prefix is the lane used for the following:
  - WUSER/RUSER in x72 configuration or
  - ECC Lane for x64 + ECC configuration
- x64 with ECC is not supported in the current version of the Quartus Prime software.

6.2.4.3. General Guidelines - DDR4

Observe the following general guidelines when placing pins for your Agilex 7 M-Series external memory interface.

Note:

- EMIF IP pin-out requirements for the Agilex 7 M-Series Hard Processor Subsystem (HPS) are more restrictive than for a non-HPS memory interface. The HPS EMIF IP defines a fixed pin-out in the Quartus Prime Pro Edition IP file (`.qip`), based on the IP configuration.

1. Ensure that the pins of a single external memory interface reside on the same edge I/O.
2. The address and command pins and their associated clock pins in the address and command bank must follow a fixed pin-out scheme, as defined in the table in the *Address and Command Pin Placement for DDR4* topic.
3. Not every byte lane can function as an address and command lane or a data lane. The pin assignment must adhere to the DDR4 data width mapping defined in *DDR4 Data Width Mapping*.
4. A byte lane must not be used by both address and command pins and data pins.
5. An I/O 96 bank cannot be used for more than one interface – meaning that two sub-banks belonging to two different EMIF interfaces are not permitted.
6. Sharing of byte lanes within a sub-bank for two different interfaces is not permitted; you can assign byte lanes within a sub-bank to one EMIF interface only.
7. Any pin in the same bank that is not used by an external memory interface may not be available for use as a general purpose I/O pin.
   - For fabric EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same sub-bank, pins in an I/O lane that is not assigned to an EMIF interface, can be used as general purpose I/O pins.
   - For HPS EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same bank, pins in an I/O lane that is not assigned to an EMIF interface cannot be used as general purpose I/O pins either.
   - When the network-on-a-chip (NoC) is used, the use of initiator may block the byte lane in the I/O bank adjacent to the sector where the initiator is located:
     — initiator 0 blocks the access for byte lane (or I/O lane) BL4, BL5, BL6, and partial BL7 (pin index 48 to 88 in the IO96 bank).
     — initiator 1 blocks the access to the fabric AXI command/control port for the primary/secondary controller in the same I/O bank, making the hard controller unavailable for EMIF purposes.
     — initiator 2 blocks the access for BL0,1,2,3 (P0 to P47).
8. All address and command pins and their associated clock pins (CK and CK#) must reside within a single sub-bank. Refer to the table in *DDR4 Data Width Mapping* for the supported address and command and data lane placements for DDR4.
9. An external memory interface can occupy one or more banks on the same edge. When an interface must occupy multiple banks, ensure the following:

- That the banks are adjacent to one another.
- That you used only the supported data width mapping as defined in the *DDR4 Data Width Mapping* table in the *DDR4 Data Width Mapping* topic. Be aware that not every byte lane can be used as an address and command lane or a data lane.
- That for lockstep configuration, you used only supported configuration as defined in the *Supported DDR4 Mapping for Lockstep Configuration for x64* and *Supported Lockstep configuration for DDR4 x72 or x64 (with ECC)* tables in the *DDR4 Data Width Mapping* topic.

10. Place read data groups according to the DQS grouping in the pin table and the Pin Planner. Read data strobes (such as DQS and DQS#) must reside at physical pins capable of functioning as DQS/CQ and DQSn/CQn for a specific read data group size. You must place the associated read data pins (such as DQ and Q), within the same group.

    Note: For DDR4 interfaces with x4 components, you can use the strobe pins with either of the upper or lower DQ nibbles that are placed within a x8 DQS group in an I/O lane. You must place the DQ pins and associated strobes entirely in either the upper or lower half of a 12-bit bank sub-group.

    Consult the pin table for your device to identify the association between DQ pins and DQS pins for x4 mode operation. Additional restrictions apply for x4/x8 DIMM interoperability.

11. One of the sub-banks in the device (typically the left sub-bank within corner Bank 3A) may not be available if you use certain device configuration schemes. For some schemes, there may be one byte lane available for EMIF data group:
    - AVST-8 – This is contained entirely within the SDM, therefore all lanes of sub-bank 3A can be used by the external memory interface.
    - AVST-32 – Byte lanes 4, 5, 6, and 7 are all effectively occupied and are not usable by the external memory interface.
    - AVST-16 – Byte lanes 6 contains SDM_DATA[25:16], and is not used by AVSTx16. However, the external memory interface cannot use byte lane 6 when byte lanes 4 and 5 are not usable for EMIF purposes.

6.2.4.4. x4 DIMM Implementation

DIMMS using a x4 DQS configuration require remapping of the DQS signals to achieve compatibility between the EMIF IP and the JEDEC standard DIMM socket connections.

The necessary remapping is shown in the table below. You can implement this DQS remapping in either RTL logic or in your schematic wiring connections.

Table 86. Mapping of DQS Signals Between DIMM and the EMIF IP

DIMM			Quartus Prime EMIF IP	
DQS0_A	DQ[3:0]_A		DQS0	DQ[3:0]_A
DQS5_A	DQ[7:4]_A		DQS1	DQ[7:4]_A
DQS1_A	DQ[11:8]_A		DQS2	DQ[11:8]_A
DQS6_A	DQ[15:12]_A		DQS3	DQ[15:12]_A
DQS2_A	DQ[19:16]_A		DQS4	DQ[19:16]_A

continued...

DIMM			Quartus Prime EMIF IP	
DQS7_A	DQ[23:20]_A		DQS5	DQ[23:20]_A
DQS3_A	DQ[27:24]_A		DQS6	DQ[27:24]_A
DQS8_A	DQ[31:28]_A		DQS7	DQ[31:28]_A
DQS4_A	CB[3:0]_A		DQS8	CB[3:0]_A
DQS9_A	CB[7:4]_A		DQS9	CB[7:4]_A
DQS0_B	DQ[3:0]_B		DQS10	DQ[3:0]_B
DQS5_B	DQ[7:4]_B		DQS11	DQ[7:4]_B
DQS1_B	DQ[11:8]_B		DQS12	DQ[11:8]_B
DQS6_B	DQ[15:12]_B		DQS13	DQ[15:12]_B
DQS2_B	DQ[19:16]_B		DQS14	DQ[19:16]_B
DQS7_B	DQ[23:20]_B		DQS15	DQ[23:20]_B
DQS3_B	DQ[27:24]_B		DQS16	DQ[27:24]_B
DQS8_B	DQ[31:28]_B		DQS17	DQ[31:28]_B
DQS4_B	CB[3:0]_B		DQS18	CB[3:0]_B
DQS9_B	CB[7:4]_B		DQS19	CB[7:4]_B

Data Bus Connection Mapping Flow

1. Connect all FPGA DQ pins accordingly to DIMM DQ pins. No remapping is required.
2. DQS/DQS*n* remapping is required either on the board schematics or in the RTL code.

When designing a board to support x4 DQS groups, Intel recommends that you make it compatible for x8 mode, for the following reasons:

- Provides the flexibility of x4 and x8 DIMM support.
- Allows use of x8 DQS group connectivity rules.
- Allows use of x8 timing rules for matching. Adhere to x4/x8 interoperability rules when designing a DIMM interface, even if the primary use case is to support x4 DIMMs only, because doing so facilitates debug and future migration capabilities. Regardless, the rules for length matching for two nibbles in a x4 interface must match those of the signals for a corresponding x8 interface, as the data terminations are turned on and off at the same time for both x4 DQS groups in an I/O lane. If the two x4 DQS groups were to have significantly different trace delays, it could adversely affect signal integrity. Trace delays for two nibbles packed within the IO12 lanes are matched using the same guidelines as a single x8 byte lane.

6.2.4.5. Specific Pin Connection Requirements

PLL

You must constrain the PLL reference clock to the address and command sub-bank only.

- You must constrain differential reference clocks to pin indices 0 and 1 in lane AC2.
- The sharing of PLL reference clocks across multiple interfaces is permitted; however, pin indices 0 and 1 of lane 2 of the address and command sub-bank for all slave EMIF interfaces can be used only for supplying reference clocks. Intel recommends that you consider connecting these clock input pins to a reference clock source to facilitate greater system implementation flexibility.

Note: Agilex 7 M-Series FPGAs do not support single-ended I/O PLL reference clocks for EMIF IP.

OCT

For DDR4, you must constrain the RZQ pin to pin index 2 in lane AC2.

- Every EMIF instance requires its own dedicated RZQ pin.
- The sharing of RZQ pins is not permitted.

Address and Command

For DDR4, you must constrain the ALERT_N pin to the address and command lanes only.

- In three-lane address and command schemes, you can place the ALERT_N pin at pin index 8 in lane AC2 only.
- In four-lane address and command schemes, you can place the ALERT_N pin at pin index 8 in lane AC2 or at pin index 8 in lane AC3. When you generate the IP, the resulting RTL specifies which connection to use.

DQS/DQ/DM

For DDR4 x8 DQS grouping, the following rules apply:

- You may use pin indices 0, 1, 2, 3, 8, 9, 10, and 11 within a lane for DQ mode pins only.
- You must use pin index 4 for the DQS_t pin only.
- You must use pin index 5 for the DQS_c pin only.
- You must ensure that pin index 7 remains unused. Pin index 7 is not available for use as a general purpose I/O.
- You must use pin index 6 for the DM/DBI_N pin only.

For DDR4 x4 DQS grouping, the following rules apply:

- You may use pin indices 0, 1, 2, and 3 within a lane for DQ mode pins for the lower nibble only. Pin rotation within this group is permitted.
- You must use pin index 4 for the DQS_t pin only of the lower nibble.
- You must use pin index 5 for the DQS_c pin only of the lower nibble.
- You may use pin indices 8, 9, 10, and 11 within a lane for the DQ mode pins only for the upper nibble.

- Pin rotation within this group is permitted.
- You must use pin index 6 for the DQS_t pin only of the upper nibble.
- You must use pin index 7 for the DQS_c pin only of the upper nibble.

6.2.4.6. Command and Address Signals

Command and address signals in SDRAM devices are clocked into the memory device using the CK or CK# signal. These pins operate at single data rate (SDR) using only one clock edge. The number of address pins depends on the SDRAM device capacity. The address pins are multiplexed, so two clock cycles are required to send the row, column, and bank address.

Although DDR4 operates in fundamentally the same way as other SDRAM, there are no dedicated pins for RAS#, CAS#, and WE#, as those are shared with higher-order address pins. DDR4 has CS#, CKE, ODT, and RESET# pins, similar to DDR3. DDR4 also has some additional pins, including the ACT# (activate) pin and BG (bank group) pins.

6.2.4.7. Clock Signals

DDR4 SDRAM devices use CK and CK# signals to clock the address and command signals into the memory.

The memory uses these clock signals to generate the DQS signal during a read through the DLL inside the memory. The SDRAM data sheet specifies the following timings:

- t_{DQSCK} is the skew between the CK or CK# signals and the SDRAM-generated DQS signal.
- t_{DSH} is the DQS falling edge from CK rising edge hold time.
- t_{DSS} is the DQS falling edge from CK rising edge setup time.
- t_{DQSS} is the positive DQS latching edge to CK rising edge.

SDRAM devices have a write requirement (t_{DQSS}) that states the positive edge of the DQS signal on writes must be within ± 25% (± 90°) of the positive edge of the SDRAM clock input. Therefore, you should generate the CK and CK# signals using the DDR registers in the IOE to match with the DQS signal and reduce any variations across process, voltage, and temperature. The positive edge of the SDRAM clock, CK, is aligned with the DQS write to satisfy t_{DQSS}.

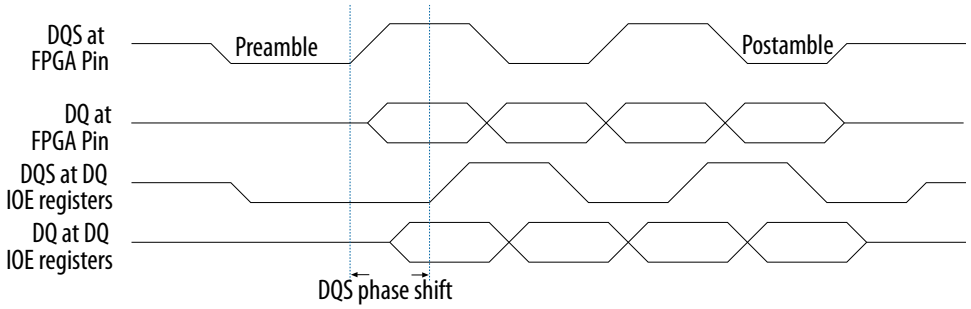
6.2.4.8. Data, Data Strobes, DM/DBI, and Optional ECC Signals

DDR4 SDRAM devices use bidirectional differential data strobes. Differential DQS operation enables improved system timing due to reduced crosstalk and less simultaneous switching noise on the strobe output drivers. The DQ pins are also bidirectional.

DQ pins in DDR4 SDRAM interfaces can operate in either ×4 or ×8 mode DQS groups, depending on your chosen memory device or DIMM, regardless of interface width. The ×4 and ×8 configurations use one pair of bidirectional data strobe signals, DQS and DQSn, to capture input data. However, two pairs of data strobes, UDQS and UDQS# (upper byte) and LDQS and LDQS# (lower byte), are required by ×16 configurations. A group of DQ pins must remain associated with its respective DQS and DQSn pins.

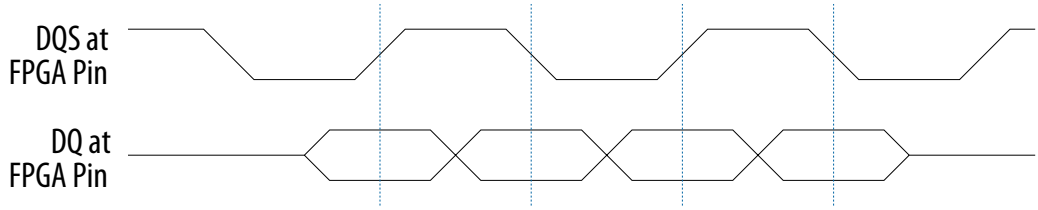
The DQ signals are edge-aligned with the DQS signal during a read from the memory and are center-aligned with the DQS signal during a write to the memory. The memory controller shifts the DQ signals by –90 degrees during a write operation to center align the DQ and DQS signals. The PHY IP delays the DQS signal during a read, so that the DQ and DQS signals are center aligned at the capture register. Intel devices use a phase-locked loop (PLL) to center-align the DQS signal with respect to the DQ signals during writes and use dedicated DQS phase-shift circuitry to shift the incoming DQS signal during reads. The following figure shows an example where the DQS signal is shifted by 90 degrees for a read from the SDRAM.

Figure 13. Edge-aligned DQ and DQS Relationship During a SDRAM Read in Burst-of-Four Mode



The following figure shows an example of the relationship between the data and data strobe during a burst-of-four write.

Figure 14. DQ and DQS Relationship During a SDRAM Write in Burst-of-Four Mode



The memory device's setup (t_{DS}) and hold times (t_{DH}) for the DQ and DM pins during writes are relative to the edges of DQS write signals and not the CK or CK# clock. Setup and hold requirements are not necessarily balanced.

The DQS signal is generated on the positive edge of the system clock to meet the t_{DQSS} requirement. DQ and DM signals use a clock shifted –90 degrees from the system clock, so that the DQS edges are centered on the DQ or DM signals when they arrive at the SDRAM. The DQS, DQ, and DM board trace lengths need to be tightly matched (within 20 ps).

The SDRAM uses the DM pins during a write operation. Driving the DM pins low shows that the write is valid. The memory masks the DQ signals if the DM pins are driven high. To generate the DM signal, Intel recommends that you use the spare DQ pin within the same DQS group as the respective data, to minimize skew.

The DM signal's timing requirements at the SDRAM input are identical to those for DQ data. The DDR registers, clocked by the –90 degree shifted clock, create the DM signals.

DDR4 supports DM similarly to other SDRAM, except that in DDR4 DM is active LOW and bidirectional, because it supports Data Bus Inversion (DBI) through the same pin. DM is multiplexed with DBI by a Mode Register setting whereby only one function can be enabled at a time. DBI is an input/output identifying whether to store/output the true or inverted data. When enabled, if DBI is LOW, during a write operation the data is inverted and stored inside the DDR4 SDRAM; during a read operation, the data is inverted and output. The data is not inverted if DBI is HIGH. For Agilex 7 interfaces, the DM/DBI pins do not need to be paired with a DQ pin.

Some SDRAM modules support error correction coding (ECC) to allow the controller to detect and automatically correct error in data transmission. The 72-bit SDRAM modules contain eight extra data pins in addition to 64 data pins. The eight extra ECC pins should be connected to a single DQS or DQ group on the FPGA.

6.3. DDR4 Board Design Guidelines

The following topics provide guidelines for improving the signal integrity of your system and for successfully implementing a DDR4 SDRAM interface on your system.

The following areas are discussed:

- comparison of various types of termination schemes, and their effects on the signal quality on the receiver
- proper drive strength setting on the FPGA to optimize the signal integrity at the receiver
- effects of different loading types, such as components versus DIMM configuration, on signal quality

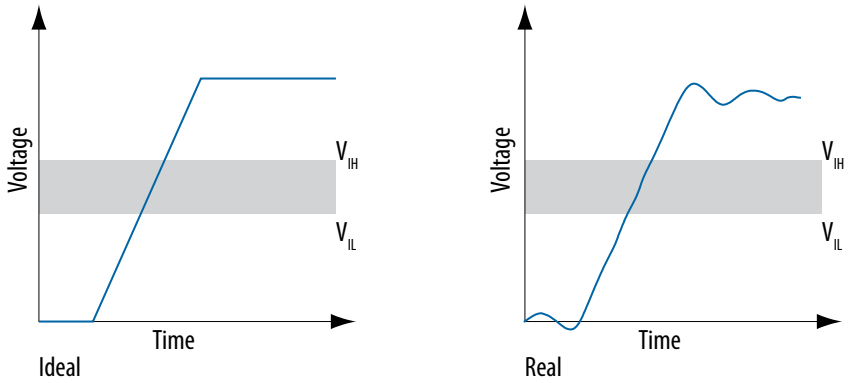
It is important to understand the trade-offs between different types of termination schemes, the effects of output drive strengths, and different loading types, so that you can swiftly navigate through the multiple combinations and choose the best possible settings for your designs.

The following key factors affect signal quality at the receiver:

- Leveling and dynamic ODT
- Proper use of termination
- Layout guidelines

As memory interface performance increases, board designers must pay closer attention to the quality of the signal seen at the receiver because poorly transmitted signals can dramatically reduce the overall data-valid margin at the receiver. The following figure shows the differences between an ideal and real signal seen by the receiver.

Figure 15. Ideal and Real Signal at the Receiver



6.3.1. Terminations for DDR4 with Agilex 7 M-Series Devices

The following topics describe considerations specific to DDR4 external memory interface protocols on Agilex 7 M-Series devices.

6.3.1.1. Dynamic On-Chip Termination (OCT)

Depending upon the Rs (series) and Rt (parallel) OCT values that you want, you should choose appropriate values for the RZQ resistor and connect this resistor to the RZQ pin of the FPGA.

Refer to the *External Memory Interfaces Agilex 7 M-Series FPGA IP* parameter editor to determine the supported termination values.

6.3.1.2. Dynamic On-Die Termination (ODT) in DDR4

In DDR4, in addition to the Rtt_nom and Rtt_wr values, which are applied during read and write respectively, a third option called Rtt_park is available. When Rtt_park is enabled, a selected termination value is set in the DRAM when ODT is driven low.

Refer to the DDR4 JEDEC* specification or your memory vendor data sheet for details about available termination values and functional description for dynamic ODT in DDR4 devices.

6.3.1.3. Choosing Terminations on Agilex 7 M-Series FPGA Devices

To determine optimal on-chip termination (OCT) and on-die termination (ODT) values for best signal integrity, you should simulate your memory interface in HyperLynx or a similar tool, using a simulation model extracted from the PCB of your memory interface channel.

If the optimal OCT and ODT values as determined by simulation are not available in the list of available values in the parameter editor, select the closest available termination values for OCT and ODT.

For information about available ODT choices, refer to your memory vendor data sheet.

6.3.1.4. On-Chip Termination Recommendations for Agilex 7 M-Series FPGA Devices

In the EMIF IP parameter editor you can select values from drop-down lists for each of the following:

- output mode drive strength for the address/command bus.
- output mode drive strength for the memory clock.
- output mode drive strength for the data bus.
- input mode termination strength for the data bus.

The range of available values may vary, depending on your memory protocol and silicon revision.

You can use the default values as starting points; however, for best results, you should sweep the entire range of legal values and generate multiple hardware designs to determine the optimal settings for your board and memory device.

Once you have found the optimal settings for your design, uncheck the **Use Default I/O settings** checkbox and use your optimal settings for all future compilations, even if those settings align with the default settings. This ensures that your settings are preserved if the IP is upgraded to a future version.

6.3.2. General Layout Routing Guidelines

Follow the guidelines in this section for routing from the FPGA to memory for Agilex 7 M-Series devices.

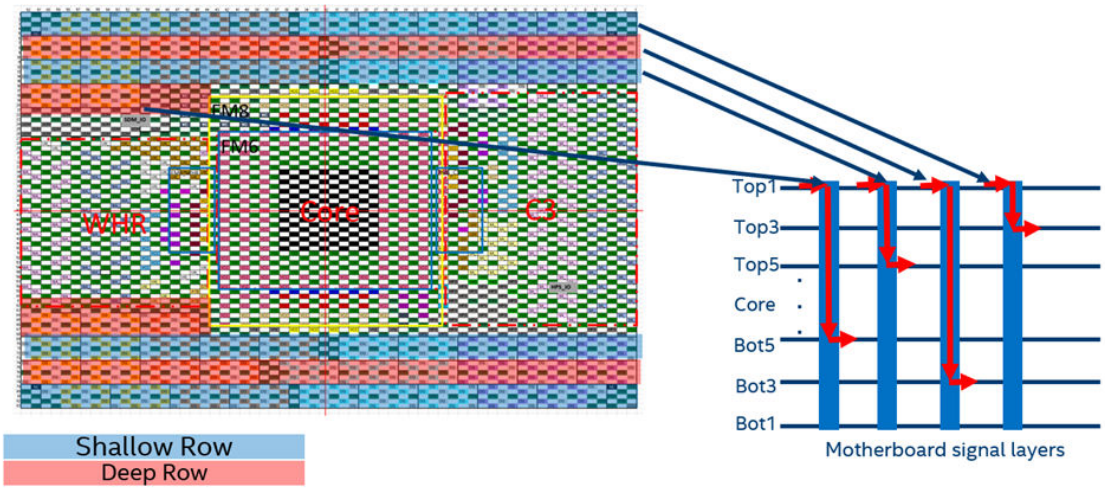
For maximum channel margin, you should consider the following general routing optimizations during the layout design phase:

- When routing the memory interface, ensure that there are solid ground reference planes without any plane splits or voids, to ensure an uninterrupted current return path.
- For signal vias in layer transitions, you must place ground stitching vias close by, within 80 mil in distance (closer is better), and in between signal vias, to minimize crosstalk among signal vias. Avoid any unnecessary signal layer transitions to minimize crosstalk, loss, and skews.
- Trace impedance plays an important role in signal integrity; board designers must follow impedance recommendations for each signal group and configuration according to the guidelines in this document. If you use a different stackup than the reference stackup in the PCB design, you must tune the trace width and geometries to achieve the impedance recommendations.
- Intel recommends using 45-degree angles (not sharp 90-degree corners) when routing signal turns. Use $3\times h$ spacing for serpentine routing, where h is the height or distance from the trace to the nearest GND reference plane.
- Avoid referencing a signal to both power and ground planes at the same time (dual referencing), for signal return paths. When this cannot be avoided, ensure that the closer reference plane is solid ground, and the far side power plane is not noisy.
- Avoid routing two internal signal layers adjacent to each other (dual stripline routing). When this cannot be avoided, use angled routing between two signal layers to minimize crosstalk and coupling between the layers.

- Follow time-domain length and skew matching rules to ensure that your interface meets timing requirements. You should route signals from the same byte or group together on the same layer to avoid any out-of-phase crosstalk caused by varying layer transition lengths.
- To optimize memory interface margins, Intel recommends the following routing strategies:
  - For DIMM configurations, route DQ and DQS signals on shallow layers with short via transition lengths, because they have tighter timing margins than address, command, and control signals. (Shallow layers are those above the PCB core where via transition lengths are short.)
  - For discrete device configurations, route address, command, and control signals on shallow layers.

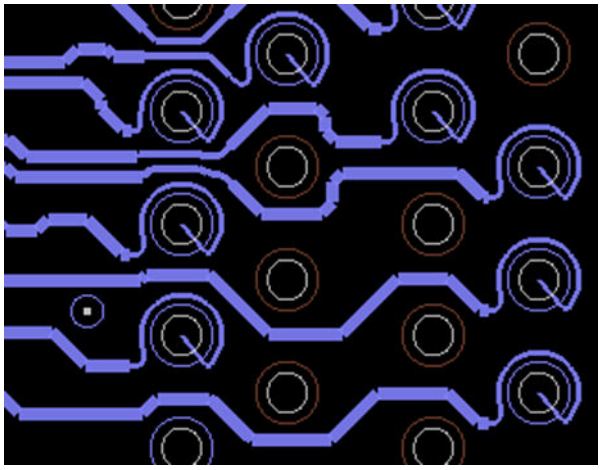
- For boards thicker than 65 mil, Intel recommends alternating adjacent FPGA EMIF BGA/ball rows with deep and shallow board via transitions to minimize crosstalk between adjacent bytes. This method is illustrated in the following figure:

Figure 16. Recommended alternate adjacent via transitions to avoid crosstalk between adjacent bytes



- For boards thicker than 65 mil, using the pin-through-hole (PTH) type of DIMM connector, Intel recommends implementing a loop-routing-around-DIMM-pin structure (Lcomp) to improve impedance matching between signal routing and the DIMM connector. Refer to the following figure.

Figure 17. Recommended Lcomp structure for better impedance matching



- For PCB designs using a surface mount technology (SMT) type of DIMM connector, Intel recommends placing a cutout (void) in the ground reference plane underneath the connector pads for DDR4 signals to minimize connector pad capacitance. Refer to the following figure for the recommended cutout on ground reference plane underneath the connector pad on surface layer.

Figure 18. Recommended Cutout on Ground Reference Plane

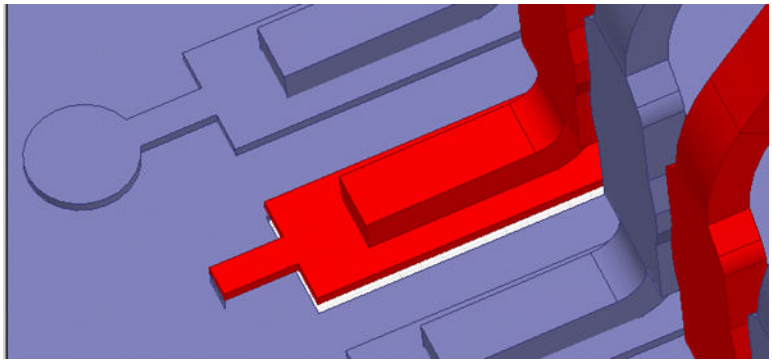


Figure 19. Closeup View of Connections

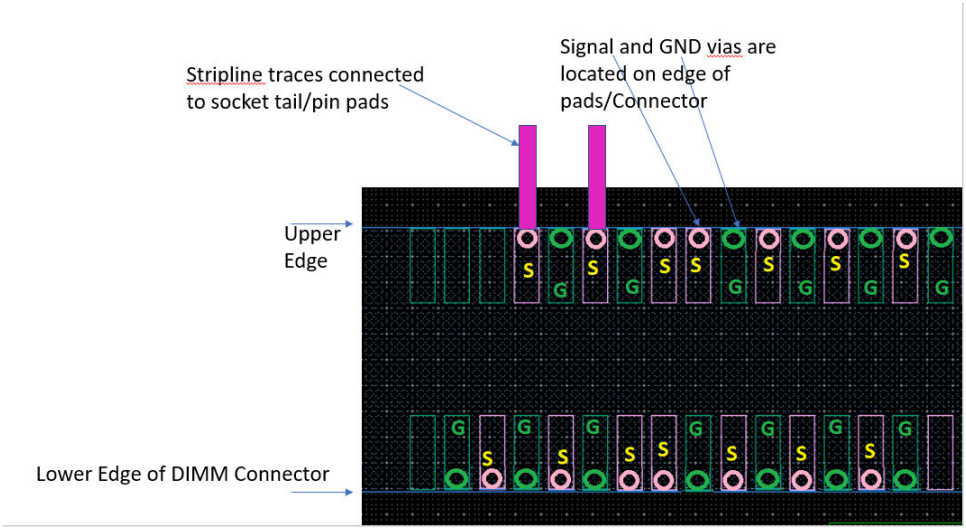
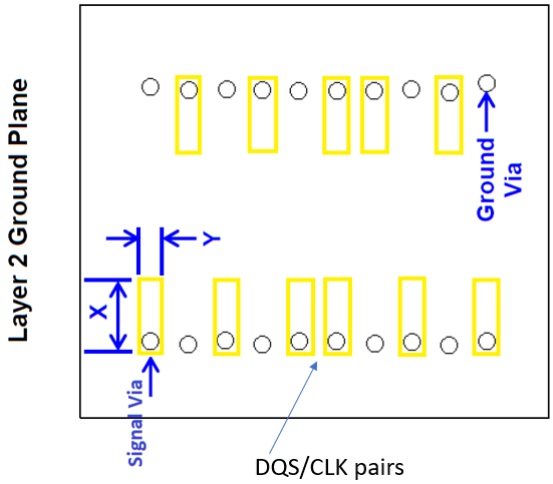


Figure 20. Closeup View of Connections



6.3.3. Reference Stackup

This topic illustrates the reference stackup on which EMIF routing design guidelines are based.

It is important to understand that trace geometry such as width, thickness, and edge-to-edge spacing, and the distance to reference planes, all impact trace impedance and crosstalk levels.

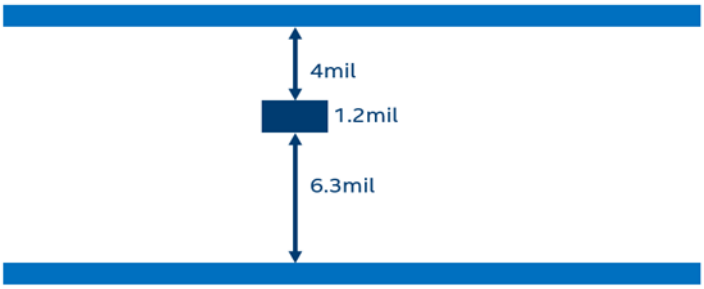
Table 87. Reference stackup details

Layer	Type	Thickness
SM TOP		0.5
L1	signal	1.8
D1	prepreg	2.7
L2	gnd/power	1.2
D2	core	4.0
L3	signal	1.2
D3	prepreg	6.3
L4	gnd/power	1.2
D4	core	4.0
L5	signal	1.2
D5	prepreg	6.3
L6	gnd/power	1.2
D6	core	4.0
L7	signal	1.2
D7	prepreg	6.3
L8	gnd	1.2
D8	core	4
	Power	1.2
	prepreg	6.3
	power	1.2
	core	4
	gnd	1.2
	prepreg	6.3
	power	1.2
	core	4
L9	gnd	1.2
D9	prepreg	6.3
L10	signal	1.2

continued...

Layer	Type	Thickness
D10	core	4.0
L11	gnd/power	1.2
D11	prepreg	6.3
L12	signal	1.2
D12	core	4.0
L13	gnd/power	1.2
D13	prepreg	6.3
L14	signal	1.2
D14	core	4.0
L15	gnd/power	1.2
D15	prepreg	2.7
L16	signal	1.8
SM BOT		0.5
	Total	120.1

Figure 21. Reference trace geometries



The reference stackup height is selected to be 120 mil to cover maximum signal via coupling (110mil) in simulation while extracting EMIF design guideline. Intel recommends that board designers do not exceed 110mil signal via coupling (stripline routing on inner layers) in the EMIF layout PCB design for DDR4 interfaces.

If the PCB stackup exceeds 120 mil in height, Intel recommends routing EMIF signals on upper layers, not to exceed more than 110 mil of signal via coupling.

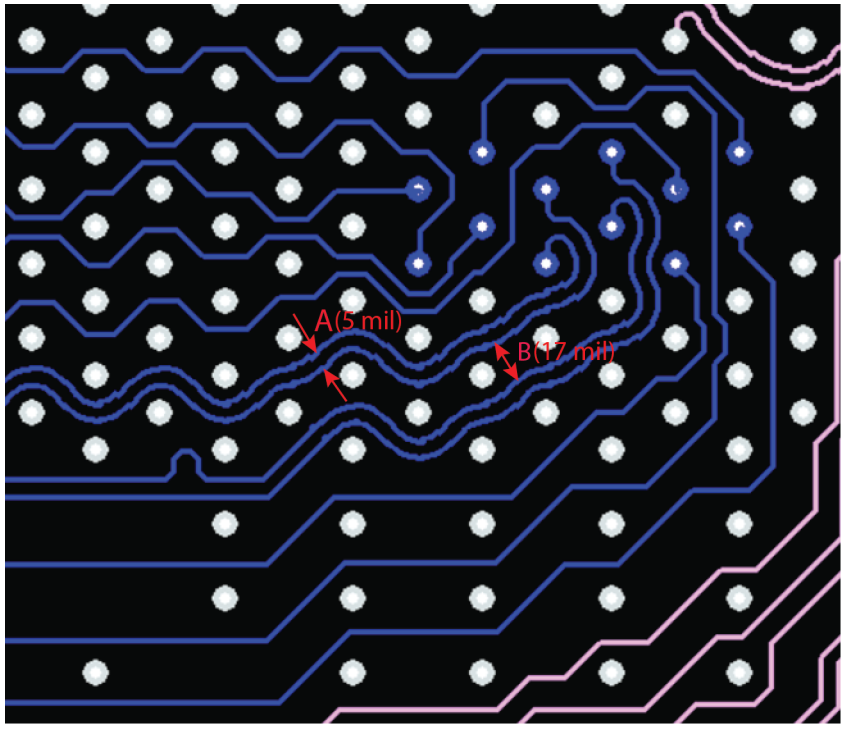
The reference stackup materials in the above figure are selected as FR4, to represent worst-case signal loss in design phase simulation. In case of low-loss materials, the maximum end-to-end routing length shall be larger than the recommended end-to-end routing length in the design guidelines; however, you must perform time-domain channel simulation to ensure that timing requirements are met.

6.3.4. Agilex 7 M-Series EMIF-Specific Routing Guidelines for Various DDR4 Topologies

This section discusses EMIF-related layout guidelines for Agilex 7 M-Series devices.

The Agilex 7 M-Series family pin floorplan is a HEX pattern with 1mm pitch. The following figure shows an example of DDR routing for an IO12 (one-byte data) on PCB within FPGA fan-out region.

Figure 22. Agilex 7 1mm HEX pin pattern/floorplan and recommended routing for one byte of data (IO12)



The following general notes apply to the EMIF routing guidelines tables in subsequent topics:

- All spacing requirements are the minimum requirement to be met on PCB in EMIF routing guideline table.
- Breakout (BO1/BO2) spacings have two different values in guideline tables. The first value represents minimum spacing between two signals routed as a pair (tightly coupled signals); this value is marked as A (5 mil) in the above figure. The second value represents minimum spacing between two pairs, and is marked as B (17 mil) in the above figure.
- Main route (M) spacings have both value in mil and formula. In formula, *h* represents the trace-to-nearest-reference-plane height or distance. In cases using a stackup different than the reference stackup, board designers shall use formula to calculate the correct spacing requirements.
- There is no differential impedance target for CLK nor DQS. Board designers shall follow single-ended impedance target and keep the signals within the pair closely coupled, within 3-4 mil spacing. For information on DQS/DQSB and CLK/CLKB, refer to the *Skew Matching Guidelines for DDR4 DIMM Topologies* and *Skew Matching Guidelines for DDR4 Discrete Topologies* tables, for DIMM and discrete device implementations, respectively.

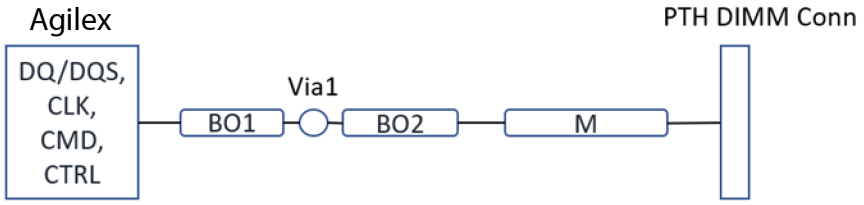
- In guideline tables, *SL* stands for stripline routing recommendation and *US* stands for upper surface (Microstrip) routing recommendation.
- The trace width value/geometry in guideline tables stands for trace designed for target impedance based on the reference stackup. This trace geometry shall be designed based on actual stackup and target impedance in guideline table.
- In guideline tables, *BO1* and *BO2* represent fan-out routing lengths. *M* stands for out of fan-out (PCB main) routing lengths

6.3.4.1. One DIMM per Channel (1DPC) for UDIMM, RDIMM, and SODIMM DDR4 Topologies

The interface covers data bytes (DQ/DQS), address signals, command signals (BA, BG, RAS, CAS, WE, ACT, PAR), control signals (CKE, CS, ODT) and clocks (CLK).

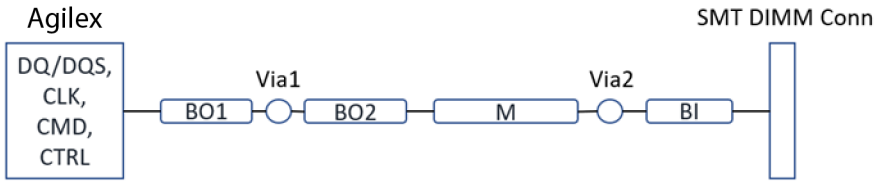
The following figure illustrates the signal connection topology for a PTH type of connector for UDIMM and RDIMM topologies.

Figure 23. Signal connections for DDR4 1DPC DIMM configuration using PTH DIMM connector



The following figure illustrates the signal connection topology for an SMT type of connector for UDIMM, RDIMM, and SODIMM topologies.

Figure 24. Signal connections for DDR4 1DPC DIMM configuration using SMT DIMM connector



The following table provides specific routing guidelines for one DIMM per channel in UDIMM, RDIMM, and SODIMM topologies for all supported signals in the interface.

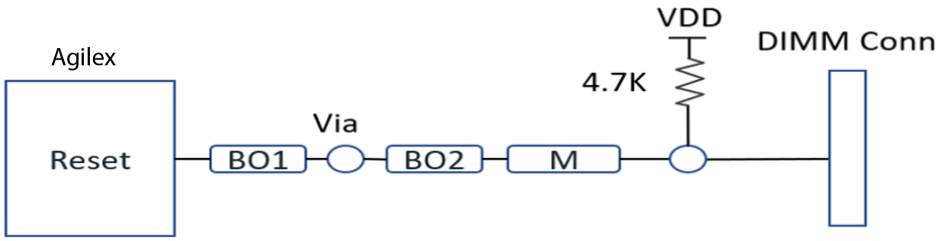
Table 88. Specific DDR4 1DPC routing guidelines for UDIMM, RDIMM, and SODIMM configurations

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohms)	Target Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil):DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/CTRL/CKE	Channel to Channel Spacing (DQ to DQ, between two channels)
			Segment	Total MB									
CLK	BO1	US	50	4500		4	5, 17	5, 17		4		17	
	BO2	SL	1000			4	5, 17	5, 17		4		17	
	M	SL			45	4.5		12 (3h)		4		12 (3)	
	BI	US	50			4		12 (3h)		4		12 (3h)	
CMD, CTRL, ALERT	BO1	US	50	4500		4	5, 17	5, 17					
	BO2	SL	1000			4	5, 17	5, 17					
	M	SL			45	4.5	8 (2h)	12 (3h)					
	BI	US	100			4	8 (2h)	12 (3h)					
DQ	BO1	US	50	4500		3	5, 17		17				17
	BO2	SL	1000			3	5, 17		17				17
	M	SL			50	3.5	8 (2h)		12 (3h)				16 (4h)
	BI	US	50			3.5	8 (2h)		12 (3h)				16 (4h)
DQS	BO1	US	50	4500		3	5, 17			4	17		
	BO2	SL	1000			3	5, 17			4	17		
	M	SL			50	3.5				4	12 (3h)		
	BI	US	50			3.5							

For an explanation of the guidelines represented in this table, refer to the bullet points immediately following Agilex 7 M-Series EMIF-Specific Routing Guidelines for Various DDR4 Topologies on page 110.

The following figure shows the RESET signal scheme and routing guideline for one DIMM per channel topologies.

Figure 25. Reset scheme for 1DPC DIMM topologies



The target impedance for the RESET signal is 50 ohms. The RESET signal shall have at least 3×h (where *h* stands for trace to nearest reference plane height or distance) spacing to other nearby signals on the same layer. The end-to-end RESET trace length is not limited but shall not exceed 5 inches.

6.3.4.2. Skew Matching Guidelines for DIMM Configurations

The guidelines in this topic apply to any DIMM topology, regardless of DIMM type or number of ranks.

Board designers must observe the following guidelines for DDR4 DIMM skew matching:

- Perform skew matching in time (picoseconds) rather than in actual trace length, to better account for via delays when signals are routed on different layers.
- Include both package per-pin skew and PCB delay when performing skew matching.
- Skew (length) matching for the alert signal is not required.

The following table provides skew matching guidelines for DDR4 DIMM topologies.

Table 89. Skew Matching Guidelines for DDR4 DIMM Topologies

DIMM Skew Matching Rule	Length in Time (ps)
Length matching between DQS and CLK	-255ps < CLK - DQS < 425ps
Length matching between DQ and DQS within byte	-3.5ps < DQ - DQS < 3.5ps
Length matching between DQS and DQS#	< 1ps
Length matching between CLK and CLK#	< 1ps
Length matching between CLK0 and CLK1	< 8ps
Length matching between CMD/ADDR/CTRL and CLK	-20ps < CLK - CMD/ADDR/CTRL < 20ps
Length matching among CMD/ADDR/CTRL within each channel	< 20ps
Include package length in skew matching for FPGA device with no migration	Required
Include package length in skew matching for FPGA device with migration when all package net length are available	It is recommended to use the final migrated package net length
Include package length in skew matching for FPGA device with migration when all package net length are not available	Not recommended

6.3.4.3. Power Delivery Recommendations for the Memory / DIMM Side

This topic describes power distribution network (PDN) design guidelines for one DIMM per channel (1DPC) memory interfaces at the DIMM/memory side.

Note: For information on power distribution network design at the FPGA to meet timing margins, refer to the AG014 PDN design guideline.

In the following table, the number of decoupling capacitors is based on a single channel. If multiple channels are sharing the same power rail at the DIMM, the number of decoupling capacitors at the DIMM must be scaled accordingly.

Physically small decoupling capacitors are recommended to minimize area, inductance, and resistance on the PDN path on the printed circuit board.

Table 90. Required Decoupling Capacitors on the PCB for the Memory/DIMM Side

Memory Configuration	Power Domain	Decoupling Location	Quantity × Value (size)
DDR4 1DPC	VDDQ	4 near each side of DIMM connector close to VDDQ pins	8 × 47uF (0805)
		4 near each side of DIMM connector close to VDDQ pins	8 × 1uF (0402)
	VTT	Place capacitor on VTT plane close to DIMM	1 × 47uF (0805)
		Place capacitor on VTT plane close to DIMM	2 × 1uF (0402)
	VPP	Place capacitor close to DIMM	1 × 47uF (0805)
		Place capacitor close to DIMM	1 × 1uF (0402)
	VDDSPD	Place capacitor close to DIMM	1 × 0.1uF (0402)
		Place capacitor close to DIMM	1 × 2.2uF (0402)

6.3.5. DDR4 Routing Guidelines: Discrete (Component) Topologies

This section discusses two topologies for down-memory configurations: DDR4 single rank × 8 and DDR4 single rank × 16 for a 72 bit interface.

Intel strongly recommends that you perform simulations using extracted PCB models to ensure that component topologies remain robust under all PCB manufacturing tolerances. Also, carefully consider the number of components on the flyby chain, because every additional component on the flyby chain reduces timing margin on the address/command bus. Take care to provide a proper VTT termination voltage network with a reference voltage that feeds back to the V_{REFCA} input of every component on the flyby chain. Agilex 7 M-Series FPGA circuitry cannot compensate for discontinuities or trace length mismatches along the flyby chain, or for crosstalk between address/command or DQ signals.

6.3.5.1. Single Rank x 8 Discrete (Component) Topology

Nine memory devices are required to cover 72 bits of data in a single channel, with one rank and ×8 memory devices.

The interface covers data bytes (DQ/DQS), address signals, command signals (BA, BG, RAS, CAS, WE, ACT, PAR), control signals (CKE, CS, ODT) and clocks (CLK).

Figure 26. Signal connections for DDR4 Single Rank × 8 Discrete Topology (9 memory devices to cover 72 bits)

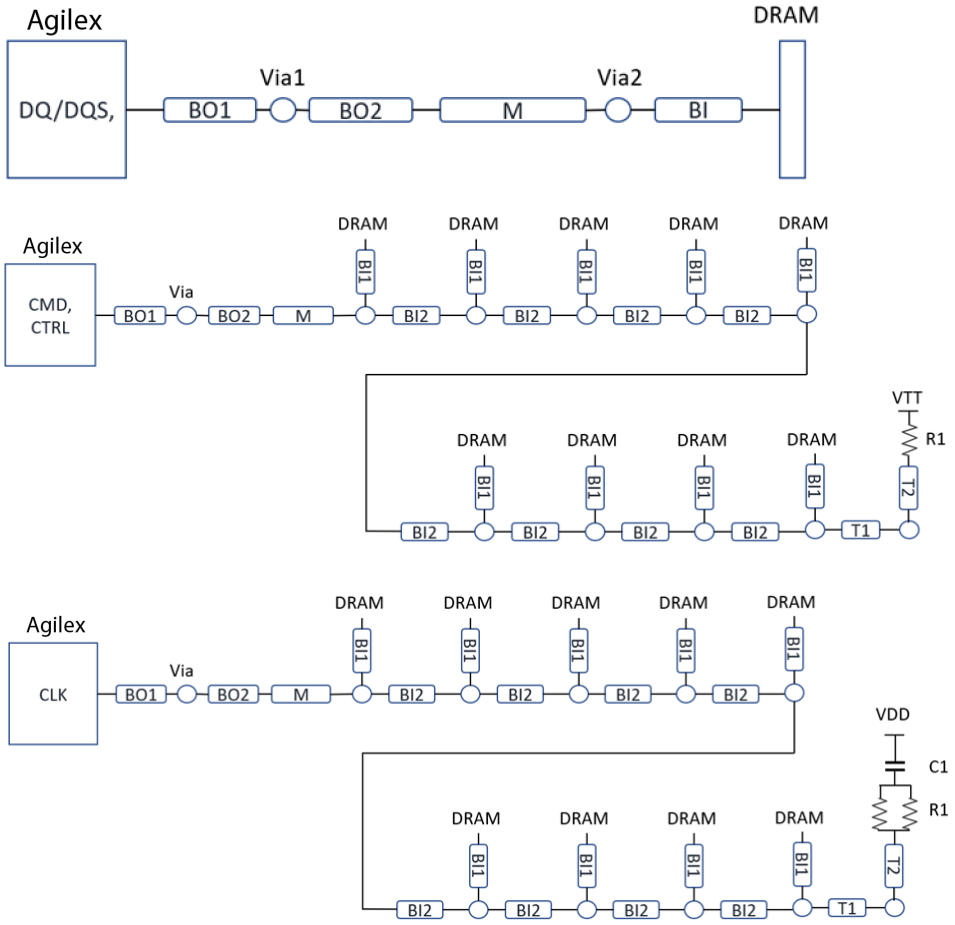


Table 91. Specific Routing Guidelines for Single Rank x8 Discrete Memory Topology for All Supported Signals in the Interface

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil): DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/CTRL/CKE	Rtt / Ctt
			Segment	Total MB									
CLK	BO1	US	50	To first DRAM: 4000 To last DRAM: 9600		4	5, 17	17		4		17	R1=36Ω C1=10nF
	BO2	SL	1000			4	5, 17	17		4		17	
	M	SL			40	5.5		12 (3h)		4		12 (3h)	
	BI1	US	50			3		12 (3h)		4		12 (3h)	
	BI2	SL	700		50	3		12 (3h)		4		12 (3h)	
	T1	SL	300			3		12 (3h)		4		12 (3h)	
	T2	US	50			3		12 (3h)		4		12 (3h)	
CMD, CTRL, Alert	BO1	US	50	To first DRAM: 4000 To last DRAM: 9600		4	5, 17	17					R1=36Ω alert_n requires an external pullup resistor to VDD (1.2V) of approximately 1KΩ.
	BO2	SL	1000			4	5, 17	17					
	M	SL			40	5.5	8 (2h)	12 (3h)					
	BI1	US	50			3	8 (2h)	12 (3h)					
	BI2	SL	700		50	3	8 (2h)	12 (3h)					
	T1	SL	300			3	8 (2h)	12 (3h)					
	T2	US	50			3	8 (2h)	12 (3h)					
DQ	BO1	US	50	5000		4	5, 17		17				
	BO2	SL	1000			4	5, 17		17				
	M	SL			45	4.5	8 (2h)		12 (3h)				
	BI	US	50			4	8 (2h)		12 (3h)				
DQS	BO1	US	50	5000		4				4	17		
	BO2	SL	1000			4				4	17		

continued...

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil): DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/CTRL/CKE	Rtt / Ctt
			Segment	Total MB									
	M	SL			45	4.5				4	12 (3h)		
	BI	US	50			4				4	12 (3h)		

For an explanation of the guidelines represented in this table, refer to the bullet points immediately following Figure 22 on page 111.

6.3.5.2. Single Rank x 16 Discrete (Component) Topology

Five memory devices are required to cover 72 bits of data in a single channel, with one rank and ×16 memory devices.

The interface covers data bytes (DQ/DQS), address signals, command signals (BA, BG, RAS, CAS, WE, ACT, PAR), control signals (CKE, CS, ODT) and clocks (CLK).

Figure 27. Signal connections for DDR4 Single Rank × 16 Discrete Topology (5 memory devices to cover 72 bits)

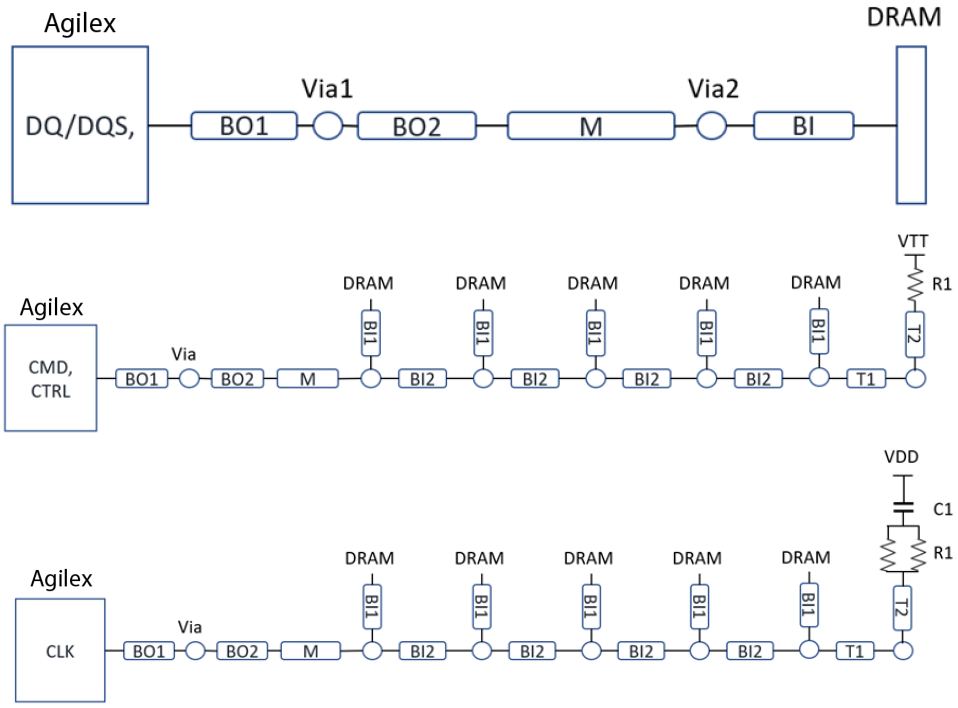


Table 92. Specific Routing Guidelines for Single Rank x16 Discrete Memory Topology for All Supported Signals in the Interface

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/ CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil): DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/ CTRL/CKE	Rtt / Ctt
			Segment	Total MB									
CLK	BO1	US	50	To first DRAM: 4000. To last DRAM: 6800.		4	5, 17	17		4		17	R1=36Ω. C1=10nF
	BO2	SL	1000			4	5, 17	17		4		17	
	M	SL			40	5.5		12 (3h)		4		12 (3h)	
	BI1	US	50			3		12 (3h)		4		12 (3h)	
	BI2	SL	700		50	3		12 (3h)		4		12 (3h)	
	T1	SL	300			3		12 (3h)		4		12 (3h)	
	T2	US	50			3		12 (3h)		4		12 (3h)	
CMD, CTRL, Alert	BO1	US	50	To first DRAM: 4000. To last DRAM: 6800.		4	5, 17	17					R1=36Ω alert_n requires an external pullup resistor to VDD (1.2V) of approximately 1KΩ.
	BO2	SL	1000			4	5, 17	17					
	M	SL			40	5.5	8 (2H)	12 (3h)					
	BI1	US	50			3	8 (2H)	12 (3h)					
	BI2	SL	700		50	3	8 (2H)	12 (3h)					
	T1	SL	300			3	8 (2H)	12 (3h)					
	T2	US	50			3	8 (2H)	12 (3h)					
DQ	BO1	US	50	5000		4	5, 17		17				
	BO2	SL	1000			4	5, 17		17				
	M	SL			45	4.5	8 (2H)		12 (3h)				
	BI	US	50			4	8 (2H)		12 (3h)				
DQS	BO1	US	50	5000		4				4	17		
	BO2	SL	1000			4				4	17		

continued...

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil): DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/CTRL/CKE	Rtt / Ctt
			Segment	Total MB									
	M	SL			45	4.5				4	12 (3h)		
	BI	US	50			4				4	12 (3h)		

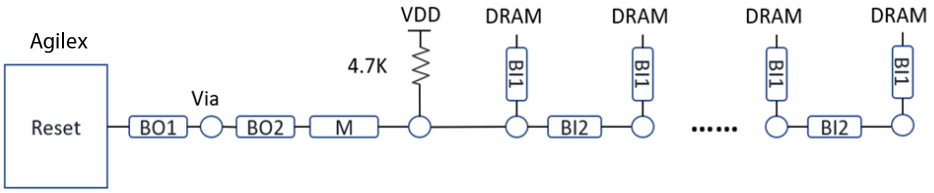
For an explanation of the guidelines represented in this table, refer to the bullet points immediately following Figure 22 on page 111.

6.3.5.3. ADDR/CMD Reference Voltage/RESET Signal Routing Guidelines for Single Rank x 8 and Single Rank x 16 Discrete (Component) Topologies

The target impedance for the RESET signal is 50 ohms. The RESET signal must have at least 3 × h (where *h* is the distance from the trace to the nearest reference plane) spacing to other nearby signals on the same layer. The end-to-end RESET trace length is not limited but must not exceed 5 inches to the first DRAM.

The following figure shows the RESET routing scheme, which you can apply to both single rank x 8 and single rank x16 topologies.

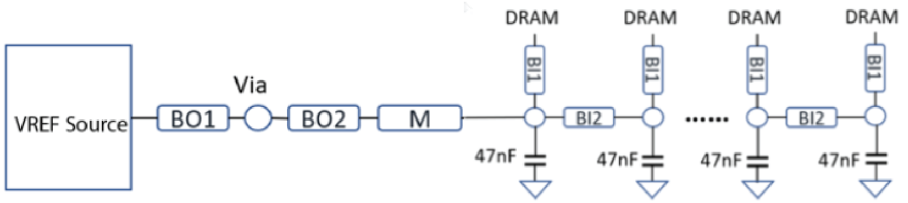
Figure 28. RESET Scheme for Single Rank x8 and Single Rank x16 Topologies



The Address/Command reference voltage input (V_{REF_CA}) must track the V_{TT} regulator output as closely as possible. There are two methods to achieve this:

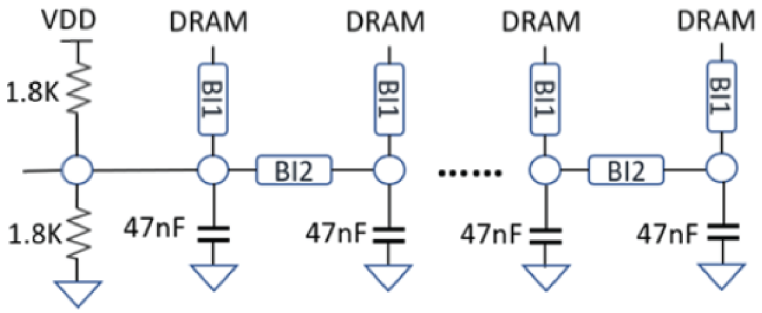
One method is to use a regulator that provides a dedicated tracking voltage reference output that can be connected directly to memory component V_{REF_CA} inputs, as shown in the figure below.

Figure 29. VTT Regulator Supplies V_{REF_CA} Output



A second method is to create a voltage divider using precision resistors. Place the resistor network in a location that is likely to track IR losses on the V_{DD} supply due to memory loading (that is, close to the V_{TT} regulator or memory components, rather than next to the V_{DD} regulator output). The following figure illustrates this configuration.

Figure 30. Resistor Divider Provides the V_{REF_CA} Signal



Intel recommends using a PCB trace width of at least 10 mils for V_{REF_CA} routing. The V_{REF_CA} signal must have at least 3×*h* spacing (where *h* is the distance or height from the trace to the nearest reference plane) to other nearby signals on the same layer.

6.3.5.4. Skew Matching Guidelines for DDR4 Discrete Configurations

This topic describes skew matching guidelines for single rank x 8 and single rank x 16 topologies.

Observe the following rules when skew matching DDR4 discrete configurations:

- Perform skew matching in time (picoseconds) rather than in actual trace length, to better account for via delays when signals are routed on different layers.
- Include both package per-pin skew and PCB delay when performing skew matching.
- Skew (length) matching for the alert signal is not required.

The following table provides skew matching guidelines for DDR4 down-memory topologies.

Table 93. Skew Matching Guidelines for DDR4 Discrete Topologies

DDR4 Device-down Length Matching Rules	Length Matching in Time (ps)
Skew matching between DQS and CLK	-85ps < CLK - DQS < 935ps
Skew matching between DQ and DQS within byte	-3.5ps < DQ - DQS < 3.5ps
Skew matching between DQS and DQS#	< 1ps
Skew matching between CLK and CLK#	< 1ps
Skew matching between CMD/ADDR/CTRL and Clock	-20ps < CLK - CMD/ADDR/CTRL < 20ps
Skew matching among CMD/ADDR/CTRL within each channel	< 20ps
Include package length in skew matching for FPGA device with no migration	Required
Include package length in skew matching for FPGA device with migration when all package net length are available	It is recommended to use the final migrated package net length
Include package length in skew matching for FPGA device with migration when all package net length are not available	Not recommended

6.3.5.5. Power Delivery Recommendations for DDR4 Discrete Configurations

This topic describes power distribution network (PDN) design guidelines for the memory side in discrete topologies.

Note: For information on power distribution network design at the FPGA to meet timing margins, refer to the Agilex 7 M-Series PDN design guidelines.

In the following table, the number of decoupling capacitors is based on a single channel. If multiple channels are sharing the same power rail, the number of decoupling capacitors at the memories must be scaled accordingly for all channels.

Physically small decoupling capacitors are recommended to minimize area, inductance, and resistance on the PDN path on the printed circuit board.

Table 94. Required Decoupling Capacitors on the PCB for the Memory Side

Memory Configuration	Power Domain	Decoupling Location	Quantity × Value (size)
Discrete (Component) Single Rank x8	VDDQ/VDD shorted	4 near each x8 DRAM device	36 x 1uF (0402)
		Distribute around DRAM devices	9 x 10uF (0603)
	VPP	2 near each x8 DRAM device	18 x 1uF (0402)
		Distribute around DRAM devices	5 x 10uF (0603)
	VTT	Place near Rtt (termination resistors)	16 x 1uF (0402)
		Place near Rtt (termination resistors)	4 x 10uF (0603)
Discrete (Component) Single Rank x16	VDDQ/VDD shorted	4 near each x16 DRAM device	18 x 1uF (0402)
		Distribute around DRAM devices	5 x 10uF (0603)
	VPP	2 near each x16 DRAM device	10 x 1uF (0402)
		Distribute around DRAM devices	3 x 10uF (0603)
	VTT	Place near Rtt (termination resistors)	8 x 1uF (0402)
		Place near Rtt (termination resistors)	2 x 10uF (0603)

6.3.5.6. Agilex 7 M-Series EMIF Pin Swapping Guidelines

In Agilex 7 M-Series devices, EMIF pin swapping is allowed under certain conditions.

A byte lane in an EMIF data byte includes 12 signal pins (pins 0,1,2,3,4,5,6,7,8,9,10,11) at the package level. These 12 x I/O pins are arranged into 6 groups of 2 pins each, called *pairs* (pair 0 for pins 0/1, pair 1 for pins 2/3, pair 2 for pins 4/5, pair 3 for pins 6/7, pair 4 for pins 8/9, and pair 5 for pins 10/11).

6.3.5.6.1. DDR4 Byte Lane Swapping

The data lane can be swapped when the byte-lanes are utilized as DQ/DQS pins. Byte lane swapping on utilized lanes is allowed when you swap all the DQ/DQS/DM/DBI pins in the same byte lane with the other utilized byte lane.

The rules for swapping DQ byte lane are as follows:

- You can only swap between utilized DQ lanes.
- You cannot swap a DQ lane with an AC lane.
- You cannot swap a DQ lane with an ECC lane when out-of-band ECC is enabled. For x40 interfaces, the highest-indexed DQ byte lane cannot be swapped.
- Additional restrictions apply when you use a x16 memory component:
  - You must place DQ group 0 and DQ group 1 on adjacent byte lanes, unless they are separated by AC Lanes. These 2 groups must be connected to the same x16 memory component.
  - You must place DQ group 2 and DQ group 3 on adjacent byte lanes, unless they are separated by AC Lanes. These 2 groups must be connected to the same x16 memory component.
  - If you use only one byte of the x16 memory component, you must use only the lower byte of the memory component.
- Additional restrictions apply in lockstep configuration implemented with 2 adjacent IO96 banks:
  - Any DQ lane with a letter *s* prefix must remain in its designated byte lane. You cannot swap an *s*prefixed DQ lane with any other byte-lane.
  - DQ lane swapping between IO96 banks is not allowed. You can only swap between utilized DQ lanes within the same IO96 bank.

Table 95. Byte Lane Swapping

Address/ Command Scheme	Data Width usage	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 2	DDR4 x32 + ECC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
Scheme 2	DDR4 x40	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]

Example: DDR4 x 32 +ECC implemented with AC Scheme 2

BL7 is used as ECC DQ lane, while Lane 0, 4, 5 and 6 are used DQ lanes. Byte lane swapping between BL0,4,5,6 is allowed.

Example: DDR4 x 40 implemented with AC Scheme 2

BL0,4,5,6,7 are used as DQ lanes. Byte lane swapping between BL0,4,5,6 is allowed. The highest-index DQ byte lane (that is, DQ[4]), cannot be swapped and must be placed at BL7.

Table 96. Byte Lane Swapping for Lockstep Configuration

A/C Placement Option		BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7		BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
		Primary								Secondary								
AC Pri Top Sub-Bank / Sec DQ Bot		DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]		DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
AC Pri Bot Sub-Bank / Sec DQ Bot		DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	sDQ[4]		DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
AC Pri Top Sub-Bank / Sec DQ Top(m)		DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]		GPIO	X	X	X	DQ[5]	DQ[6]	DQ[7]	DQ[8]

Example: x72 or x64+ECC Configuration with AC Pri Top Sub-Bank/Sec DQ Bot Placement

In Primary IO96 Bank: sDQ[0] is used for ECC or RUSER/WUSER. It cannot be swapped to other byte lanes.
Byte lane swapping between BL0, 1,2 and 3 is allowed.

In Secondary IO96 Bank: BL0,1,2,3 are used as DQ lanes. Byte lane swapping between BL0,1,2,3 is allowed.

Example: x72 or x64+ECC Configuration with AC Pri Bot Sub-Bank/Sec DQ Bot Placement

In Primary IO96 Bank: sDQ[4] is used for ECC or RUSER/WUSER. It cannot be swapped to other byte lanes.
Byte lane swapping between BL0, 4,5 and 6 is allowed.

In Secondary IO96 Bank: BL0,1,2,3 are used as DQ lanes. Byte lane swapping between BL0,1,2,3 is allowed.

Example: x72 or x64+ECC Configuration with AC Pri Top Sub-Bank / Sec DQ Top(m)

In Primary IO96 Bank: sDQ[0] is used for ECC or RUSER/WUSER. It cannot be swapped to other byte lanes.
Byte lane swapping between BL0, 1,2 and 3 is allowed.

In Secondary IO96 Bank: BL4,5,6,7 are used as DQ lanes. Byte lane swapping between BL4,5,6,7 is allowed.

6.3.5.6.2. DDR4 Address and Command and CLK Lane

Address and command and control signals in a bank cannot be swapped.

Pin mapping must adhere to the requirements defined in the table in the *Address and Command Pin Placement for DDR4* topic.

You cannot swap address and command lanes. You cannot swap among AC1/AC2/AC3/AC4 lanes. The address and command lane placement must adhere to the specific placement defined in the table in the *DDR4 Data Width Mapping* topic.

The T and C lanes for the CLKt/c cannot be swapped with each other, nor can the T and C lanes for the DQS-T/DQS-N be swapped with each other.

6.3.5.6.3. DDR4 Interface x8 Data Lane

A byte lane in an external memory interface consists of 12 signal pins, denoted 0-11.

For DDR4 interfaces composed of ×8 devices, two pins are reserved for DQS-T and DQS-C signals, one pin is reserved for the optional DM/DBI signal, one pin must be reserved, and the remaining eight pins are for DQ signals. One-byte data lane must be

assigned for each byte lane, where the byte lane covers DQ [0:7], DQS_T/DQS_C and DBI_N. The following are EMIF I/O pin swapping restrictions applicable to a DDR4 interface with a ×8 data lane:

- DQS_T must go to pin 4 in IO12 pins.
- DQS_C must go to pin 5 in IO12 pins.
- DBI_N must go to pin 6 in IO12 pins. If the interface does not use the DBI_N pin, this pin 6 in IO12 lane must remain unconnected.
- Pin 7 in IO12 lane remains unconnected. Intel recommends that you connect this pin 7 to the T_{DQS} dummy load of the memory component and route it as a differential trace along with DBI_N (pin 6). This facilitates ×4 or ×8 data interoperability in DIMMs configuration.
- You can connect data byte (DQ [0:7]) to any pins [0,1,2,3,8,9,10,11] in the byte lane. Any permutation within selected pins is permitted.

Table 97. Pin Swapping Rules for DDR4 x8 Interfaces

Pin Index Within Byte Lane	DDR4 x8 Data Lane Function	Swap Consideration
0	DQ Pin	Swap group A
1	DQ Pin	Swap group A
2	DQ Pin	Swap group A
3	DQ Pin	Swap group A
4	DQS-T Pin	Fixed location (not swappable)
5	DQS-C Pin	Fixed location (not swappable)
6	DM/DBI Pin	Fixed location (not swappable)
7	Unused	Fixed location (not swappable)
8	DQ Pin	Swap group A
9	DQ Pin	Swap group A
10	DQ Pin	Swap group A
11	DQ Pin	Swap group A

6.3.5.6.4. DDR4 Interface x4 Data Lane

For DDR4 x4 interfaces, two nibbles must be packed into the same IO12 lane.

Four pins are reserved for DQS_T and DQS_C signals and the remaining eight pins implement the DQ signals. The IO12 lane is divided into upper and lower halves to accommodate each nibble. You cannot swap signals belonging to one nibble with signals belonging to the other nibble. DQ signals within a nibble swap group may be swapped with each other. You may also swap entire nibbles—that is, nibble 0 and nibble 1—with each other provided the DQS pin functionality transfers to the correct pin locations. However, this process is not recommended for JEDEC-compliant DIMM interfaces, as it prohibits the interoperability between DIMMs constructed with x4 components and DIMMs constructed with x8 components.

The following table lists the supported pin functionality in x4 mode and the pins that may be swapped with each other. Pins belonging to the same swap group may be freely interchanged with each other.

Table 98. Pin Swapping Rules for DDR4 x4

Pin Index Within Byte Lane	DDR4 x4 Data Lane Function	Swap Consideration	
0	DQ Pin (lower nibble)	Swap group A	Nibble 0
1	DQ Pin (lower nibble)	Swap group A	
2	DQ Pin (lower nibble)	Swap group A	
3	DQ Pin (lower nibble)	Swap group A	
4	DQS_T Pin (lower nibble)	Fixed location (not swappable)	
5	DQS_C Pin (lower nibble)	Fixed location (not swappable)	
6	DQS_T Pin (upper nibble)	Fixed location (not swappable)	Nibble 1
7	DQS_C Pin (upper nibble)	Fixed location (not swappable)	
8	DQ Pin (upper nibble)	Swap group B	
9	DQ Pin (upper nibble)	Swap group B	
10	DQ Pin (upper nibble)	Swap group B	
11	DQ Pin (upper nibble)	Swap group B	

- Nibble 1 must correspond to DQS[17:9] on a physical JEDEC-compliant DIMM for x4/x8 interoperability.
- Nibbles 0 and 1 must follow the same skew matching rules among all 12 signals in the IO12 lane as are specified for a x8-based DQS group.

Note:

- Although the current version of the Quartus Prime software may not enforce all of the rules listed in the above table, be aware that all of these rules may be enforced in later versions of the software.
- At present, the Quartus Prime software checks the following:
  - Address and command pin placement, per the table in the *Address and Command Pin Placement for DDR4* topic, or the *Agilex 7 External Memory Interface Pin Information* file, which is available here: Pin-Out Files for Intel FPGA Devices.
  - For x8, the Quartus Prime software checks the following:
    - DQS T/C are on pin index 4 and pin index 5 in a byte lane.
    - DM/DBI is on pin index 6.
    - DQ[*x*] are on pin indices [11:8] and [3:0].
  - For x4, the Quartus Prime software checks the following:
    - DQS T/C on pin index 4 and pin index 5 and associated DQs are within the corresponding byte lane.
    - DQS T/C on pin index 6 and pin index 7 and associated DQs are within the corresponding byte lane.

    You are responsible for ensuring that these conditions are met.
- The Quartus Prime software does not currently check whether DQ pins associated with the lower nibble DQS are actually placed in pin[3:0] or whether DQ pins associated with the upper nibble DQS are actually placed in pin[11:8].

6.3.5.6.5. Pin Swizzling

For information on pin swizzling, refer to *Configuring DQ Pin Swizzling* in the *External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide*.

7. Agilex 7 M-Series FPGA EMIF IP – DDR5 Support

This chapter contains IP parameter descriptions and pin planning information for Agilex 7 M-Series FPGA external memory interface IP for DDR5.

7.1. Agilex 7 M-Series FPGA EMIF IP Parameters for DDR5

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP.

Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

7.1.1. Agilex 7 FPGA EMIF IP Parameter for DDR5

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 99. Group: General IP Parameters / High-Level Parameters

Display Name	Description
Technology Generation	Denotes the specific memory technology generation to be used Note: This parameter can be auto-computed. (Identifier: MEM_TECHNOLOGY)
Memory Format	Specifies the packaging format of the memory device (Identifier: MEM_FORMAT)
Memory Device Topology	Topology used by memory device (Identifier: MEM_TOPOLOGY)
Memory Ranks	Total number of physical ranks in the interface (Identifier: MEM_NUM_RANKS)
Number of Channels	Number of channels (Identifier: MEM_NUM_CHANNELS)
Device DQ Width	If the device is a DIMM: Specifies the full DQ width of the DIMM. If the interface is composed of discrete components: Specifies the DQ width of each discrete component. (Identifier: MEM_DEVICE_DQ_WIDTH)
Number of Components Per Rank	Number of components per rank. If each component contains more than one rank, then set this parameter to 1. (Identifier: MEM_COMPS_PER_RANK)

continued...

Display Name	Description
ECC Mode	Specifies the type of ECC (if any) and the required number of side-band bits per channel that will be used by this EMIF instance. While not all required side-band bits necessarily carry ECC bits, all need to be connected to the memory device. If enabling ECC requires more side-band bits than necessary ECC bits, then ECC bits are transmitted on the least significant side-band bits. Note: This parameter can be auto-computed. (Identifier: CTRL_ECC_MODE)
Total DQ Width	(Derived Parameter) This will be the width (in bits) of the mem_dq port on the memory interface. For a component interface, it is calculated based on: (MEM_COMPS_PER_RANK * MEM_DEVICE_DQ_WIDTH + (8 bits if Side-band ECC is enabled, or 8 bits if AXI4 User Data is enabled in fabric modes, or 4 bits if AXI4 User Data is enabled in NoC mode)) * MEM_NUM_CHANNELS For a DIMM-based interface, it is just MEM_DEVICE_DQ_WIDTH + (8 bits if side-band ECC is enabled, or 8 bits if AXI4 User Data is enabled in fabric modes, or 4 bits if AXI4 User Data is enabled in NoC mode) * MEM_NUM_CHANNELS. (Identifier: MEM_TOTAL_DQ_WIDTH)
Memory Clock Frequency	Specifies the operating frequency of the memory interface in MHz. If you change the memory frequency, you must select a matching Preset from the dropdown (or create a custom one), to update all the timing parameters. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FREQ_MHZ)
Instance ID	Instance ID of the EMIF IP. EMIF in the same bank, or connected to related user logic (e.g. to the same INIU), should have unique IDs in order to distinguish them when using the side-band interface. Valid values are 0-6. (Identifier: INSTANCE_ID)

Table 100. Group: General IP Parameters / Memory Device Preset Selection

Display Name	Description
Use Memory Device Preset from file	Specifies whether MEM_PRESET_ID will be a value from Quartus (if false), or a value from a custom preset file path (if true) (Identifier: MEM_PRESET_FILE_EN)
Memory Preset custom file path	Path to a .qprs file on the users disk (Identifier: MEM_PRESET_FILE_QPRS)
Memory Preset	The name of a preset that the user would like to load, describing the memory device that this emif will be targeting. Note: This parameter can be auto-computed. (Identifier: MEM_PRESET_ID)

Table 101. Group: General IP Parameters / Advanced Parameters / PHY / Topology

Display Name	Description
Use NOC	Specifies whether we are using the NOC or bypassing it Note: This parameter can be auto-computed. (Identifier: PHY_NOC_EN)
Asynchronous Enable	Specifies whether the user logic is clocked based on the clock provided by the IP (Sync), or by a separate user clock (Async). If true - async mode is used, if false - sync mode is used.

continued...

Display Name	Description
	(Identifier: PHY_ASYNC_EN)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms IO BANK and TOP vs BOT part of the IO BANK). Legal ranges are derived from device floorplan. By default (value=AUTO), the most optimal location is selected (to maximize available frequency and data width). Note: This parameter can be auto-computed. (Identifier: PHY_AC_PLACEMENT)
PLL Reference Clock Frequency	Specifies what PLL reference clock frequency the user will supply. It is recommended to use the fastest possible PLL reference clock frequency because it leads to better jitter performance. Note: This parameter can be auto-computed. (Identifier: PHY_REFCLK_FREQ_MHZ)

Table 102. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings

Display Name	Description
Voltage	The voltage level for the I/O pins driving the signals between the memory device and the FPGA memory interface. (Identifier: PHY_IO_VOLTAGE)

Table 103. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Address/Command

Display Name	Description
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the Address/Command Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_AC_X_R_S_AC_OUTPUT_OHM)

Table 104. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Memory Clock

Display Name	Description
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the CK Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_CLK_X_R_S_CK_OUTPUT_OHM)

Table 105. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Data Bus

Display Name	Description
I/O Standard	Specifies the I/O electrical standard for the data bus pins. The selected I/O standard configures the circuit within the I/O buffer to match the industry standard. (Identifier: GRP_PHY_DATA_X_DQ_IO_STD_TYPE)
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the DQ Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_DATA_X_R_S_DQ_OUTPUT_OHM)

continued...

Display Name	Description
Slew Rate	Specifies the slew rate of the data bus pins. The slew rate (or edge rate) describes how quickly the signal can transition, measured in voltage per unit time. *Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.* (Identifier: GRP_PHY_DATA_X_DQ_SLEW_RATE)
Input Termination	This parameter allows you to change the input on chip termination settings for the selected I/O standard on the DQ Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_DATA_X_R_T_DQ_INPUT_OHM)
Initial Vrefin	Specifies the **initial value for the reference voltage on the data pins(Vrefin)**. The specified value serves as a starting point and may be overridden by calibration to provide better timing margins. (Identifier: GRP_PHY_DATA_X_DQ_VREF)

Table 106. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / PHY Inputs

Display Name	Description
PLL Reference Clock Input Termination	This parameter allows you to change the input on chip termination settings for the selected I/O standard on the refclk input pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_IN_X_R_T_REFCLK_INPUT_OHM)

Table 107. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Decision Feedback Equalization (DFE)

Display Name	Description
DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_1)
DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_2)
DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_3)
DFE Tap 4	This parameter allows you to select the amount of bias used on tap 4 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_4)

Table 108. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Data Bus On-Die Termination (ODT)

Display Name	Description
Target Write Termination	Specifies the target termination to be used during a write (Identifier: GRP_MEM_ODT_DQ_X_TGT_WR)
Non-Target Write Termination	Specifies the termination to be used for the non-target rank in a multi-rank configuration during a write (Identifier: GRP_MEM_ODT_DQ_X_NON_TGT_WR)
Non-Target Read Termination	Specifies the termination to be used for the non-target rank in a multi-rank configuration during a read

continued...

Display Name	Description
	(Identifier: GRP_MEM_ODT_DQ_X_NON_TGT_RD)
Drive Strength	Specifies the termination to be used when driving read data from memory (Identifier: GRP_MEM_ODT_DQ_X_RON)

Table 109. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Data Bus Reference Voltage (Vref)

Display Name	Description
VrefDQ Value	Specifies the initial VrefDQ value to be used (Identifier: GRP_MEM_DQ_VREF_X_VALUE)

Table 110. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Address/Command Bus On-Die Termination (ODT)

Display Name	Description
CA Termination	Specifies the termination to be used for the CA bus. This setting only applies to Group B, Group A will always be unterminated. (Identifier: GRP_MEM_ODT_CA_X_CA)
CS Termination	Specifies the termination to be used for the CS bus. This setting only applies to Group B, Group A will always be unterminated. (Identifier: GRP_MEM_ODT_CA_X_CS)
CK Termination	Specifies the termination to be used for the CK bus. This setting only applies to Group B, Group A will always be unterminated. (Identifier: GRP_MEM_ODT_CA_X_CK)

Table 111. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Address/Command Bus Reference Voltage (Vref)

Display Name	Description
VrefCA Value	Specifies the initial VrefCA value to be used (Identifier: GRP_MEM_VREF_CA_X_CA_VALUE)
VrefCS Value	Specifies the initial VrefCS value to be used (Identifier: GRP_MEM_VREF_CA_X_CS_VALUE)

Table 112. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Decision Feedback Equalization (DFE)

Display Name	Description
DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the memory DFE (Identifier: GRP_MEM_DFE_X_TAP_1)
DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the memory DFE (Identifier: GRP_MEM_DFE_X_TAP_2)
DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the memory DFE (Identifier: GRP_MEM_DFE_X_TAP_3)
DFE Tap 4	This parameter allows you to select the amount of bias used on tap 4 of the memory DFE (Identifier: GRP_MEM_DFE_X_TAP_4)

Table 113. Group: General IP Parameters / Advanced Parameters / AXI Settings / AXI Interface Settings

Display Name	Description
Enable Debug Tools	If enabled, the AXI-L port will be connected to SLD nodes, allowing for a system-console avalon manager interface to interact with this AXI-L subordinate interface. (Identifier: DEBUG_TOOLS_EN)
AXI-Lite Port Access Mode	Specifies whether the AXI-Lite port is connected to the fabric, the NOC, or disabled Note: This parameter can be auto-computed. (Identifier: AXI_SIDEBAND_ACCESS_MODE)

Table 114. Group: General IP Parameters / Advanced Parameters / Additional Parameters / Additional String Parameters

Display Name	Description
User Extra Parameters	Semi-colon separated list of key/value pairs of extra parameters (Identifier: USER_EXTRA_PARAMETERS)

Table 115. Group: Example Design / Example Design

Display Name	Description
HDL Selection	This option lets you choose the format of HDL in which generated simulation and synthesis files are created. You can select either Verilog or VHDL. (Identifier: EX_DESIGN_HDL_FORMAT)
Synthesis	Generate Synthesis Example Design (Identifier: EX_DESIGN_GEN_SYNTH)
Simulation	Generate Simulation Example Design (Identifier: EX_DESIGN_GEN_SIM)
Core Clock Freq	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode) Note: This parameter can be auto-computed. (Identifier: EX_DESIGN_CORE_CLK_FREQ_MHZ)
Core Refclk Freq	PLL reference clock frequency in MHz for PLL supplying the core clock (Identifier: EX_DESIGN_CORE_REFCLK_FREQ_MHZ)
NOC Refclk Freq	NOC Refclk Freq for the NOC control IP Note: This parameter can be auto-computed. (Identifier: EX_DESIGN_NOC_REFCLK_FREQ_MHZ)
Traffic Generator Remote Access	Specifies whether the traffic generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled (Identifier: EX_DESIGN_HYDRA_REMOTE)

Table 116. Group: Example Design / Performance Monitor

Display Name	Description
Enable performance monitoring	Enable performance monitor on all channels for measuring read/write transaction metrics (Identifier: EX_DESIGN_PMON_ENABLED)

Table 117. Traffic Generator Program

Display Name	Description
Traffic Generator Program	Specifies the traffic pattern to run (Identifier: EX_DESIGN_HYDRA_PROG)

7.1.2. Agilex 7 FPGA EMIF Memory Device Description IP (DDR5) Parameter Descriptions

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 118. Group: Configuration Save

Display Name	Description
Configuration Filepath	Filepath to Save to (.qprs extension) (Identifier: MEM_CONFIG_FILE_QPRS)

Table 119. Group: High-Level Parameters

Display Name	Description
Memory Format	Specifies the packaging format of the memory device (Identifier: MEM_FORMAT)
Enable Data Mask	Specifies whether byte masking is to be enabled by the memory. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_DM_EN)
Density of Each Memory Component	Specifies the density of each memory component in Gbits. (Identifier: DDR5_MEM_DEVICE_DENSITY_GBITS)

Table 120. Group: Memory Interface Parameters / Data Bus

Display Name	Description
Device DQ Width	If the device is a DIMM: Specifies the full DQ width of the DIMM. If the interface is composed of discrete components: Specifies the DQ width of each discrete component. (Identifier: MEM_DEVICE_DQ_WIDTH)
Memory Component Data Width	Specifies the data width of the memory component in bits. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_COMPONENT_DQ_WIDTH)
Burst Length	Specifies the burst length of the memory interface. (Identifier: DDR5_MEM_DEVICE_BURST_LENGTH)

Table 121. Group: Memory Interface Parameters / Device Topology

Display Name	Description
Number of Ranks per DIMM	Number of ranks per DIMM Note: This parameter can be auto-computed.

continued...

Display Name	Description
	(Identifier: MEM_RANKS_PER_DIMM)
Number of Channels Per DIMM	Number of channels per DIMM Note: This parameter can be auto-computed. (Identifier: MEM_CHANNELS_PER_DIMM)
Device Bank Group Address Width	Specifies the width of the bank group address. (Identifier: DDR5_MEM_DEVICE_BANK_GROUP_ADDR_WIDTH)
Device Bank Address Width	Specifies the width of the bank address. (Identifier: DDR5_MEM_DEVICE_BANK_ADDR_WIDTH)
Device Row Address Width	Specifies the width of the row address. (Identifier: DDR5_MEM_DEVICE_ROW_ADDR_WIDTH)
Device Column Address Width	Specifies the width of the column address. (Identifier: DDR5_MEM_DEVICE_COL_ADDR_WIDTH)
Number of Differential Memory Clock Pairs	Specifies the width of clock interface according to the number of ranks. (Identifier: DDR5_MEM_DEVICE_CK_WIDTH)

Table 122. Group: Memory Timing Parameters / Timing Parameters

Display Name	Description
Memory Clock Frequency	Specifies the **operating frequency** of the memory interface in MHz. If you change the memory frequency, you must select a matching Preset from the dropdown (or create a custom one), to update all the timing parameters. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FREQ_MHZ)
Memory Speed Bin	Specifies the memory speed bin. (Identifier: DDR5_MEM_DEVICE_SPEEDBIN)
Memory Read Latency	Specifies the read latency of the memory interface in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_CL_CYC)
Memory Write Latency	Specifies the write latency of the memory interface in cycles. (Identifier: DDR5_MEM_DEVICE_CWL_CYC)
Memory Fine Granularity Refresh Mode	Specifies the Fine Granularity Refresh (FGR) mode of the memory interface. (Identifier: DDR5_MEM_DEVICE_FINE_GRANULARITY_REFRESH_MODE)

Table 123. Group: Memory Timing Parameters / Advanced Timing Parameters

Display Name	Description
tREFI1	Specifies the maximum average refresh interval in normal refresh mode in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TREFI1_NS)
tREFI2	Specifies the maximum average refresh interval in fine granularity refresh mode in nanoseconds. (Identifier: DDR5_MEM_DEVICE_TREFI2_NS)
tREFISB	Specifies the maximum average refresh interval in fine granularity and same bank refresh mode in nanoseconds. (Identifier: DDR5_MEM_DEVICE_TREFISB_NS)
tCCD_S	Specifies the CAS_n to CAS_n command delay for different bank group in cycles.

continued...

Display Name	Description
	Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCCD_S_CYC)
tCCD_L	Specifies the CAS_n to CAS_n command delay for same bank group in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCCD_L_CYC)
tCCD_L_WR	Specifies the write CAS_n to write CAS_n command delay for same bank group in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCCD_L_WR_CYC)
tCCD_L_WR2	Specifies the write CAS_n to write CAS_n command delay for same bank group and the second write is not RMW, in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCCD_L_WR2_CYC)
tRRD_S	Specifies the Activate-to-Activate command delay to different bank group for 1KB page size in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRRD_S_CYC)
tRRD_L	Specifies the Activate-to-Activate command delay to same bank group for 1KB page size in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRRD_L_CYC)
tFAW	Specifies the four activate window for 1KB page size in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TFAW_NS)
tRFC1	Specifies the refresh operation delay in normal refresh mode in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRFC1_NS)
tRFC2	Specifies the refresh operation delay in fine granularity refresh mode in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRFC2_NS)
tRFCSB	Specifies the refresh operation delay in fine granularity and same bank refresh mode in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRFCSB_NS)
tRCD	Specifies the Activate-to-internal-Read-or-Write delay in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRCD_NS)
tRP	Specifies the row precharge time in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRP_NS)
tRAS	Specifies the Activate-to-Precharge command period in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRAS_NS)
tRC (tRAS+tRP)	Specifies the Activate-to-Activate or Refresh command period in nanoseconds. (Identifier: DDR5_MEM_DEVICE_TRC_NS)

continued...

Display Name	Description
tREFSBRD	Specifies the same bank refresh to activate delay in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TREFSBRD_NS)
tWR	Specifies the write recovery time in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TWR_NS)
tZQLAT	Specifies the ZQ calibration latch time in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TZQLAT_CYC)
tZQCAL	Specifies the ZQ calibration time in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TZQCAL_NS)
tMRR	Specifies the Mode Register Read (MRR) command period in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TMRR_CYC)
tMRR_P	Specifies the Mode Register Read (MRR) pattern to mode register read pattern command spacing in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TMRR_P_CYC)
tMRW	Specifies the Mode Register Write (MRW) command period in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TMRW_CYC)
tMRD	Specifies the Mode Register Set (MRS) command delay in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TMRD_CYC)
tDFE	Specifies the Decision Feedback Equalization (DFE) Mode Register Write update delay time in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TDFE_NS)
tDLLK	Specifies the timing of DLLK in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TDLLK_CYC)
tWTR_S	Specifies the delay from start of internal write transaction to internal read command for different bank group in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TWTR_S_NS)
tWTR_L	Specifies the delay from start of internal write transaction to internal read command for same bank group in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TWTR_L_NS)
tRTP	Specifies the internal read command to precharge command delay in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TRTP_CYC)
tPPD	Specifies the Precharge-to-Precharge delay in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TPPD_CYC)
tPD	Specifies the minimum power down time in cycles.

continued...

Display Name	Description
	Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TPD_CYC)
tACTPDEN	Specifies the timing of Activate command to power down entry command in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TACTPDEN_CYC)
tPRPDEN	Specifies the timing of Precharge All Banks (PREab), Precharge Same Bank (PREsb), or Normal Precharge (PREpb) to power down entry command in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TPRPDEN_CYC)
tREFPDEN	Specifies the timing of Refresh All Banks (REFab) or Refresh Same Bank (REFsb) command to power down entry command in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TREFPDEN_CYC)
tXP	Specifies the exit power down to next valid command in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TXP_CYC)
tCPDED	Specifies the command pass disable delay in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCPDED_CYC)
tCSL	Specifies the Self-Refresh CS_n low pulse width in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCSL_NS)
tCKSRX	Specifies the valid clock requirement before SRX in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCKSRX_CYC)
tCSH_SREXIT	Specifies the self-refresh exit CS_n high pulse width in nanoseconds. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TCSH_SREXIT_NS)
tDQSCK	Specifies the DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TDQSCK_CYC)
tWPRE_EN	Specifies the write preamble enable window in cycles. The window size depends on the write preamble mode. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TWPRE_EN_CYC)
tDQSS	Specifies the host and system voltage/temperature drift window of first rising DQS_t preamble edge relative to CAS Write Latency (CWL) CK_t-CK_c edge in cycles. Note: This parameter can be auto-computed. (Identifier: DDR5_MEM_DEVICE_TDQSS_CYC)

7.1.3. Agilex 7 FPGA EMIF Calibration IP Parameter Descriptions

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 124. Group: High-Level Parameters

Display Name	Description
Unique Instance ID for the Calibration IP	Instance ID (Identifier: INSTANCE_ID)
Calibration IP is part of Bank Adjacent Pair	Calibration IP is part of Bank Adjacent Pair (Identifier: IS_PART_OF_BANK_ADJACENT_PAIR)
Number of Peripheral IPs	Number of Peripheral IPs (EMIFs, PHYLites) to be calibrated (Identifier: NUM_CALBUS_PERIPHS)
Number of Standalone I/O PLLs	Number of Standalone I/O PLLs to calibrate (Identifier: NUM_CALBUS_PLLS)
AXI-L Subordinate Port Mode	AXI-L subordinate port can be disabled, or can be used in one of two modes: directly exported to fabric, or connect to the NoC (i.e. to a TNIU) (Identifier: PORT_S_AXIL_MODE)

7.2. Agilex 7 M-Series FPGA EMIF IP Pin and Resource Planning

The following topics provide guidelines on pin placement for external memory interfaces.

Typically, all external memory interfaces require the following FPGA resources:

- Interface pins
- PLL and clock network
- RZQ pins
- Other FPGA resources—for example, core fabric logic, and debug interfaces

Once all the requirements are known for your external memory interface, you can begin planning your system.

7.2.1. Agilex 7 M-Series FPGA EMIF IP Interface Pins

Any I/O banks that do not support transceiver operations in Agilex 7 M-Series FPGAs support external memory interfaces. However, DQS (data strobe or data clock) and DQ (data) pins are listed in the device pin tables and are fixed at specific locations in the device. You must adhere to these pin locations to optimize routing, minimize skew, and maximize margins. Always check the pin table for the actual locations of the DQS and DQ pins.

Note: Maximum interface width varies from device to device depending on the number of I/O pins and DQS or DQ groups available. Achievable interface width also depends on the number of address and command pins that the design requires. To ensure adequate PLL, clock, and device routing resources are available, you should always test fit any IP in the Quartus Prime software before PCB sign-off.

Intel devices do not limit the width of external memory interfaces beyond the following requirements:

- Maximum possible interface width in any particular device is limited by the number of DQS groups available.
- Sufficient clock networks are available to the interface PLL as required by the IP.
- Sufficient spare pins exist within the chosen bank or side of the device to include all other address and command, and clock pin placement requirements.

Note: The greater the number of banks, the greater the skew, hence Intel recommends that you always generate a test project of your desired configuration and confirm that it meets timing.

7.2.1.1. Estimating Pin Requirements

You should use the Quartus Prime software for final pin fitting. However, you can estimate whether you have enough pins for your memory interface using the EMIF Device Selector on www.intel.com, or perform the following steps:

1. Determine how many read/write data pins are associated per data strobe or clock pair.
2. Calculate the number of other memory interface pins needed, including any other clocks (write clock or memory system clock), address, command, and RZQ. Refer to the External Memory Interface Pin Table to determine necessary Address/Command/Clock pins based on your desired configuration.
3. Calculate the total number of I/O banks required to implement the memory interface, given that an I/O bank supports up to 96 pins.

Test the proposed pin-outs with the rest of your design in the Quartus Prime software (with the correct I/O standard and OCT connections) before finalizing the pin-outs. There can be interactions between modules that are illegal in the Quartus Prime software that you might not know about unless you compile the design and use the Quartus Prime Pin Planner.

7.2.1.2. DIMM Options

DDR5 unbuffered DIMMs (UDIMMs) and small outline DIMMs (SODIMMs) command and address pins are clocked at single data rate (SDR). DDR5 registered DIMMs (RDIMMs) command and address pins are clocked at double data rate (DDR). The table below shows a pin comparison of UDIMM, SODIMM, and RDIMM modules up to dual rank. You should always check your memory vendor's data sheet to be sure.

Table 125. UDIMM, SODIMM, and RDIMM Pin Options for DDR5

Pins	UDIMM Pins	SODIMM Pins	RDIMM Pins
Data	72 bit DQ[31:0]_A DQ[31:0]_B CB[3:0]_A CB[3:0]_B	72 bit DQ[31:0]_A DQ[31:0]_B CB[3:0]_A CB[3:0]_B	80 bit DQ[31:0]_A DQ[31:0]_B CB[7:0]_A CB[7:0]_B
Data Mask	DM[3:0]_A_n(1) DM[3:0]_B_n(1)	DM[3:0]_A_n(1) DM[3:0]_B_n(1)	DM[4:0]_A_n(1) DM[4:0]_B_n(1)

continued...

Pins	UDIMM Pins	SODIMM Pins	RDIMM Pins
Data Strobe	x8: DQS[4:0]_A_t DQS[4:0]_A_c DQS[4:0]_B_t DQS[4:0]_B_c	x8: DQS[4:0]_A_t DQS[4:0]_A_c DQS[4:0]_B_t DQS[4:0]_B_c	x8: DQS[4:0]_A_t DQS[4:0]_A_c DQS[4:0]_B_t DQS[4:0]_B_c x4: DQS[9:0]_A_t DQS[9:0]_A_c DQS[9:0]_B_t DQS[9:0]_B_c
Command / Address	CA[12:0}_A CA[12:0}_B CA[1:0}_A_n CA[1:0}_B_n	CA[12:0}_A CA[12:0}_B CA[1:0}_A_n CA[1:0}_B_n	CA[6:0]_A CA[6:0]_B CS[1:0]_A_n CS[1:0]_B_n
Clock	CK[1:0]_A_t CK[1:0]_A_c CK[1:0]_B_t CK[1:0]_B_c	CK[1:0]_A_t CK[1:0]_A_c CK[1:0]_B_t CK[1:0]_B_c	CK_t CK_c
Parity	ALERT_n	ALERT_n	ALERT_n PAR_A PAR_B
Other Pins	RESET_n HSDA, HSCL, HSA	RESET_n HSDA, HSCL, HSA	RESET_n HSDA, HSCLL, HSA LBD/RSP_A_n LBS/RSP_B_n

Notes to Table:
1. DM pins are available only for DIMMs constructed using x8 or greater components.
2. The Agilex 7 M-Series memory controller supports up to 2 ranks per channel. Agilex 7 M-Series devices support only 1 DIMM per channel (1DPC).

7.2.1.3. Maximum Number of Interfaces

The maximum number of interfaces supported for a given memory protocol varies, depending on the FPGA in use.

Unless otherwise noted, the calculation for the maximum number of interfaces is based on independent interfaces where the address or command pins are not shared.

Note: You may need to share PLL clock outputs depending on your clock network usage.

For interface information for Agilex 7 M-Series devices, consult the EMIF Device Selector on www.intel.com.

Timing closure depends on device resource and routing utilization. For more information about timing closure, refer to the *Area and Timing Optimization Techniques* chapter in the *Quartus Prime Handbook*.

7.2.2. Agilex 7 M-Series FPGA EMIF IP Resources

The Agilex 7 M-Series FPGA memory interface IP uses several FPGA resources to implement the memory interface.

7.2.2.1. OCT

You require an OCT calibration block if you are using an Agilex 7 M-Series FPGA OCT calibrated series, parallel, or dynamic termination for any I/O in your design. There are two OCT blocks in an I/O bank, one for each sub-bank.

You must observe the following requirements when using OCT blocks:

- The I/O bank where you place the OCT calibration block must use the same V_{CCIO_PIO} voltage as the memory interface.
- The OCT calibration block uses a single fixed R_{ZQ}. You must ensure that an external termination resistor is connected to the correct pin for a given OCT block.

For specific pin connection requirements, refer to *Specific Pin Connection Requirements*.

7.2.2.2. PLL

When using PLL for external memory interfaces, you must consider the following guidelines:

For the clock source, use the clock input pin specifically dedicated to the PLL that you want to use with your external memory interface. The input and output pins are only fully compensated when you use the dedicated PLL clock input pin.

For specific pin connection requirements, refer to *Specific Pin Connection Requirements*.

7.2.3. Pin Guidelines for Agilex 7 M-Series FPGA EMIF IP

The Agilex 7 M-Series FPGA contains I/O banks on the top and bottom edges of the device, which can be used by external memory interfaces.

Agilex 7 M-Series FPGA I/O banks contain 96 I/O pins. Each bank is divided into two sub-banks with 48 I/O pins in each. Sub-banks are further divided into four I/O lanes, where each I/O lane is a group of twelve I/O ports.

The I/O bank, I/O lane, and pairing pin for every physical I/O pin can be uniquely identified by the following naming convention in the device pin table:

- The I/O pins in a bank are represented as P#X#Y#, where:
  - P# represents the pin number in a bank. It ranges from P0 to P95, for 96 pins in a bank.
  - X# represents the bank number on a given edge of the device. X0 is the farthest bank from the zipper.
  - Y# represents the top or bottom edge of the device. Y0 and Y1 refer to the I/O banks on the bottom and top edge, respectively.
- Because an IO96 bank comprises two IO48 sub-banks, all pins with P# value less than 48 (P# <48) belong to the same I/O sub-bank. All other pins belong to the second IO48 sub-bank.
- The *Index Within I/O Bank* value falls within one of the following ranges: 0 to 11, 12 to 23, 24 to 35, or 36 to 47, and represents one of I/O lanes 0, 1, 2, or 3, respectively.
- To determine whether I/O banks are adjacent, you can refer to the sub-bank-ordering figures for your device family in the *Architecture: I/O Bank* topic. In general, you can assume that I/O banks are adjacent within an I/O edge, unless the I/O bank is not bonded out on the package (indicated by the presence of the " - " symbol in the I/O table), or if the I/O bank does not contain 96 pins, indicating that it is only partially bonded out. If an I/O bank is not fully bonded out in a particular device, it cannot be included within the span of sub-banks for a larger external memory interface. In all cases, you should use the Quartus Prime software to verify that your usage can be implemented.
- The pairing pin for an I/O pin is in the same I/O bank. You can identify the pairing pin by adding 1 to its *Index Within I/O Bank* number (if it is an even number), or by subtracting 1 from its *Index Within I/O Bank* number (if it is an odd number).

7.2.3.1. General Guidelines - DDR5

You should follow the recommended guidelines when performing pin placement for all external memory interface pins targeting Agilex 7 M-Series devices, whether you are using the hard memory controller or your own solution.

Note: QDR*x* is not supported with HPS.

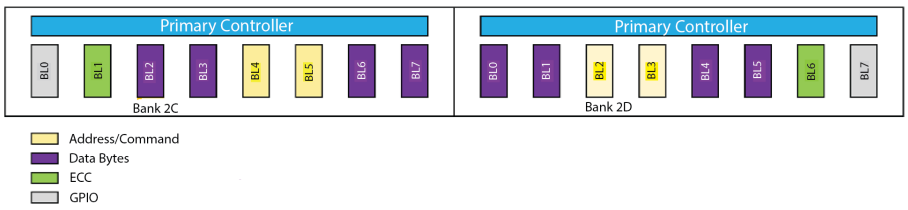
Observe the following general guidelines when placing pins for your Agilex 7 M-Series external memory interface:

1. Ensure that the pins of a single external memory interface reside on the same edge I/O.
2. The address and command pins and their associated clock pins in the address and command bank must follow a fixed pin-out scheme, as defined in the table in the *Address and Command Pin Placement for DDR5* topic.
3. Not every byte lane can function as an address and command lane or a data lane. The pin assignment must adhere to the DDR5 data width mapping defined in *DDR5 Data Width Mapping*.
4. A byte lane must not be used by both address and command pins and data pins.

5. An external memory interface can occupy one or more banks on the same edge. When an interface must occupy multiple banks, ensure that those banks are adjacent to one another.
   - If an I/O bank is shared between two interfaces—meaning that two sub-banks belong to two different EMIF interfaces—then both the interfaces must share the same voltage.
   - Sharing of I/O lanes within a sub-bank for two different EMIF interfaces is not permitted; I/O lanes within a sub-bank can be assigned to one EMIF interface only.
6. Any pin in the same bank that is not used by an external memory interface may not be available for use as a general purpose I/O pin:
   - For fabric EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same sub-bank, pins in an I/O lane that is not assigned to an EMIF interface, can be used as general-purpose I/O pins.
   - For HPS EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same bank, pins in an I/O lane that is not assigned to an EMIF interface cannot be used as general-purpose I/O pins either.
7. All address and command pins and their associated clock pins (CK_t and CK_c) must reside within a single sub-bank. The sub-bank containing the address and command pins is identified as the address and command sub-bank. Refer to the table in *DDR5 Data Width Mapping* for the supported address and command and data lane placements for DDR5.
8. The address and command pins and their associated clock pins in the address and command bank must follow a fixed pin-out scheme, as defined in the *Agilex 7 M-Series External Memory Interface Pin Information* file.
9. An external memory interface can occupy one or more banks on the same edge. When an interface must occupy multiple banks, ensure the following:
   - That the banks are adjacent to one another.
   - That you used only the supported data width mapping as defined in the table in *DDR5 Data Width Mapping*. Be aware that not every byte lane can be used as an address and command lane or a data lane.

   The following figure shows one possible pin placement for a DDR5 2ch x32 + ECC interface on Bank 2C and Bank 2D.

Figure 31. x72 DDR5 Pin Placement using Bank 2C and 2D



10. An unused I/O lane in the address and command sub-bank can serve to implement a data group, such as a x8 DQS group. The data group must be from the same controller as the address and command signals.
11. An I/O lane must not be used by both address and command pins and data pins.

12. Place read data groups according to the DQS grouping in the pin table and Pin Planner. Read data strobes (such as DQS_t and DQS_c) must reside at physical pins capable of functioning as DQS_t and DQS_c for a specific read data group size. You must place the associated read data pins (DQ), within the same group.

    Note: For DDR5 interfaces with x4 components, place DQ pins and DQS entirely in either the upper or lower half of a 12-bit bank sub-group. Consult the pin table for your device to identify the association between DQ pins and DQS pins for x4 mode operation. Additional restrictions apply for x4/x8 DIMM interoperability.

13. One of the sub-banks in the device (typically the sub-bank within corner bank 3A) may not be available if you use certain device configuration schemes. For some schemes, there may be an I/O lane available for EMIF data group.
    - AVST-8 – This is contained entirely within the SDM, therefore all lanes of sub-bank 3A can be used by the external memory interface.
    - AVST-16/AVST-32– Lanes 4, 5, 6, and 7 are all effectively occupied and are not usable by the external memory interface.

14. Two memory interfaces cannot share an I/O 48 sub-bank.

7.2.3.2. x4 DIMM Implementation

DIMMS using a x4 DQS configuration require remapping of the DQS signals to achieve compatibility between the EMIF IP and the JEDEC standard DIMM socket connections.

The necessary remapping is shown in the table below. You can implement this DQS remapping in either RTL logic or in your schematic wiring connections.

Table 126. Mapping of DQS Signals Between DIMM and the EMIF IP

DIMM			Quartus Prime EMIF IP	
DQS0_A	DQ[3:0]_A		DQS0	DQ[3:0]_A
DQS5_A	DQ[7:4]_A		DQS1	DQ[7:4]_A
DQS1_A	DQ[11:8]_A		DQS2	DQ[11:8]_A
DQS6_A	DQ[15:12]_A		DQS3	DQ[15:12]_A
DQS2_A	DQ[19:16]_A		DQS4	DQ[19:16]_A
DQS7_A	DQ[23:20]_A		DQS5	DQ[23:20]_A
DQS3_A	DQ[27:24]_A		DQS6	DQ[27:24]_A
DQS8_A	DQ[31:28]_A		DQS7	DQ[31:28]_A
DQS4_A	CB[3:0]_A		DQS8	CB[3:0]_A
DQS9_A	CB[7:4]_A		DQS9	CB[7:4]_A
DQS0_B	DQ[3:0]_B		DQS10	DQ[3:0]_B
DQS5_B	DQ[7:4]_B		DQS11	DQ[7:4]_B
DQS1_B	DQ[11:8]_B		DQS12	DQ[11:8]_B
DQS6_B	DQ[15:12]_B		DQS13	DQ[15:12]_B
DQS2_B	DQ[19:16]_B		DQS14	DQ[19:16]_B

continued...

DIMM			Quartus Prime EMIF IP	
DQS7_B	DQ[23:20]_B		DQS15	DQ[23:20]_B
DQS3_B	DQ[27:24]_B		DQS16	DQ[27:24]_B
DQS8_B	DQ[31:28]_B		DQS17	DQ[31:28]_B
DQS4_B	CB[3:0]_B		DQS18	CB[3:0]_B
DQS9_B	CB[7:4]_B		DQS19	CB[7:4]_B

Data Bus Connection Mapping Flow

1. Connect all FPGA DQ pins accordingly to DIMM DQ pins. No remapping is required.
2. DQS/DQS*n* remapping is required either on the board schematics or in the RTL code.

When designing a board to support x4 DQS groups, Intel recommends that you make it compatible for x8 mode, for the following reasons:

- Provides the flexibility of x4 and x8 DIMM support.
- Allows use of x8 DQS group connectivity rules.
- Allows use of x8 timing rules for matching. Adhere to x4/x8 interoperability rules when designing a DIMM interface, even if the primary use case is to support x4 DIMMs only, because doing so facilitates debug and future migration capabilities. Regardless, the rules for length matching for two nibbles in a x4 interface must match those of the signals for a corresponding x8 interface, as the data terminations are turned on and off at the same time for both x4 DQS groups in an I/O lane. If the two x4 DQS groups were to have significantly different trace delays, it could adversely affect signal integrity. Trace delays for two nibbles packed within the IO12 lanes are matched using the same guidelines as a single x8 byte lane.

7.2.3.3. Specific Pin Connection Requirements

PLL

For DDR5, you must constrain the PLL reference clock to the address and command lanes only.

- You must constrain differential reference clocks to pin indices 10 and 11 in lane 2 when placing command address pins in lane 3 and lane 2.
- You must constrain differential reference clocks to pin indices 10 and 11 in lane 4 when placing command address pins in lane 5 and lane 4.
- The sharing of PLL reference clocks across multiple DDR5 interfaces is permitted within an I/O bank.

Note: Lane 3:0 is the bottom sub-bank and lane 7:4 is the top sub-bank.

OCT

For DDR5, you must constrain the RZQ pin to the address and command lanes only.

- You must constrain RZQ to pin index 2 in lane 3 when placing command address pins in lane 3 and lane 2.
- You must constrain RZQ to pin index 2 in lane 5 when placing command address pins in lane 5 and lane 4.
- The sharing of RZQ across multiple DDR5 interfaces is permitted within an I/O bank.

Note: Lane 3:0 is the bottom sub-bank and lane 7:4 is the top sub-bank.

Address / Command / Parity

For DDR5, you must constrain the ALERT_N pin to the address and command lanes only.

- You must constrain ALERT_N to pin index 1 in lane 3 when placing command address pins in lane 3 and lane 2.
- You must constrain ALERT_N to pin index 1 in lane 5 when placing command address pins in lane 5 and lane 4.

Note: Lane 3:0 is the bottom sub-bank and lane 7:4 is the top sub-bank.

DQS/DQ/DM

For DDR5 x8 DQS/DQ/DM grouping, the following rules apply:

- You may use pin indices 0, 1, 2, 3, 8, 9, 10, and 11 within a lane for DQ mode pins only.
- You must use pin index 4 for the DQS_t pin only.
- You must use pin index 5 for the DQS_c pin only.
- You must ensure that pin index 7 remains unused. Pin index 7 is not available for use as a general purpose I/O.
- You must use pin index 6 for the DM pin only.

For DDR5 x4 DQS/DQ/DM grouping, the following rules apply:

- You may use pin indices 0, 1, 2, and 3 within a lane for DQ mode pins for the lower nibble only. Pin rotation within this group is permitted.
- You must use pin index 4 for the DQS_t pin only of the lower nibble.
- You must use pin index 5 for the DQS_c pin only of the lower nibble.
- You may use pin indices 8, 9, 10, and 11 within a lane for the DQ mode pins only for the upper nibble. Pin rotation within this group is permitted.
- You must use pin index 6 for the DQS_t pin only of the upper nibble.
- You must use pin index 7 for the DQS_c pin only of the upper nibble.

7.2.3.4. Command and Address Signals

Command and address signals in SDRAM devices are clocked into the memory device using the CK_T or CK_C signal. These pins operate at single data rate (SDR) using only one clock edge. The number of address pins depends on the SDRAM device capacity. The address pins are multiplexed, so two clock cycles are required to send the row, column, and bank address.

Although DDR5 operates in fundamentally the same way as other SDRAM, there are no dedicated pins for RAS_N, CAS_N, and WE_N, as those are shared with higher-order address pins. DDR5 has CS_N, CKE, ODT, and RESET_N pins, similar to DDR4. DDR5 also has some additional pins, including the ACT_N (activate) pin and BG (bank group) pins.

7.2.3.5. Clock Signals

DDR5 SDRAM devices use CK_t and CK_c signals to clock the address and command signals into the memory.

The memory uses these clock signals to generate the DQS signal during a read through the DLL inside the memory.

7.2.3.6. Data, Data Strobes, DM, and Optional ECC Signals

DDR5 SDRAM devices use bidirectional differential data strobes. Differential DQS operation enables improved system timing due to reduced crosstalk and less simultaneous switching noise on the strobe output drivers. The DQ pins are also bidirectional.

DQ pins in DDR5 SDRAM interfaces can operate in either ×4 or ×8 mode DQS groups, depending on your chosen memory device or DIMM, regardless of interface width. The ×4 and ×8 configurations use one pair of bidirectional data strobe signals, DQS and DQSn, to capture input data.

The DQ signals are edge-aligned with the DQS signal during a read from the memory and are center-aligned with the DQS signal during a write to the memory. The memory controller shifts the DQ signals by –90 degrees during a write operation to center align the DQ and DQS signals. The PHY IP delays the DQS signal during a read, so that the DQ and DQS signals are center aligned at the capture register. Intel devices use a phase-locked loop (PLL) to center-align the DQS signal with respect to the DQ signals during writes and use dedicated DQS phase-shift circuitry to shift the incoming DQS signal during reads.

The memory device's setup (t_{DS}) and hold times (t_{DH}) for the DQ and DM pins during writes are relative to the edges of DQS write signals and not the CK or CK# clock. Setup and hold requirements are not necessarily balanced.

The DQS signal is generated on the positive edge of the system clock to meet the t_{DQSS} requirement. DQ and DM signals use a clock shifted –90 degrees from the system clock, so that the DQS edges are centered on the DQ or DM signals when they arrive at the SDRAM. The DQS, DQ, and DM board trace lengths need to be tightly matched.

The SDRAM uses the DM pins during a write operation. Driving the DM pins low shows that the write is valid. The memory masks the DQ signals if the DM pins are driven high. To generate the DM signal, Intel recommends that you use the spare DQ pin within the same DQS group as the respective data, to minimize skew.

The DM signal's timing requirements at the SDRAM input are identical to those for DQ data. The DDR registers, clocked by the –90 degree shifted clock, create the DM signals.

Some SDRAM modules support error correction coding (ECC) to allow the controller to detect and automatically correct errors in data transmission. UDIMMs or SODIMMs with ECC will have CB[3:0] bits per sub channel. Depending on the RDIMM module you can have CB[7:0] or CB[3:0] bits per sub channel.

7.2.4. Pin Placements for Agilex 7 M-Series FPGA DDR5 EMIF IP

7.2.4.1. Address and Command Pin Placement for DDR5

Table 127. Address and Command Pin Placement

Address/Command Lane	Index Within Byte Lane	Scheme 1 UDIMM/SODIMM/ Component	Scheme 2 RDIMM
AC1	11	CK_C[1]/SCL(i3c)	SCL (i3c)
	10	CK_T[1]/SCL(i3c)	SDA (i3c)
	9	CS_N[0]	CS_N[0]
	8	CS_N[1]	CS_N[1]
	7	CK_C[0]	CK_C[0]
	6	CK_T[0]	CK_T[0]
	5	CA[12]	
	4	CA[11]	
	3	RESET_N	RESET_N
	2	RZQ Site	
	1	ALERT_N	ALERT_N
	0	CA[10]	
AC0	11	Differential "N-Side" reference clock input site	
	10	Differential "P-Side" reference clock input site	
	9	CA[9]	LBD, RSP_A_n
	8	CA[8]	LBS, RSP_B_n
	7	CA[7]	PAR_A
	6	CA[6]	CA[6]
	5	CA[5]	CA[5]
	4	CA[4]	CA[4]
	3	CA[3]	CA[3]
	2	CA[2]	CA[2]
	1	CA[1]	CA[1]
	0	CA[0]	CA[0]

The Agilex 7 M-Series FPGA DDR5 IP supports fixed Address and Command pin placement as shown in the above table. The IP supports up to 2 ranks for the following schemes:

- Scheme 1 supports component, UDIMM, and SODIMM.
- Scheme 2 supports RDIMM.

7.2.4.2. DDR5 Data Width Mapping

The EMIF IP for Agilex 7 M-Series devices does not support flexible data lane placement.

Only fixed byte lanes within the I/O bank can be used as data lanes. The following table lists the supported address and command and data lane placements in an I/O bank.

Table 128. Component

Controller	Address Command Scheme	Data Width Usage	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Primary	Scheme 1	DDR5x16	GPIO	GPIO	GPIO	GPIO	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Secondary	Scheme 1	DDR5x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	GPIO	GPIO	GPIO	GPIO
Primary & Secondary	Scheme 1	DDR5 2x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Primary	Scheme 1	DDR5x16 + ECC	GPIO	GPIO	GPIO	DQ[ECC] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Primary	Scheme 1	DDR5x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Primary	Scheme 1	DDR5x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P

Note: • P Primary controller.
• S Secondary controller.

DIMM Support

Table 129. Bank 3A, 3D, 2D

Address Command Scheme	Data Width per Channel	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 1	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC[0] P	DQ[0] P	DQ[1] P
Scheme 1	DDR5 x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC[0] P	DQ[0] P	DQ[1] P
Scheme 2	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC[0] P	DQ[0] P	DQ[1] P
Scheme 2	DDR5 x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC[0] P	DQ[0] P	DQ[1] P

Note: P Primary controller.

Table 130. Bank 3B, 3C, 2C

Address Command Scheme	Data Width per Channel	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 1	DDR5 x32	DQ[1] P	DQ[0] P	AC1 P	AC0 P	DQ[2] P	DQ[3] P	GPIO 1	GPIO 1
Scheme 1	DDR5 x32 + ECC	DQ[1] P	DQ[0] P	AC1 P	AC0 P	DQ[2] P	DQ[3] P	DQ[ECC] P	GPIO 1
Scheme 2	DDR5 x32	DQ[1] P	DQ[0] P	AC1 P	AC0 P	DQ[2] P	DQ[3] P	GPIO 1	GPIO 1
Scheme 2	DDR5 x32 + ECC	DQ[1] P	DQ[0] P	AC1 P	AC0 P	DQ[2] P	DQ[3] P	DQ[ECC] P	GPIO 1

Note:
- 1 GPIO is available if using NoC access mode.
- P Primary controller.
- For banks 3A/3B, 3C/3D, or 2C/2D there are two options available for the address and command placement parameter (**AC Placement**):
  - **Ch0 Top Sub-Bank/Ch1 Bot Sub-Bank**, must be selected when placing Ch0 A/C pins in the top sub-bank.
  - **Ch0 Bot Sub-Bank/Ch1 Top Sub-Bank**, must be selected when placing Ch0 A/C pins in the bottom sub-bank.

Table 131. Bank 2A

Address Command Scheme	Data Width per Channel	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 1	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 1	DDR5 x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 2	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 2	DDR5 x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P

Note: P Primary controller.

Table 132. Bank 2B

Address Command Scheme	Data Width per Channel	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 1	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 1	DDR5 x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 2	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 2	DDR5 x32 + ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P

Note:
- P Primary controller.
- For banks 2A and 2B, select **Ch0 Bot Sub-Bank/Ch1 Bot Sub-Bank** for the address and command placement parameter (**AC Placement**).

7.2.5. Agilex 7 M-Series EMIF Pin Swapping Guidelines

In Agilex 7 M-Series devices, EMIF pin swapping is allowed under certain conditions.

A byte lane in an EMIF data byte includes 12 signal pins (pins 0,1,2,3,4,5,6,7,8,9,10,11) at the package level. These 12 x I/O pins are arranged into 6 groups of 2 pins each, called *pairs* (pair 0 for pins 0/1, pair 1 for pins 2/3, pair 2 for pins 4/5, pair 3 for pins 6/7, pair 4 for pins 8/9, and pair 5 for pins 10/11).

7.2.5.1. DDR5 Byte Lane Swapping

The data lane can be swapped when the byte-lanes are utilized as DQ/DQS pins. Byte lane swapping on utilized lanes is allowed when you swap all the DQ/DQS/DM pins in the same byte lane with the other utilized byte lane.

The rules for swapping DQ byte lane are as follows:

- You can only swap between utilized DQ lanes.
- You cannot swap a DQ lane with an AC lane.
- You cannot swap a DQ lane with an ECC lane when out-of-band ECC is enabled. For x40 interfaces, the highest-indexed DQ byte lane cannot be swapped.
- Additional restrictions apply when you use a x16 memory component:
  - You must place DQ group 0 and DQ group 1 on adjacent byte lanes, unless they are separated by AC Lanes. These 2 groups must be connected to the same x16 memory component.
  - You must place DQ group 2 and DQ group 3 on adjacent byte lanes, unless they are separated by AC Lanes. These 2 groups must be connected to the same x16 memory component.
  - If you use only one byte of the x16 memory component, you must use only the lower byte of the memory component.

Table 133. Byte Lane Swapping

Address/ Command Scheme	Data Width per Channel	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 1	DDR5 x32	GPIO	GPIO	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Scheme 1	DDR5 x32+ ECC	GPIO	DQ[ECC] P	DQ[3] P	DQ[2] P	AC1 P	AC0 P	DQ[0] P	DQ[1] P
Note: P = Primary controller.									

Example 1: DDR5 x32

BL0, 1, 4, 5 are used as DQ lanes. Byte lane swapping is allowed.

Example 2: DDR5 x32 + ECC

BL6 is used as ECC DQ lane, while BL0, 1, 4, and 5 are used as DQ lanes. Byte lane swapping is allowed on BL0, 1, 4, and 5 only.

7.2.5.2. DDR5 Address and Command and CLK Lane

Address and command and control signals in a bank cannot be swapped.

Pin mapping must adhere to the requirements defined in the table in the *Address and Command Pin Placement for DDR5* topic.

You cannot swap address and command lanes. You cannot swap among AC0/AC1 lanes. The address and command lane placement must adhere to the specific placement defined in the table in the *DDR5 Data Width Mapping* topic.

The T and C lanes for the CK_T/_C cannot be swapped with each other, nor can the T and C lanes for the DQS_T/DQS_C be swapped with each other.

7.2.5.3. DDR5 Interface x8 Data Lane

A byte lane in an external memory interface consists of 12 signal pins, denoted 0-11.

For DDR5 interfaces composed of ×8 devices, two pins are reserved for DQS-T and DQS-C signals, one pin is reserved for the optional DM signal, one pin must be reserved, and the remaining eight pins are for DQ signals. One-byte data lane must be assigned for each byte lane, where the byte lane covers DQ [0:7], DQS_T/DQS_C and DM_N. The following are EMIF I/O pin swapping restrictions applicable to a DDR5 interface with a ×8 data lane:

- DQS_T must go to pin 4 in IO12 pins.
- DQS_C must go to pin 5 in IO12 pins.
- DM_N must go to pin 6 in IO12 pins. If the interface does not use the DM_N pin, this pin 6 in IO12 lane must remain unconnected.
- Pin 7 in IO12 lane remains unconnected. Intel recommends that you connect this pin 7 to the T_{DQS} dummy load of the memory component and route it as a differential trace along with DM_N (pin 6). This facilitates ×4 or ×8 data interoperability in DIMMs configuration.
- You can connect data byte (DQ [0:7]) to any pins [0,1,2,3,8,9,10,11] in the byte lane. Any permutation within selected pins is permitted.

Table 134. Pin Swapping Rules for DDR5 x8 Interfaces

Pin Index Within Byte Lane	DDR5 x8 Data Lane Function	Swap Consideration
0	DQ Pin	Swap group A
1	DQ Pin	Swap group A
2	DQ Pin	Swap group A
3	DQ Pin	Swap group A
4	DQS_T Pin	Fixed location (not swappable)
5	DQS_C Pin	Fixed location (not swappable)
6	DM Pin	Fixed location (not swappable)
7	Unused	Fixed location (not swappable)
8	DQ Pin	Swap group A
9	DQ Pin	Swap group A
10	DQ Pin	Swap group A
11	DQ Pin	Swap group A

7.2.5.4. DDR5 Interface x4 Data Lane

For DDR5 x4 interfaces, two nibbles must be packed into the same IO12 lane.

Four pins are reserved for DQS_T and DQS_C signals and the remaining eight pins implement the DQ signals. The IO12 lane is divided into upper and lower halves to accommodate each nibble. You cannot swap signals belonging to one nibble with signals belonging to the other nibble. DQ signals within a nibble swap group may be swapped with each other. You may also swap entire nibbles—that is, nibble 0 and nibble 1—with each other provided the DQS pin functionality transfers to the correct pin locations. However, this process is not recommended for JEDEC-compliant DIMM interfaces, as it prohibits the interoperability between DIMMs constructed with x4 components and DIMMs constructed with x8 components.

The following table lists the supported pin functionality in x4 mode and the pins that may be swapped with each other. Pins belonging to the same swap group may be freely interchanged with each other.

Table 135. Pin Swapping Rules for DDR5 x4

Pin Index Within Byte Lane	DDR5 x4 Data Lane Function	Swap Consideration	
0	DQ Pin (lower nibble)	Swap group A	Nibble 0
1	DQ Pin (lower nibble)	Swap group A	
2	DQ Pin (lower nibble)	Swap group A	
3	DQ Pin (lower nibble)	Swap group A	
4	DQS_T Pin (lower nibble)	Fixed location (not swappable)	
5	DQS_C Pin (lower nibble)	Fixed location (not swappable)	
6	DQS_T Pin (upper nibble)	Fixed location (not swappable)	Nibble 1
7	DQS_C Pin (upper nibble)	Fixed location (not swappable)	
8	DQ Pin (upper nibble)	Swap group B	
9	DQ Pin (upper nibble)	Swap group B	
10	DQ Pin (upper nibble)	Swap group B	
11	DQ Pin (upper nibble)	Swap group B	

- Nibble 1 must correspond to DQS[17:9] on a physical JEDEC-compliant DIMM for x4/x8 interoperability.
- Nibbles 0 and 1 must follow the same skew matching rules among all 12 signals in the IO12 lane as are specified for a x8-based DQS group.

Note:

- Although the current version of the Quartus Prime software may not enforce all of the rules listed in the above table, be aware that all of these rules may be enforced in later versions of the software.
- At present, the Quartus Prime software checks the following:
  - Address and command pin placement, per the table in the *Address and Command Pin Placement for DDR5* topic, or the *Agilex 7 External Memory Interface Pin Information* file, which is available here: Pin-Out Files for Intel FPGA Devices.
  - For x8, the Quartus Prime software checks the following:
    - DQS T/C are on pin index 4 and pin index 5 in a byte lane.
    - DM is on pin index 6.
    - DQ[*x*] are on pin indices [11:8] and [3:0].
  - For x4, the Quartus Prime software checks the following:
    - DQS T/C on pin index 4 and pin index 5 and associated DQs are within the corresponding byte lane.
    - DQS T/C on pin index 6 and pin index 7 and associated DQs are within the corresponding byte lane.

    You are responsible for ensuring that these conditions are met.
- The Quartus Prime software does not currently check whether DQ pins associated with the lower nibble DQS are actually placed in pin[3:0] or whether DQ pins associated with the upper nibble DQS are actually placed in pin[11:8].

7.2.5.5. Pin Swizzling

For information on pin swizzling, refer to *Configuring DQ Pin Swizzling* in the *External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide*.

7.3. DDR5 Board Design Guidelines

This section provides board layout design recommendations and guidelines for Agilex 7 M-Series FPGAs, with GPIO-B (input/output) silicon implementation to support DDR5.

This PCB layout guideline covers various supported DDR5 topologies along with maximum supported data rate that you can use for a successful PCB design.

A successful PCB design requires not only following the topology and routing guidelines here, but must also meet PDN design requirements.

For related information, refer also to the Agilex 7 F, I, and M-Series PDN design guidelines and the Agilex 7 high speed transceiver PCB design guidelines, available on the Intel website.

7.3.1. PCB Stack-up and Design Considerations

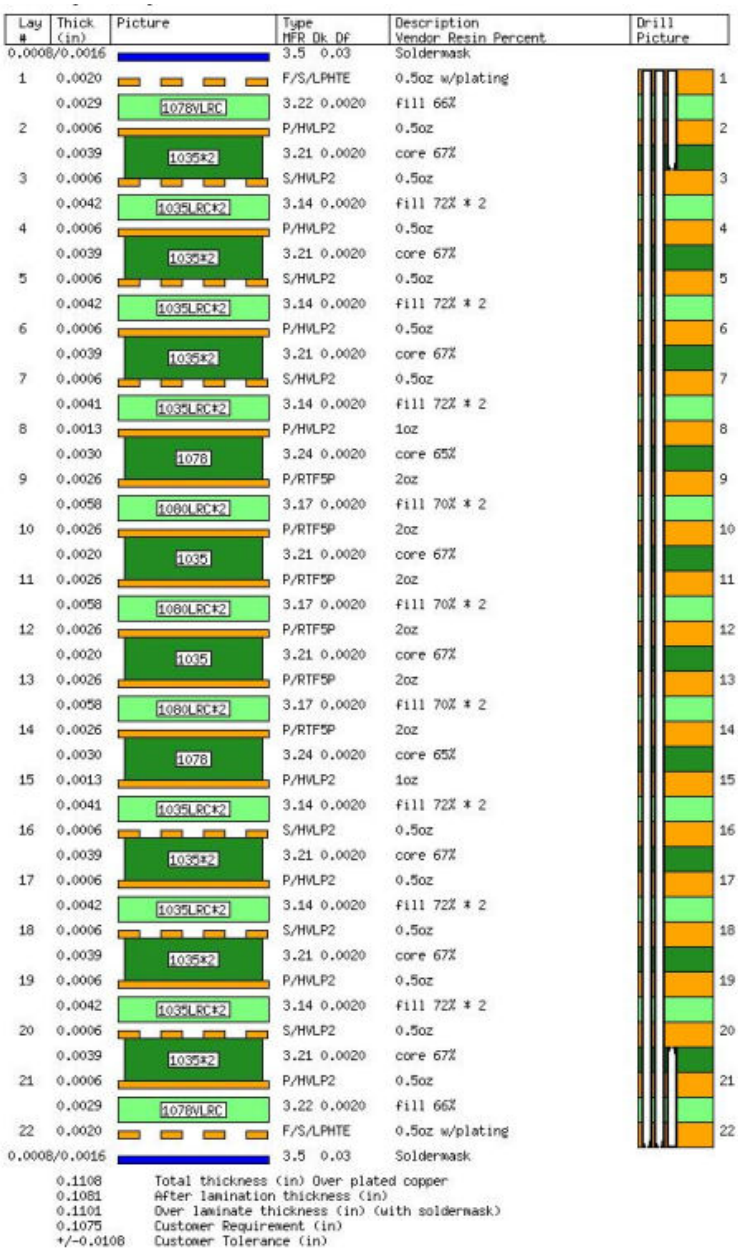
The following figure shows an example of an 18-layer PCB stackup that has been used for DDR5 on an Intel platform board. You may use other stackups (thin such as PCIE board, or thick board), provided you follow the recommendations in these guidelines.

Figure 32. 18-Layer Thin Board Type-4 PCB with Micro Via, Stacked Via, Buried Via and Through Via

Layer		Cu Weight	Structure	Proposed Thickness (mils)	Structure	Dk @ 1Ghz	Df @ 1Ghz	Copper type	Ref
	Soldermask			0.50		3.6	0.0195		
L1	Top	1/3oz+Plating		1.60				RTF	L2
	Prepreg			2.20	1035 RC 72	2.9	0.001		
L2	Plane	1/3oz+Plating		0.65				RTF	
				3.11	1078 RC70	2.9	0.001		
L3	Signal	1/3oz+Plating		0.65				RTF	L2&4
				3.30	1078 RC70	2.9	0.001		
L4	Plane	1/3oz+Plating		0.65				RTF	
				3.02	1078 RC70	2.9	0.001		
L5	Signal	1/3oz+Plating		0.80				RTF	L4&6
				3.32	1078 RC70	2.9	0.001		
L6	Plane	Hoz		0.60				HVLP	
				2.00	2.0mil core	2.9	0.001		
L7	Signal	Hoz		0.60				HVLP	L6&8
				2.96	1078 RC70	2.9	0.001		
L8	Plane	Hoz		0.60				RTF	
				2.00	2.0mil core	2.9	0.001		
L9	Plane	2oz		2.40				RTF	
				3.35	2 x 1035 RC 72	2.9	0.001		
L10	Plane	2oz		2.40				RTF	
				2.00	2.0mil core	2.9	0.001		
L11	Plane	Hoz		0.60				RTF	
				2.96	1078 RC70	2.9	0.001		
L12	Signal	Hoz		0.60				HVLP	L11&13
				2.00	2.0mil core	2.9	0.001		
L13	Plane	Hoz		0.60				HVLP	
				3.32	1078 RC70	2.9	0.001		
L14	Signal	1/3oz+Plating		0.80				RTF	L13&15
				3.02	1078 RC70	2.9	0.001		
L15	Plane	1/3oz+Plating		0.65				RTF	
				3.30	1078 RC70	2.9	0.001		
L16	Signal	1/3oz+Plating		0.65				RTF	L15&17
				3.11	1078 RC70	2.9	0.001		
L17	Plane	1/3oz+Plating		0.65				RTF	
	Prepreg			2.20	1035 RC 72	2.9	0.001		
L18	Bottom	1/3oz+Plating		1.60				RTF	L17
	Soldermask			0.50		3.6	0.0195		
Finished Thickness (mils)				65.27					

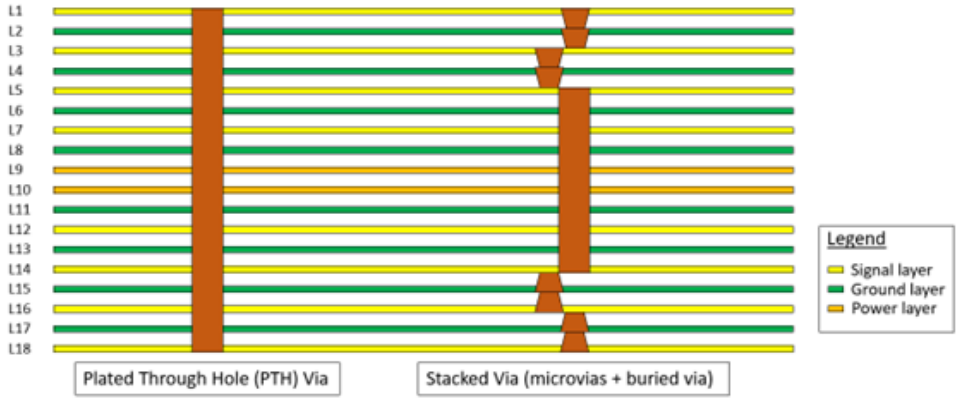
The following figure shows an example of a 22-layer thick PCB stackup, as used with some Intel platform boards and development kits.

Figure 33. 22-Layer Thick Type-4 Board Stack-up, High Performance with Microvia and PTH, with and without Backdrill



A high-quality type-4 PCB uses not only plated-through-hole (PTH) vias to connect from the top to bottom layer, but also stacked vias, micro vias, and buried vias to connect between layers. For example, a full-height stacked via of an 18 layer PCB consists of a combination of dual-stacked micro vias and buried vias. The following figure shows a cross-sectional comparison of a PTH and stacked via.

Figure 34. Cross-sectional Comparison of Plated-Through Hole Via and Stacked Via



To support maximum data rate operation, DDR5 board design requires a high-quality PCB stackup using micro vias, buried vias, or stacked vias to reduce crosstalk for high performance. Reducing the length of signal via is essential to minimizing the crosstalk between signals.

A type-3 PCB with zero-built up layers and PTH vias which is used to implement a DDR4 design can also be used for DDR5 designs.

7.3.2. General Design Considerations

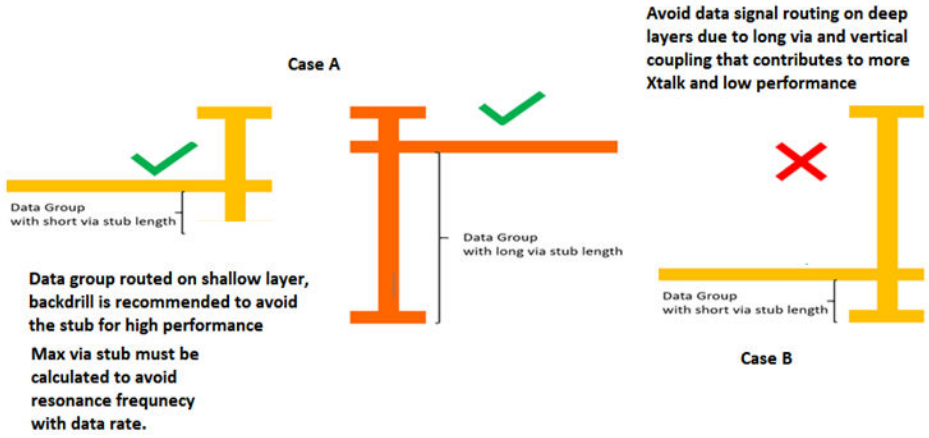
Intel recommends that you route all data signals within a specific group on the same layer.

The following figure illustrates a routing example for a type-3 PCB for a DDR5 design. Intel recommends that you route Data Group signals such as DQ, DM and DQS on shallow layers as stripline, with the least Z-height via transition to avoid vertical crosstalk for high performance.

The recommended routing layers for Data Group on an 18-layer board using plated-through vias are on the top half of the PCB, such as layers 3, 5, and 7. Other signals such as CA, CTRL, and clock signals can be routed with longer Z-height via transitions on the bottom half of the PCB, such as layers 12, 14, and 16.

Minimal stub effect or back drill is recommended but not mandatory to avoid high reflection for maximum data rate performance for a DDR5 interface. Long via stubs will affect the intersymbol interference (ISI) of the channel, but the impact of ISI is less than the impact of crosstalk for maximum performance.

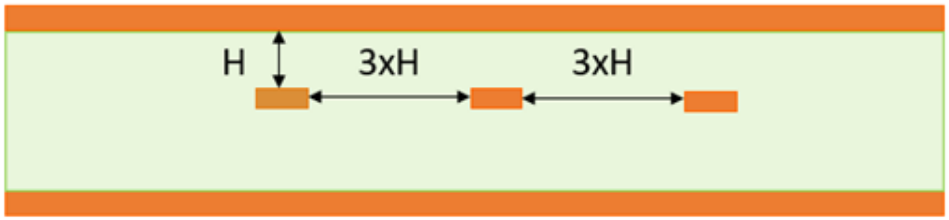
Figure 35. Suggested Routings



In the above figure, case A routing is suggested for DDR5 Data Group signals over case B, to support maximum data rate. If data signals are routed on deeper layers (as in case B, with long via and short stub), the impact of crosstalk is significant and causes reduced data rate and performance.

To minimize crosstalk horizontally between the signals on the same layer, PCB designers must maintain adequate signal trace-to-trace (edge to edge) space with a minimum spacing of 3 x *H* separation distance, where *H* is the dielectric thickness to the closest reference plane, as illustrated below.

Figure 36. Minimum Trace-to-Trace Separation Distance



7.3.3. DDR Differential Signals Routing

DQS and CLK signals in the DDR interface are differential signals and must be routed on PCB as differential signals unless there is a limitation for PCB routing, such as having a very small pitch at DRAM.

You should have a symmetrical fan-out routing at the FPGA pin field. Non-symmetrical routing for differential signals causes shifting on common-mode voltage and contributes to reduced timing margins at the receiver. The following figures show the recommended differential routing at the FPGA pin field for DQS/CLK signals.

Figure 37. Symmetrical Routing of Differential Signals (DQS/CLK) at FPGA Pin Field, with Length/Skew Matching Between P/N Lanes After FPGA Device Edge

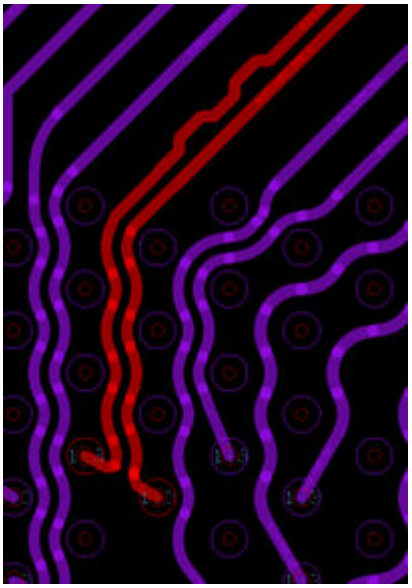
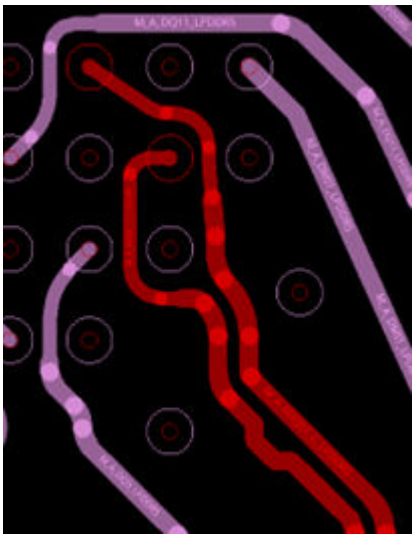


Figure 38. Single-Ended Routing for Differential Signals (DQS/CLK) at DRAM Pin Field with Very Small Pitch and Skew Matching at Edge of DRAM Pin Field



Intel recommends implementing length and skew matching for differential signals immediately after the FPGA device to avoid additional shifting on differential signals common mode voltage.

In cases where very small DRAM device pitch limits the implementation of symmetrical routing at the DRAM pin field for differential signals, it is recommended to route the differential signals as single-ended signals within the DRAM pin field, ensuring to maintain the same impedance while changing from differential to single-ended

configuration. Designers must also keep the same length of routing for each P and N single-ended lane within the DRAM pin field. The skew matching between P and N lanes must be applied before reaching the DRAM pin field.

7.3.4. Ground Plane and Return Path

A continuous and solid ground-reference plane is crucial for data lines, to ensure good signal integrity.

It is important to provide a low-impedance ground return path between the FPGA and DRAM devices, and to keep ground stitching vias within 80 mils from signal transition, for best return path on signal vias and improved signal integrity.

7.3.5. RDIMM, UDIMM, and SODIMM Break-in Layout Guidelines

The following figures show recommended signal and ground cutout and via transition, and placement patterns for RDIMMs, UDIMMs, and SODIMMs.

Intel recommends placing the vias on the connector pad, or with a short trace connected to connector pad. The placement of transition vias is critical to avoid stubs during transition to connector pads. It is also recommended to void under connector pads to avoid large capacitance and reflection in this area, and to control the impedance at the DIMM pads.

Figure 39. Recommended Signal and Ground Cutouts

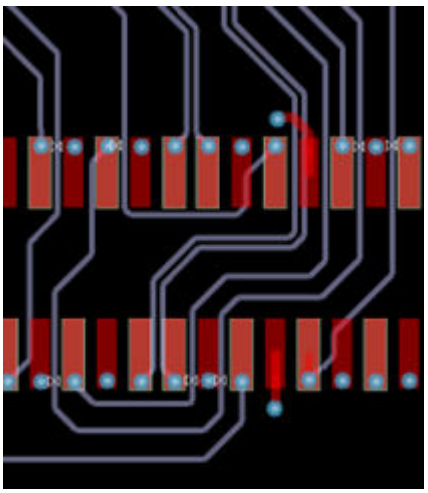
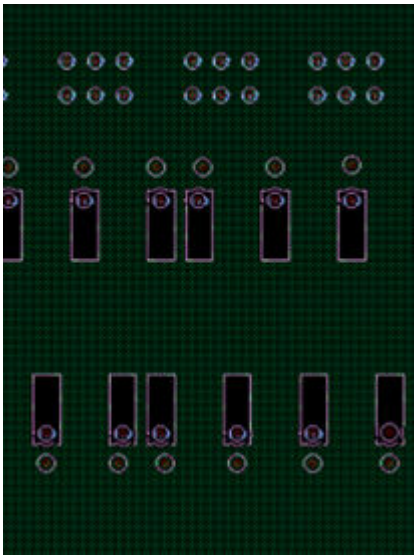


Figure 40. Recommended Signal and Ground Via Transitions

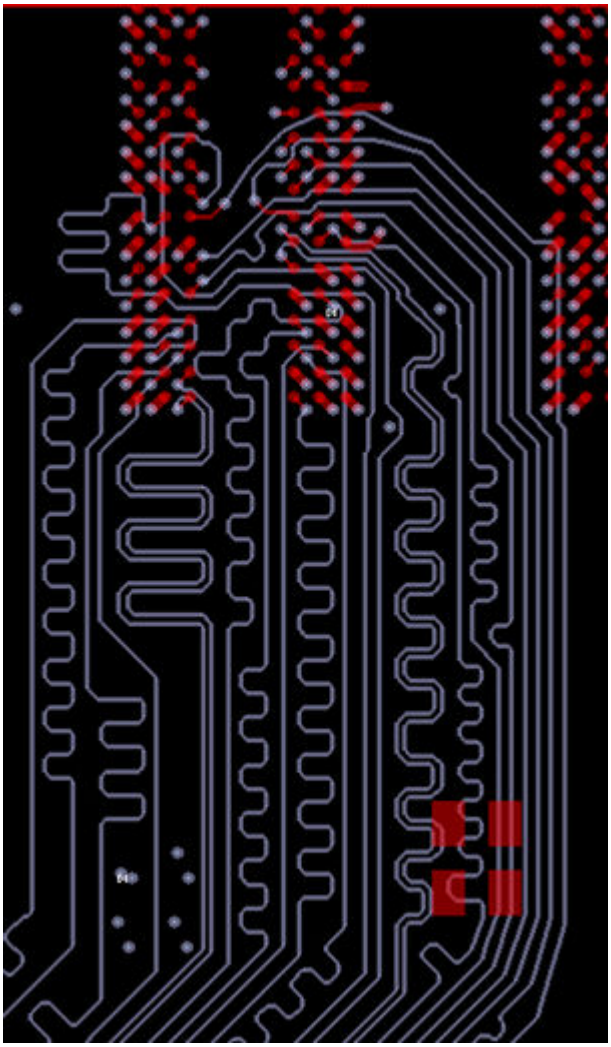


7.3.6. DRAM Break-in Layout Guidelines

For discrete DRAM components on PCB, you can use either the dog-bone or via in pad at the DRAM for the signal transition from inner layer to DRAM.

If you use dog-bone via transitions, you should separate them with larger pitch, to avoid crosstalk between the signal vias.

Figure 41. Data Signal Group Routing on PCB to Memory Down Configuration



7.3.7. DDR5 PCB Layout Guidelines

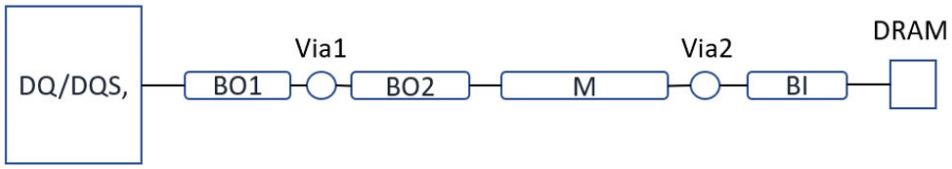
This section describes PCB layout guidelines for a DDR5 interface.

Agilex 7 M-Series devices support DDR5 interfaces for both discrete components and DIMMs, RDIMMs, SODIMMs, and LRDIMMs, with both thin and thick PCB stackups. The maximum supported data rates vary depending on the selected topology and thickness of circuit board.

7.3.7.1. DDR5 Discrete Component/Memory Down Topology: up to 40-Bit Interface (1 Rank x8 or x16, 2 Rank x8 or x16)

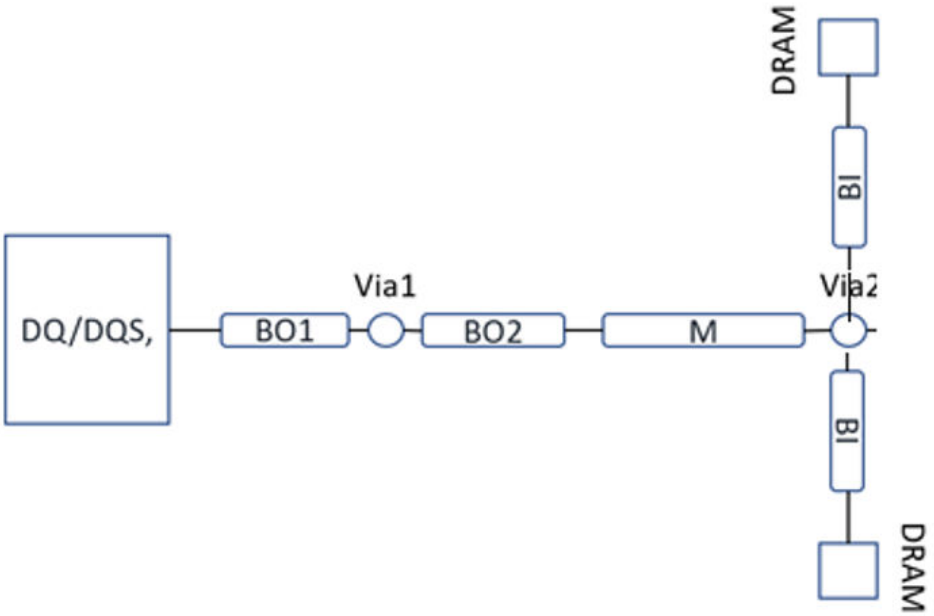
Data Group includes Data Strobe and its complement (DQS and DQS#), Data (DQ), and Data Mask (DM). The connection from the FPGA to DRAM is point-to-point topology as shown in the figure below, for single rank.

Figure 42. DRAM x8 or x16 (Single Rank)



Double rank topology has clamshell/fly-by configuration, as shown in the figure below.

Figure 43. DRAM x8 or x16 (Dual Rank)



For address, command, control and clock signals, a fly-by or clamshell topology as shown in the figure below is recommended to meet signal-integrity performance and for easier routing. The termination approach for DDR5 is through programmable on-die-termination (ODT).

Figure 44. Single Rank, DRAM x 8 bits, 40-bit Interface

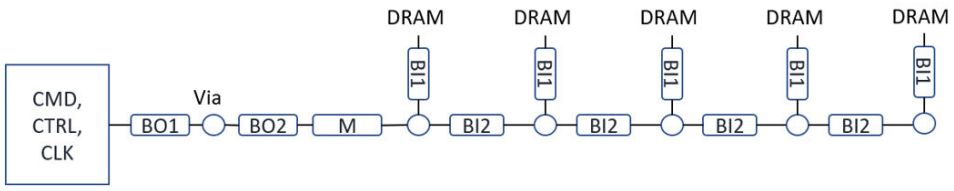


Figure 45. Dual Rank, DRAM x 8 bits, 40-bit Interface

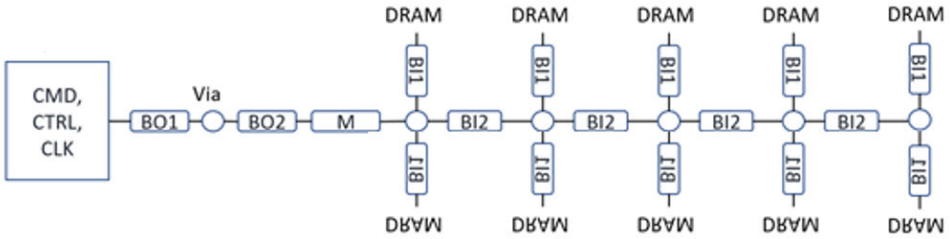


Figure 46. Single Rank, DRAM x 16 bits, 40-bit Interface (only 8 bits ECC of the last DRAM are used)

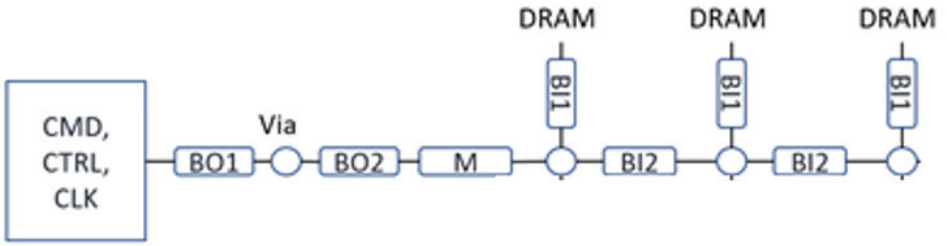
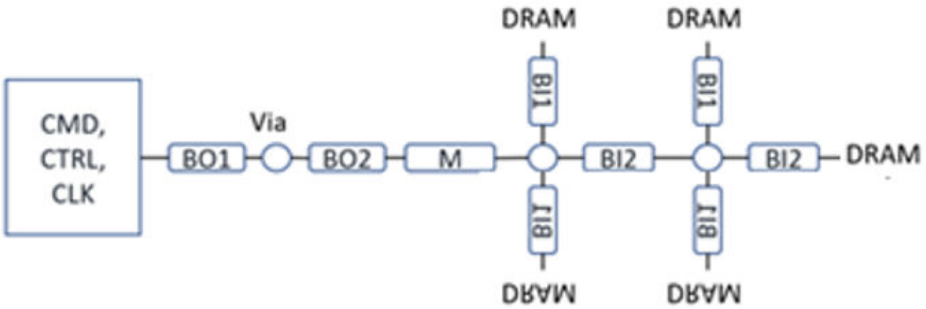


Figure 47. Dual Rank, DRAM x 16, 40-bit Interface (only 8 bits ECC of the last DRAM are used)



The above figures show examples of CA/CTL/CLK Fly-by clamshell routing topology for DDR5 memory down configuration.

Fly-by routing starts with the FPGA, then followed by DRAM chips daisy-chained together. The table in the following topic outlines routing guidelines for the command, control, and clock (CMD/CTRL, CLK) signals.

7.3.7.2. Routing Guidelines for DDR5 Memory Down: 1 Rank or 2 Rank (x8 bit or x16 bit) Configurations

The following table provides recommended trace impedance and length for each of the DDR5 signals based on a memory down topology.

For example, the maximum length of the main trace routing can be derived from total trace length by subtracting the break-out and break-in trace segment lengths routed.

In this table, the signal trace width and minimum spacing/gaps (in mils) from edge-to-edge of signal traces are based on the default stackup shown in the PCB Stack-up and Design Considerations topic; however, PCB designers can use the target impedance for any other stackups. The *h* in the table indicates the minimum substrate height from signal layer to reference layer.

Table 136. Routing Guideline for DDR5 Memory Down: 1 Rank and 2 Rank (x8 bit or x16 bit), total up to 40 bits

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Min Trace Spacing, S1 (mil): Within Group	Min Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Min Trace Spacing, S3 (mil): DQ Nibble to Nibble	Min Trace Spacing (mil), Within DIFF pair	Min Trace Spacing (mil), DQS pair to DQ	Min Trace Spacing (mil), CLK pair to CMD/CTRL/CKE
			Segment	Total MB								
CLK	BO1	US	500	4000 to first DRAM, 6600 to the last DRAM		4	5, 17	17		4		17
	BO2	SL	1000			4	5, 17	17		4		17
	M	SL	2390		40	5.2		9 (3h)		4		9 (3h)
	BI1	US	862			3		9 (3h)		4		9 (3h)
	BI2	SL	648		50+	3		9 (3h)		4		9 (3h)
CMD/ADDR/CTRL	BO1	US	500	4000 to first DRAM, 6600 to the last DRAM		4	5, 17	17				
	BO2	SL	1000			4	5, 17	17				
	M	SL	2390		40	5.2	6 (2h)	9 (3h)				
	BI1	US	862			3	6 (2h)	9 (3h)				
	BI2	SL	648		50+	3	6 (2h)	9 (3h)				
DQ	BO1	US	50	4500		4	5, 17		17			
	BO2	SL	1000			4	5, 17		17			
	M	SL			45	4.2	6 (2h)		9 (3h)			
	BI	US	50			4	6 (2h)		9 (3h)			
DQS	BO1	US	50	4500		4				4	17	
	BO2	SL	1000			4				4	17	
	M	SL			45	4.2				4	9 (3h)	
	BI	US	50			4				4	9 (3h)	

Memory Down Topology Guidelines	
Reference plane	Continuous Ground Only
Use 3x of Dielectric Height for serpentine routing spacing	

Reset signal routing also follows the CMD/ADD/CTRL routing design. Maintain an edge-to-edge space between the Reset signal and other signals on the same layer of at least 5x h. There is no requirement to have skew matching between the Reset signal and CLK signal.

The following table provides a detailed skew matching guideline. Ensure that you include both PCB physical routing skew and package routing skew in your skew matching criteria. The physical length matching criteria in the table reflects the default stackup in our platform PCB design.

Table 137. Skew Matching Requirement for DDR5 Memory Down, 1 Rank x8 bit Configuration

Length Matching		
Length Matching Rules	**Length**	**Time (assuming 170ps/in delay)**
Length Matching between DQS and CLK	-1500mil < CLK - DQS < 2500mil	-255ps < CLK - DQS < 425ps
Length Matching between DQ and DQS within	-20mil < DQ - DQS < 20mil	-3.5ps < DQ - DQS < 3.5ps
Length matching between DQS and DQS#	< 5mil	< 1ps
Length matching between CLK and CLK#	< 5mil	< 1ps
Length matching between CLK0 and CLK1	< 40mil	< 7ps
Length matching between CS and Clock	0 < CLK - CS < 100mil	0 < CLK - CS < 17ps
Length matching among CS within each channel	< 100mil	< 17ps
Length matching between CMD/ADDR and Clock	-100mil < CLK - CMD/ADDRESS < 100mil	-17ps < CLK - CMD/ADDR < 17ps
Length matching among CMD/ADDR within each	< 100mil	< 17ps
Include package length in length matching	Required	Required
Notes		
Keep GND stitching via within 80mil from signal transition which changes reference planes.		
Use 3x of Dielectric Height for serpentine routing spacing		

The maximum data rate depends on the memory configuration as shown in the following table.

Board thickness in the design can vary from thin to thick. In this design guideline, a standard thin board is 65 mil and 120 mil is a standard thick board.

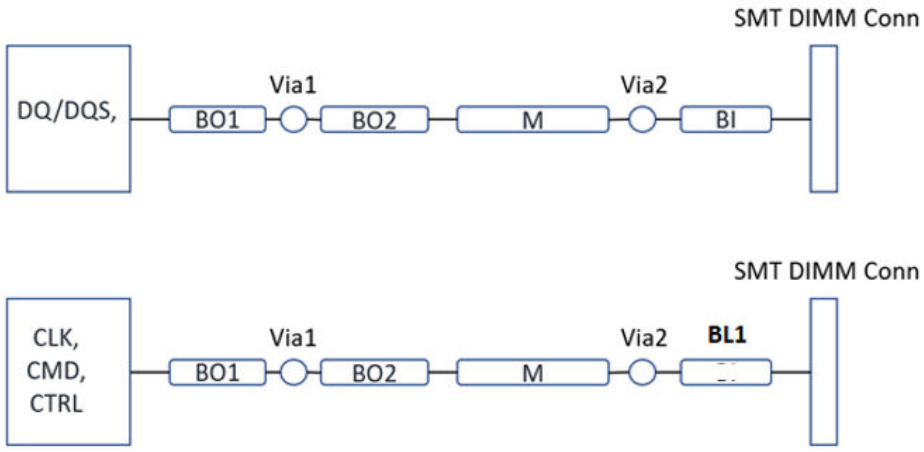
Table 138. DQ Routing Summary for DDR5 Memory Down

Memory Interface	DDR5	
Signal Group	DQ	
Board Thickness (mil)	65 mil or HDI or 120 mil (Data Routing must be on upper layers to avoid long via/vertical Xtalk)	65 mil or HDI or 120 mil (Data Routing must be on upper layers to avoid long via/vertical Xtalk)
PCB Stripline Trace Impedance (ohms)	45	45
Memory Configuration	Memory Down	Memory Down
# of Rank	1 (40bits total), x8 or x16	2 (40 bits total), x8 or x16
Maximum PCB Length Total (inches) to the DRAM	4.5	4.5
Note	Maximum package length in FPGA design is shorter than 34mm	

7.3.7.3. Routing Guidelines for DDR5 RDIMM, UDIMM, and SODIMM Configurations

The following figure shows DDR5 1 x DIMM per channel topology. A maximum of two transition vias are allowed for high performance.

Figure 48. DDR5 RDIMM, UDIMM, and SODIMM, 1 DIMM Per Channel Topology (1 Rank or 2 rank, x8 or x16 Bits, up to 40 Bits interface)



The following table shows physical trace segment routing guidelines, including the target impedance of routing for each signal and the minimum space between signal traces on the same layer. The column trace widths (in mil) and minimum space between traces (in mil) are based on a Intel board design stackup; however, the PCB designer must meet the impedance target criteria. The *h* in the routing guideline stands for the minimum substrate height between the signal layer and reference plane. Ensure that you follow the trace-to-trace-edge gap/space criteria in the guideline, based on "h" in various stackups.

Table 139. Routing Guidelines for DDR5 UDIMM, RDIMM, and SODIMM, 1 DIMM per Channel (1 Rank or 2 Rank)

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil): DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/CTRL/CKE	Channel to Channel spacing (DQ to DQ between two channels)
			Segment	Total MB									
DQ	BO1	US	50	4500		3.5	5, 17		17				17
	BO2	SL	1000			3.5	5, 17		17				17
	M	SL			50	3.5	6 (2h)		9 (3h)				12 (4h)
	BI	US	50			3.5	6 (2h)		9 (3h)				12 (4h)
DQS	BO1	US	50	4500		3.5	5, 17			4	17		
	BO2	SL	1000			3.5	5, 17			4	17		
	M	SL			50	3.5				4	9 (3h)		
	BI	US	50			3.5							
CLK	BO1	US	500	4500		3.5	5, 17		17				17
	BO2	SL	1000			3.5	5, 17		17				17
	M	SL			40	5.2	6 (2h)		9 (3h)				12 (4h)
	BI1	US	862			3.5	6 (2h)		9 (3h)				12 (4h)
CMD/ADDR/CTRL	BO1	US	500	4500		3.5	5, 17			4	17		
	BO2	SL	1000			3.5	5, 17			4	17		
	M	SL			40	5.2				4	9 (3h)		
	BI1	US	862			3.5							

DDR4 SODIMM 1DPC Topology Guidelines	
Number of vias allowed	Max 2 vias, including PTH connector
Reference plane	Continuous Ground Only
Use 3x of Dielectric Height for serpentine routing spacing	

Reset signal routing design also follows the command, address, and control (CMD/ADD/CTRL) routing design. Maintain the space from the Reset signal to other signals on the same layer (edge to edge) at least 5x h. There is no requirement to have skew matching between Reset and CLK signals.

Skew matching for a DDR interface consists of both package routing skew and PCB physical routing skew. You must maintain skew matching of CA and CTRL with respect to the clock signals to ensure that signals at the receiver are correctly sampled. There is also a skew matching requirement for DQ and DQS within a byte group, DQS and CLK.

The following table provides a detailed skew matching guideline to facilitate PCB trace routing. The length matching criteria in this table represents a default PCB on an Intel platform board design. Skew matching criteria must be always followed in any other stackup.

Table 140. Skew Matching Requirements for DDR5 UDIMM, RDIMM, and SODIMM, 1 DIMM per Channel. Length Matching Criteria in this Table Represents a Default Stackup.

Length Matching		
Length Matching Rules	**Length**	**Time (assuming 170ps/in delay)**
Length Matching between DQS and CLK	-1500mil < CLK - DQS < 2500mil	-255ps < CLK - DQS < 425ps
Length Matching between DQ and DQS within byte	-20mil < DQ - DQS < 20mil	-3.5ps < DQ - DQS < 3.5ps
Length matching between DQS and DQS#	< 5mil	< 1ps
Length matching between CLK and CLK#	< 5mil	< 1ps
Length matching between CLK0 and CLK1	< 40mil	< 7ps
Length matching between CS and Clock	0 < CLK - CS < 100mil	0 < CLK - CS < 17ps
Length matching among CS within each channel	< 100mil	< 17ps
Length matching between CMD/ADDR and Clock	-100mil < CLK - CMD/ADDRESS < 100mil	-17ps < CLK - CMD/ADDR < 17ps
Length matching among CMD/ADDR within each channel	< 100mil	< 17ps
Include package length in length matching	Required	Required
Notes		
Keep GND stitching via within 80mil from signal transition which changes reference planes.		
Use 3x of Dielectric Height for serpentine routing spacing		

The maximum DDR5 data rate depends on the configuration and PCB in the following table. For information on the maximum supported data rate, refer to the appropriate Agilex 7 FPGA device data sheet.

Board thickness in the design can vary from thin to thick. The 65 mil is a standard thin and 120 mil is a standard thick PCB in this design guideline.

Table 141. DQ Routing Summary for DDR5 DIMM

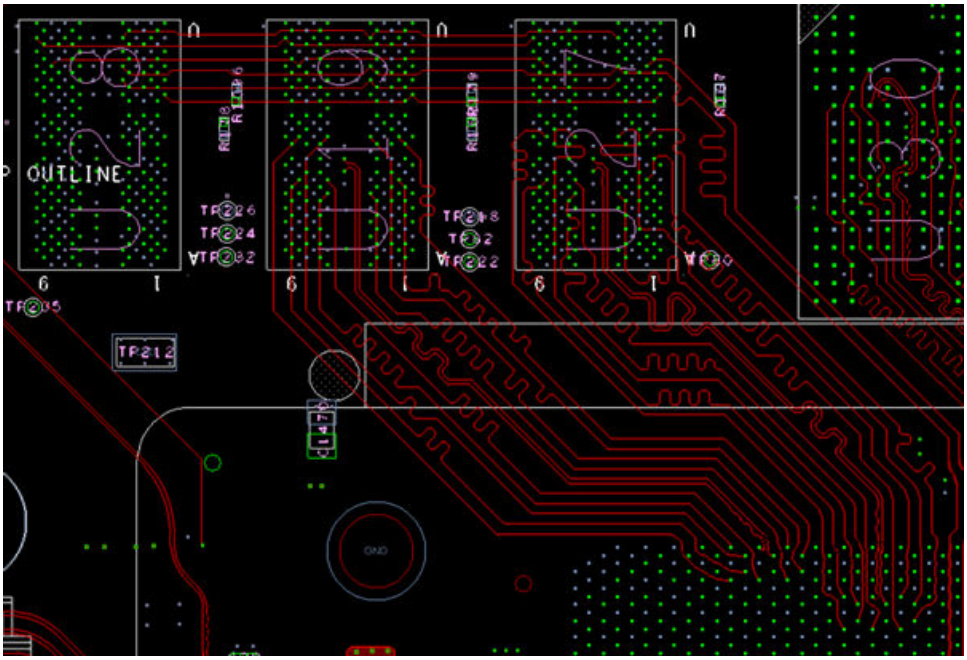
Memory Interface	DDR5 UDIMM	DDR5 RDIMM	DDR5 SODIMM
Signal group	**DQ**		
Board Thickness (Thin or Thick)	65 mil or HDI or 120mil (Data Routing must be on upper layers to avoid long via/vertical Xtalk)	65 mil or HDI or 120mil (Data Routing must be on upper layers to avoid long via/vertical Xtalk)	65mil or HDI or 120mil (Data Routing must be on upper layers to avoid long via/vertical Xtalk)
PCB Stripline Trace Impedance (ohms)	50	50	50
Memory Configuration	1 xDIMM per Channel (40 Bits total), x8 or x16	1x DIMM per Channel (40 Bits total), x8 or x16	1x DIMM per Channel (40 bits total), x8 or x16
# of Rank	2 per DIMM	2 per DIMM	2 per DIMM
Max. Length Total (Inch)	4.5	4.5	4.5
Notes	Max. package length in FPGA design is shorter than 34mm.		

7.3.7.4. Example of a DDR5 layout on Intel FPGA Platform Board

The following figure shows the layout example of a DDR5 x32 plus 8x ECC (3x DRAM Dual-Die Memory Down).

This layout is designed on a thick PCB (120mil stackup) using micro vias and through vias with backdrill. The DDR5 Data signal routing is on upper layers to avoid vertical crosstalk and achieve high performance; the CS/CTRL signals can be routed on deeper layers.

Figure 50. DDR5 32bits+ 8x bits ECC (3 x DRAMs Dual Die) PCB routing on an Intel FPGA Platform Board with thick stackup



7.3.8. DDR5 Simulation Strategy

The simulation strategy has two parts:

- Data Signal signal integrity simulation with respect to their DQS on the worse signal integrity of a data group (considering the longest routing and maximum vertical crosstalk between signals).
- CS/CTRL/CMD signal integrity simulation with respect to their CLK signals on the worst signal integrity of those signals (considering the longest routing and maximum vertical crosstalk between signals).

Intel recommends that a signal integrity engineer reviews the layout and picks the worst data group (select a victim and surrounded aggressors and DQS in the group) that has the worst signal integrity on the layout (that is, the worst cross talk coupling between deep vertical vias), long trace/PCB routing and maximum reflection on routing path due to long via stubs if backdrilling is not applied.

Designers must perform signal integrity simulation of the board layout for the selected victim surrounded by aggressor signals.

Ensure that the channel analysis is performed in the time domain (using PRBS pattern for I/O signal generator) while the channel is built, by using the actual per-pin package model at both ends, and PCB model in the format of scattering parameter along with I/O buffer model at both ends. DDR5 requires an IBIS AMI buffer model (due to equalizations/FFE/DFE at both TX and RX) at both ends to recover the data. Evaluate the eye diagram after the simulation to ensure that the design meets eye specification at both ends.

Note: Currently the FPGA DDR5 GPIO-B buffer IBIS AMI model is not available for designers to do the signal integrity simulation. Intel recommends that designers strictly follow the PCB routing design guidelines in this chapter, to achieve the maximum supported data rate for the selected configuration.

8. Agilex 7 M-Series FPGA EMIF IP – LPDDR5 Support

This chapter contains IP pin planning information for Agilex 7 M-Series FPGA external memory interface IP for LPDDR5.

8.1. Agilex 7 M-Series FPGA EMIF IP Parameters for LPDDR5

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP.

Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

8.1.1. Agilex 7 FPGA EMIF IP Parameter for LPDDR5

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 142. Group: General IP Parameters / High-Level Parameters

Display Name	Description
Technology Generation	Denotes the specific memory technology generation to be used Note: This parameter can be auto-computed. (Identifier: MEM_TECHNOLOGY)
Memory Format	Specifies the packaging format of the memory device (Identifier: MEM_FORMAT)
Memory Device Topology	Topology used by memory device (Identifier: MEM_TOPOLOGY)
Memory Ranks	Total number of physical ranks in the interface (Identifier: MEM_NUM_RANKS)
Number of Channels	Number of channels (Identifier: MEM_NUM_CHANNELS)
Device DQ Width	If the device is a DIMM: Specifies the full DQ width of the DIMM. If the interface is composed of discrete components: Specifies the DQ width of each discrete component. (Identifier: MEM_DEVICE_DQ_WIDTH)
Number of Components Per Rank	Number of components per rank. If each component contains more than one rank, then set this parameter to 1. (Identifier: MEM_COMPS_PER_RANK)

continued...

Display Name	Description
Enable Frequency Set Point 1	Specifies whether or not a second Frequency Set Point will be used for FSP-enabled technologies (Identifier: PHY_FSP1_EN)
Enable Frequency Set Point 2	Specifies whether or not a third Frequency Set Point will be used for FSP-enabled technologies (Identifier: PHY_FSP2_EN)
ECC Mode	Specifies the type of ECC (if any) and the required number of side-band bits per channel that will be used by this EMIF instance. While not all required side-band bits necessarily carry ECC bits, all need to be connected to the memory device. If enabling ECC requires more side-band bits than necessary ECC bits, then ECC bits are transmitted on the least significant side-band bits. Note: This parameter can be auto-computed. (Identifier: CTRL_ECC_MODE)
Total DQ Width	(Derived Parameter) This will be the width (in bits) of the mem_dq port on the memory interface. For a component interface, it is calculated based on: (MEM_COMPS_PER_RANK * MEM_DEVICE_DQ_WIDTH + (8 bits if Side-band ECC is enabled, or 8 bits if AXI4 User Data is enabled in fabric modes, or 4 bits if AXI4 User Data is enabled in NoC mode)) * MEM_NUM_CHANNELS For a DIMM-based interface, it is just MEM_DEVICE_DQ_WIDTH + (8 bits if side-band ECC is enabled, or 8 bits if AXI4 User Data is enabled in fabric modes, or 4 bits if AXI4 User Data is enabled in NoC mode) * MEM_NUM_CHANNELS. (Identifier: MEM_TOTAL_DQ_WIDTH)
Memory Clock Frequency for Frequency Set Point 0	Specifies the **FSP0 operating frequency** of the memory interface in MHz. If you change the memory frequency, you should update the memory latency parameters on the **Memory** tab and the memory timing parameters on the **Mem Timing** tab. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FSP0_FREQ_MHZ)
Memory Clock Frequency for Frequency Set Point 1	Specifies the **FSP1 operating frequency** of the memory interface in MHz. If you change the memory frequency, you should update the memory latency parameters on the **Memory** tab and the memory timing parameters on the **Mem Timing** tab. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FSP1_FREQ_MHZ)
Memory Clock Frequency for Frequency Set Point 2	Specifies the **FSP2 operating frequency** of the memory interface in MHz. If you change the memory frequency, you should update the memory latency parameters on the **Memory** tab and the memory timing parameters on the **Mem Timing** tab. Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FSP2_FREQ_MHZ)
Instance ID	Instance ID of the EMIF IP. EMIF in the same bank, or connected to related user logic (e.g. to the same INIU), should have unique IDs in order to distinguish them when using the side-band interface. Valid values are 0-6. (Identifier: INSTANCE_ID)

Table 143. Group: General IP Parameters / Memory Device Preset Selection

Display Name	Description
Use Memory Device Preset from file for FSP 0	Specifies whether MEM_PRESET_ID_FSP0 will be a value from Quartus (if false), or a value from a custom preset file path (if true)

continued...

Display Name	Description
	(Identifier: MEM_PRESET_FILE_EN_FSP0)
Memory Preset custom file path for FSP 0	Path to a .qprs file on the users disk for Frequency Set Point 0 (Identifier: MEM_PRESET_FILE_QPRS_FSP0)
Use Memory Device Preset from file for FSP 1	Specifies whether MEM_PRESET_ID_FSP1 will be a value from Quartus (if false), or a value from a custom preset file path (if true) (Identifier: MEM_PRESET_FILE_EN_FSP1)
Memory Preset custom file path for FSP 1	Path to a .qprs file on the users disk for Frequency Set Point 1, if enabled (Identifier: MEM_PRESET_FILE_QPRS_FSP1)
Use Memory Device Preset from file for FSP 2	Specifies whether MEM_PRESET_ID_FSP2 will be a value from Quartus (if false), or a value from a custom preset file path (if true) (Identifier: MEM_PRESET_FILE_EN_FSP2)
Memory Preset custom file path for FSP 2	Path to a .qprs file on the users disk for Frequency Set Point 2, if enabled (Identifier: MEM_PRESET_FILE_QPRS_FSP2)
Memory Preset for FSP 0	The name of a preset that the user would like to load for LPDDR5 Frequency Set Point 0, describing the memory device that this EMIF will be targeting. Note: This parameter can be auto-computed. (Identifier: MEM_PRESET_ID_FSP0)
Memory Preset for FSP 1	The name of a preset that the user would like to load for LPDDR5 Frequency Set Point 1, describing the memory device that this EMIF will be targeting. Note: This parameter can be auto-computed. (Identifier: MEM_PRESET_ID_FSP1)
Memory Preset for FSP 2	The name of a preset that the user would like to load for LPDDR5 Frequency Set Point 2, describing the memory device that this EMIF will be targeting. Note: This parameter can be auto-computed. (Identifier: MEM_PRESET_ID_FSP2)

Table 144. Group: General IP Parameters / Advanced Parameters / PHY / Topology

Display Name	Description
Use NOC	Specifies whether we are using the NOC or bypassing it Note: This parameter can be auto-computed. (Identifier: PHY_NOC_EN)
Asynchronous Enable	Specifies whether the user logic is clocked based on the clock provided by the IP (Sync), or by a separate user clock (Async). If true - async mode is used, if false - sync mode is used. (Identifier: PHY_ASYNC_EN)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms IO BANK and TOP vs BOT part of the IO BANK). Legal ranges are derived from device floorplan. By default (value=AUTO), the most optimal location is selected (to maximize available frequency and data width). Note: This parameter can be auto-computed. (Identifier: PHY_AC_PLACEMENT)
PLL Reference Clock Frequency	Specifies what PLL reference clock frequency the user will supply. It is recommended to use the fastest possible PLL reference clock frequency because it leads to better jitter performance. Note: This parameter can be auto-computed. (Identifier: PHY_REFCLK_FREQ_MHZ)

Table 145. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings

Display Name	Description
Voltage	The voltage level for the I/O pins driving the signals between the memory device and the FPGA memory interface. (Identifier: PHY_IO_VOLTAGE)

Table 146. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Address/Command

Display Name	Description
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the Address/Command Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_AC_X_R_S_AC_OUTPUT_OHM)

Table 147. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Memory Clock

Display Name	Description
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the CK Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_CLK_X_R_S_CK_OUTPUT_OHM)

Table 148. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Data Bus

Display Name	Description
I/O Standard	Specifies the I/O electrical standard for the data bus pins. The selected I/O standard configures the circuit within the I/O buffer to match the industry standard. (Identifier: GRP_PHY_DATA_X_DQ_IO_STD_TYPE)
Drive Strength	This parameter allows you to change the output on chip termination settings for the selected I/O standard on the DQ Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_DATA_X_R_S_DQ_OUTPUT_OHM)
Slew Rate	Specifies the slew rate of the data bus pins. The slew rate (or edge rate) describes how quickly the signal can transition, measured in voltage per unit time. *Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.* (Identifier: GRP_PHY_DATA_X_DQ_SLEW_RATE)
Input Termination	This parameter allows you to change the input on chip termination settings for the selected I/O standard on the DQ Pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_DATA_X_R_T_DQ_INPUT_OHM)
Initial Vrefin	Specifies the **initial value for the reference voltage on the data pins(Vrefin)**. The specified value serves as a starting point and may be overridden by calibration to provide better timing margins. (Identifier: GRP_PHY_DATA_X_DQ_VREF)

Table 149. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / PHY Inputs

Display Name	Description
PLL Reference Clock Input Termination	This parameter allows you to change the input on chip termination settings for the selected I/O standard on the refclk input pins. Perform board simulation with IBIS models to determine the best settings for your design. (Identifier: GRP_PHY_IN_X_R_T_REFCLK_INPUT_OHM)

Table 150. Group: General IP Parameters / Advanced Parameters / FPGA I/O / FPGA I/O Settings / Decision Feedback Equalization (DFE)

Display Name	Description
DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_1)
DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_2)
DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_3)
DFE Tap 4	This parameter allows you to select the amount of bias used on tap 4 of the FPGA DFE (Identifier: GRP_PHY_DFE_X_TAP_4)

Table 151. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Data Bus On-Die Termination (ODT)

Display Name	Description
Target Write Termination	Specifies the target termination to be used during a write (Identifier: GRP_MEM_ODT_DQ_X_TGT_WR)
Non-Target Termination	Specifies the termination to be used for the non-target rank in a multi-rank configuration (Identifier: GRP_MEM_ODT_DQ_X_NON_TGT)
Drive Strength	Specifies the termination to be used when driving read data from memory (Identifier: GRP_MEM_ODT_DQ_X_RON)
Data Clock Termination	Specifies the termination to be used for the data clock (WCK) (Identifier: GRP_MEM_ODT_DQ_X_WCK)

Table 152. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Data Bus Reference Voltage (Vref)

Display Name	Description
VrefDQ Value	Specifies the initial VrefDQ value to be used (Identifier: GRP_MEM_DQ_VREF_X_VALUE)

Table 153. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Address/Command Bus On-Die Termination (ODT)

Display Name	Description
Common Termination	Common termination value that can be applied to CA/CK/CS for LPDDR4 and can be applied to CA/CK for LPDDR5

continued...

Display Name	Description
	(Identifier: GRP_MEM_ODT_CA_X_CA_COMM)
CA Termination Enable	Enable the common termination value on the CA bus. For LPDDR4, enabling CA termination will have no effect unless the ODT_CA bond pad is HIGH. (Identifier: GRP_MEM_ODT_CA_X_CA_ENABLE)
CS Termination Enable	Enable the common termination value on the CS bus for LPDDR4. For LPDDR5, this enables the fixed-value 80 Ohm (RZQ/3) CS termination if it is supported by the memory. (Identifier: GRP_MEM_ODT_CA_X_CS_ENABLE)
CK Termination Enable	Enable the common termination value on the CK bus (Identifier: GRP_MEM_ODT_CA_X_CK_ENABLE)

Table 154. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Address/Command Bus Reference Voltage (Vref)

Display Name	Description
VrefCA Value	Specifies the initial VrefCA value to be used (Identifier: GRP_MEM_VREF_CA_X_CA_VALUE)

Table 155. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Decision Feedback Equalization (DFE)

Display Name	Description
DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the memory DFE (Identifier: GRP_MEM_DFE_X_TAP_1)

Table 156. Group: General IP Parameters / Advanced Parameters / AXI Settings / AXI Interface Settings

Display Name	Description
Enable Debug Tools	If enabled, the AXI-L port will be connected to SLD nodes, allowing for a system-console avalon manager interface to interact with this AXI-L subordinate interface. (Identifier: DEBUG_TOOLS_EN)
AXI-Lite Port Access Mode	Specifies whether the AXI-Lite port is connected to the fabric, the NOC, or disabled Note: This parameter can be auto-computed. (Identifier: AXI_SIDEBAND_ACCESS_MODE)

Table 157. Group: General IP Parameters / Advanced Parameters / Additional Parameters / Additional String Parameters

Display Name	Description
User Extra Parameters	Semi-colon separated list of key/value pairs of extra parameters (Identifier: USER_EXTRA_PARAMETERS)

Table 158. Group: Example Design / Example Design

Display Name	Description
HDL Selection	This option lets you choose the format of HDL in which generated simulation and synthesis files are created. You can select either Verilog or VHDL.

continued...

Display Name	Description
	(Identifier: EX_DESIGN_HDL_FORMAT)
Synthesis	Generate Synthesis Example Design (Identifier: EX_DESIGN_GEN_SYNTH)
Simulation	Generate Simulation Example Design (Identifier: EX_DESIGN_GEN_SIM)
Core Clock Freq	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode) Note: This parameter can be auto-computed. (Identifier: EX_DESIGN_CORE_CLK_FREQ_MHZ)
Core Refclk Freq	PLL reference clock frequency in MHz for PLL supplying the core clock (Identifier: EX_DESIGN_CORE_REFCLK_FREQ_MHZ)
NOC Refclk Freq	NOC Refclk Freq for the NOC control IP Note: This parameter can be auto-computed. (Identifier: EX_DESIGN_NOC_REFCLK_FREQ_MHZ)
Traffic Generator Remote Access	Specifies whether the traffic generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled (Identifier: EX_DESIGN_HYDRA_REMOTE)

Table 159. Group: Example Design / Performance Monitor

Display Name	Description
Enable performance monitoring	Enable performance monitor on all channels for measuring read/write transaction metrics (Identifier: EX_DESIGN_PMON_ENABLED)

Table 160. Group: Example Design / Traffic Generator Program

Display Name	Description
Traffic Generator Program	Specifies the traffic pattern to run (Identifier: EX_DESIGN_HYDRA_PROG)

8.1.2. Agilex 7 FPGA EMIF Memory Device Description IP (LPDDR5) Parameter Descriptions

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 161. Group: Configuration Save

Display Name	Description
Configuration Filepath	Filepath to Save to (.qprs extension) (Identifier: MEM_CONFIG_FILE_QPRS)

Table 162. Group: High-Level Parameters

Display Name	Description
Memory Format	Specifies the packaging format of the memory device

continued...

Display Name	Description
	(Identifier: MEM_FORMAT)
Memory WCK Frequency for this FSP	Specifies the Write Clock Frequency for this Frequency Set Point Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FQD_FREQ_MHZ)
Number of Channels	Number of Channels. This value must be consistent with the number of channels parameter in the top-level EMIF IP GUI. (Identifier: MEM_NUM_CHANNELS)
Memory Ranks	Total number of physical ranks in the interface. This value must be consistent with the memory ranks parameter in the top-level EMIF IP GUI. (Identifier: MEM_NUM_RANKS)
Number of Components Per Rank	Number of Components Per Rank. If each component contains more than one rank, then set this parameter to 1. (Identifier: MEM_COMPS_PER_RANK)
Density of Each Memory Die	Specifies the density of the memory die in Gb (Identifier: LPDDR5_MEM_DEVICE_DENSITY_GBITS)
Enable Write Data Bus Inversion	Enables Write Data Bus Inversion (Identifier: LPDDR5_MEM_DEVICE_WR_DBI_EN)
Enable Data Mask	Enables Data Masking for write operations (Identifier: LPDDR5_MEM_DEVICE_DM_EN)

Table 163. Group: Memory Interface Parameters / Data Bus

Display Name	Description
DQ Width per DRAM Component	Specifies the DQ width of each LPDDR5 DRAM component. As byte mode is not supported, this value is always 16. To form x32 LP5 interfaces, select 2 components per rank at the EMIF IP level. (Identifier: LPDDR5_MEM_DEVICE_DQ_WIDTH)
Total DQ Width Per Channel	Total DQ Width Per Channel. For LPDDR5 packages, this is the product of the per-DRAM DQ Width and Number of Individual DRAM Components per Rank. (Identifier: LPDDR5_MEM_DEVICE_TOTAL_DQ_WIDTH_PER_CHANNEL)
Burst Length	Burst Length (Identifier: LPDDR5_MEM_DEVICE_BURST_LENGTH)

Table 164. Group: Memory Interface Parameters / Device Topology

Display Name	Description
Device Row Address Width	Specifies the row address width of this LPDDR5 DRAM component. This value is auto-derived from the specified component density. (Identifier: LPDDR5_MEM_DEVICE_ROW_ADDR_WIDTH)
Device Maximum Bank Address Width	Specifies the maximum bank address width. This value is fixed as per the JEDEC standard and cannot be changed. (Identifier: LPDDR5_MEM_DEVICE_MAX_BA_WIDTH)
Device Maximum Bank Group Address Width	Specifies the maximum bank group address width. This value is fixed as per the JEDEC standard and cannot be changed. (Identifier: LPDDR5_MEM_DEVICE_MAX_BG_WIDTH)
Device Column Address Width	Specifies the column address width of this LPDDR5 DRAM component. This value is fixed for all component densities as per the JEDEC standard and cannot be changed.

continued...

Display Name	Description
	(Identifier: LPDDR5_MEM_DEVICE_COL_ADDR_WIDTH)
Device Burst Address Width	Specifies the burst address width. This value is fixed as per the JEDEC standard and cannot be changed. (Identifier: LPDDR5_MEM_DEVICE_BURST_ADDR_WIDTH)

Table 165. Group: Memory Timing Parameters / Timing Parameters

Display Name	Description
Memory Speedbin	Maximum Data Rate for which this memory device is rated for (Identifier: LPDDR5_MEM_DEVICE_SPEEDBIN)
Memory Write Latency Set	Selects the Write Latency Set for this device. Selection affects auto-calculation of Write Latency. (Identifier: LPDDR5_MEM_DEVICE_WLS)
Memory Read Latency	Read Latency of the memory device for this Frequency Set Point in clock cycles Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_CL_CYC_FQD)
Memory Write Latency	Write Latency of the memory device for this Frequency Set Point in clock cycles Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_CWL_CYC_FQD)

Table 166. Group: Memory Timing Parameters / Pre- and Post-Amble Options

Display Name	Description
Read Postamble Mode	RDQS Postamble Mode for this Frequency Set Point (Identifier: LPDDR5_MEM_DEVICE_RDQS_PST_MODE_FQD)
Read Preamble Cycles	RDQS Preamble Length (Identifier: LPDDR5_MEM_DEVICE_RD_PREAMBLE_CYC)
Read Postamble Cycles	RDQS Postamble Length (Identifier: LPDDR5_MEM_DEVICE_RD_POSTAMBLE_CYC)
Write Postamble Cycles	WCK Postamble Length (Identifier: LPDDR5_MEM_DEVICE_WR_POSTAMBLE_CYC)

Table 167. Group: Memory Timing Parameters / Advanced Timing Parameters

Display Name	Description
Min Number of Refs Reqd	Minimum Number of Refreshes Required Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_MINNUMREFSREQ)
tRCD	RAS-to-CAS Delay in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRCD_NS)
tRPab	All-Bank Precharge Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRPAB_NS)
tRPpb	Per-Bank Precharge Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRPPB_NS)

continued...

Display Name	Description
tRAS	Row Active Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRAS_NS)
tRAS_Max	Maximum Row Active Time in us Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRAS_MAX_US)
tWR	Write Recovery Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TWR_NS)
tRRD_L	RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Long) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRRD_L_NS)
tRRD_S	RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Short) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRRD_S_NS)
tFAW	Four-bank ACTIVE window time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TFAW_NS)
tRBTP	Read Burst End to Precharge Command Delay in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRBTP_NS)
tWTR_S	Write-to-Read Delay (Short) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TWTR_S_NS)
tWTR_L	Write-to-Read Delay (Long) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TWTR_L_NS)
tPPD	Precharge-to-Precharge Delay Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TPPD_NS)
tRC	Activate-to-Activate command period (same bank) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRC_NS)
tZQLAT	ZQCAL Latch Quiet Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TZQLAT_NS)
tPW_RESET	Min RESET_n low time for Reset Initialization with Stable Power Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TPW_RESET_NS)
tERQE	Enhanced RDQS Toggle Mode Entry Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TERQE_NS)
tERQX	Enhanced RDQS Toggle Mode Exit Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TERQX_NS)
tRDQE_OD	ODT-disable from Enhanced RDQS Toggle Mode Entry Time in ns Note: This parameter can be auto-computed.

continued...

Display Name	Description
	(Identifier: LPDDR5_MEM_DEVICE_TRDQE_OD_NS)
tRDQX_OD	ODT-enable from Enhanced RDQS Toggle Mode Exit Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRDQX_OD_NS)
tRDQSTFE	Read/Write-based RDQS_t Training Mode Entry Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRDQSTFE_NS)
tRDQSTFX	Read/Write-based RDQS_t Training Mode Exit Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRDQSTFX_NS)
tCCDMW	CAS-to-CAS Delay for Masked Write in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TCCDMW_NS)
tREFW	Refresh Window in ms Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TREFW_MS)
tREFI	Refresh Interval Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TREFI_NS)
tRFCab	All-Bank Refresh Cycle Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRFCAB_NS)
tRFCpb	Per-Bank Refresh Cycle Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TRFCPB_NS)
tpbR2pbR	Per-Bank Refresh to Per-Bank Refresh minimum interval time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TPBR2PBR_NS)
tpbR2ACT	Per-Bank Refresh to Activate minimum interval time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TPBR2ACT_NS)
tCKCSH	Valid Clock Requirement before CS goes High (Power-Down AC Timings) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TCKCSH_NS)
tCMDPD	Delay from valid command to Power Down Entry in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TCMDPD_NS)
tXP	Exit Power-Down to next valid command Delay in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TXP_NS)
tCSH	Minimum CS High Pulse Width at Power Down Exit in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TCSH_NS)
tCSLCK	Valid Clock Requirement after Power Down Entry in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TCSLCK_NS)

continued...

Display Name	Description
tCSPD	Delay time from Power Down Entry to CS going High in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TCSPD_NS)
tMRWPD	Delay from MRW Command to Power Down Entry in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TMRWPD_NS)
tZQPD	Delay from ZQ Calibration Start/Latch Command to Power Down Entry in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TZQPD_NS)
tESPD	Delay time from Self-Refresh Entry command to Power Down Entry command in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TESPD_NS)
tSR	Minimum Self-Refresh Time (Entry to Exit) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TSR_NS)
tXSR	Exit Self-Refresh time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TXSR_NS)
tMRR	Mode Register Read Command Period in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TMRR_NS)
tMRW	Mode Register Write Command Period in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TMRW_NS)
tMRD	Mode Register Set Command Delay in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TMRD_NS)
tOSCO	Delay time from Stop WCK2DQI/WCK2DQO Interval Oscillator Command to Mode Register Readout time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TOSCO_NS)
tDQSCK_MAX	Maximum additional delay needed for tDQSCK in Picoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TDQSCK_MAX_PS)
tDQSCK_MIN	Minimum additional delay needed for tDQSCK in Picoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR5_MEM_DEVICE_TDQSCK_MIN_PS)

8.1.3. Agilex 7 FPGA EMIF Calibration IP Parameter Descriptions

The following topics describe the parameters available on each tab of the IP parameter editor, which you can use to configure your IP. Each parameter with an adjacent checkbox can be auto-computed. The checkbox to the left of the parameter controls whether its value is auto-computed (true) or set manually (false). If there is no checkbox to the left of a parameter, then it must be set manually.

Table 168. Group: High-Level Parameters

Display Name	Description
Unique Instance ID for the Calibration IP	Instance ID (Identifier: INSTANCE_ID)
Calibration IP is part of Bank Adjacent Pair	Calibration IP is part of Bank Adjacent Pair (Identifier: IS_PART_OF_BANK_ADJACENT_PAIR)
Number of Peripheral IPs	Number of Peripheral IPs (EMIFs, PHYLites) to be calibrated (Identifier: NUM_CALBUS_PERIPHS)
Number of Standalone I/O PLLs	Number of Standalone I/O PLLs to calibrate (Identifier: NUM_CALBUS_PLLS)
AXI-L Subordinate Port Mode	AXI-L subordinate port can be disabled, or can be used in one of two modes: directly exported to fabric, or connect to the NoC (i.e. to a TNIU) (Identifier: PORT_S_AXIL_MODE)

8.2. Agilex 7 M-Series FPGA EMIF IP Pin and Resource Planning

The following topics provide guidelines on pin placement for external memory interfaces.

Typically, all external memory interfaces require the following FPGA resources:

- Interface pins
- PLL and clock network
- RZQ pins
- Other FPGA resources—for example, core fabric logic, and debug interfaces

Once all the requirements are known for your external memory interface, you can begin planning your system.

8.2.1. Agilex 7 M-Series FPGA EMIF IP Interface Pins

Any I/O banks that do not support transceiver operations in Agilex 7 M-Series FPGAs support external memory interfaces.

However, RDQS (read data strobe), WCK (write clock),and DQ (data) pins are listed in the device pin tables and are fixed at specific locations in the device. You must adhere to these pin locations to optimize routing, minimize skew, and maximize margins. Always check the pin table for the actual locations of the DQS and DQ pins.

Note: Maximum interface width varies from device to device depending on the number of I/O pins and DQS or DQ groups available. Achievable interface width also depends on the number of address and command pins that the design requires. To ensure adequate PLL, clock, and device routing resources are available, you should always test fit any IP in the Quartus Prime software before PCB sign-off.

8.2.1.1. Estimating Pin Requirements

You should use the Quartus Prime software for final pin fitting.

However, you can estimate whether you have enough pins for your memory interface using the EMIF Device Selector on www.intel.com, or perform the following steps:

1. Determine how many read/write data pins are associated per data strobe or clock pair.
2. Calculate the number of other memory interface pins needed, including any other clocks (write clock or memory system clock), address, command, and RZQ. Refer to the External Memory Interface Pin Table to determine necessary address/command/clock pins based on your desired configuration.
3. Calculate the total number of I/O banks required to implement the memory interface, given that an I/O bank supports up to 96 pins.

Test the proposed pin-outs with the rest of your design in the Quartus Prime software (with the correct I/O standard and OCT connections) before finalizing the pin-outs. There can be interactions between modules that are illegal in the Quartus Prime software that you might not know about unless you compile the design and use the Quartus Prime Pin Planner.

8.2.1.2. DIMM Options

The table and figures below illustrate pin placement and routing recommendation for a single 32-bit channel, and two 16-bit channels, respectively.

Note: You should always consult your memory vendor's data sheet to verify pin placement and routing plans.

Table 169. Pin Options for LPDDR5 x32 and x16

Pins	1CH x32	2CH x16	
Data	32-bit DQ[15:0]_A DQ[15:0]_B	DQ[15:0]_A	DQ[15:0]_B
Data mask	DM[1:0]_A DM[1:0]_B	DM[1:0]_A	DM[1:0]_B
Read data strobe	RDQS[1:0]_t_A RDQS[1:0]_c_A RDQS[1:0]_t_B RDQS[1:0]_t_B	RDQS[1:0]_t_A RDQS[1:0]_c_A	RDQS[1:0]_t_B RDQS[1:0]_c_B
Write clock	WCK[1:0]_t_A WCK[1:0]_c_A WCK[1:0]_t_B WCK[1:0]_c_B		
Command/address	CA[6:0]_A CS0_A CA[6:0]_B CS0_B	CA[6:0]_A CS0_A	CA[6:0]_B CS0_B
Clock	CK_t_A CK_c_A CK_t_B	CK_t_A CK_c_A	CK_t_B CK_c_B

continued...

Pins	1CH x32	2CH x16	
	CK_c_B		
Reset	RESET_n	RESET_n (Resistor jumper to select from mem_0 or mem_1.)	

Figure 51. Pin Options for LPDDR5 2ch x16

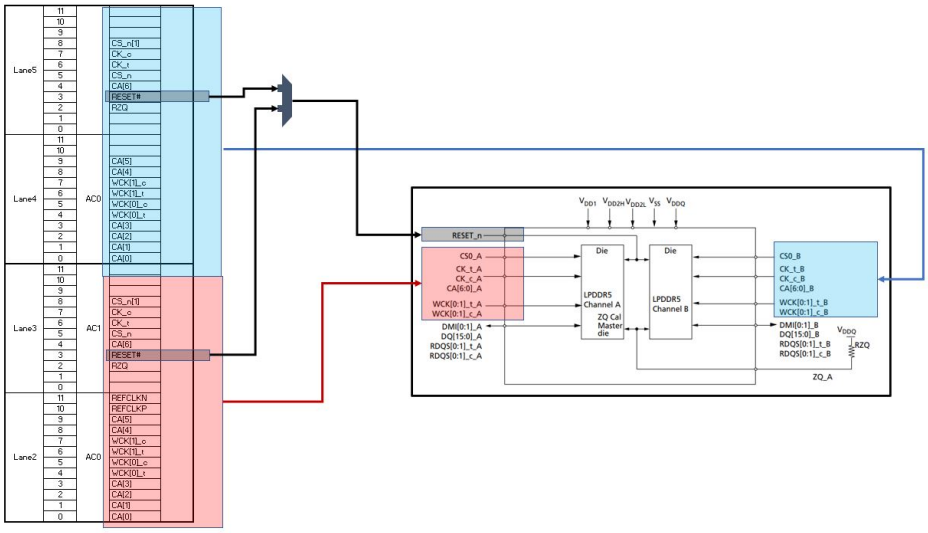
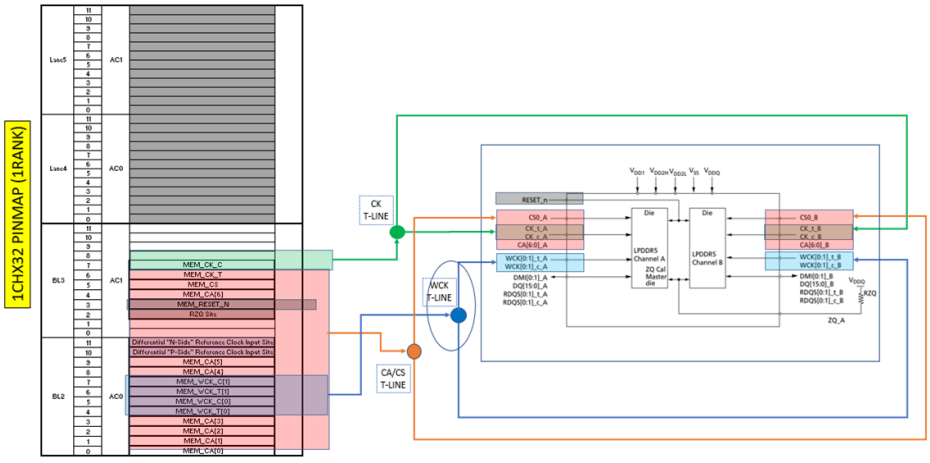


Figure 52. Pin Options for LPDDR5 x32



8.2.1.3. Maximum Number of Interfaces

The maximum number of interfaces supported for a given memory protocol varies, depending on the FPGA in use.

Unless otherwise noted, the calculation for the maximum number of interfaces is based on independent interfaces where the address or command pins are not shared.

Note: You may need to share PLL clock outputs depending on your clock network usage.

For interface information for Agilex 7 M-Series devices, consult the EMIF Device Selector on www.intel.com.

Timing closure depends on device resource and routing utilization. For more information about timing closure, refer to the *Area and Timing Optimization Techniques* chapter in the *Quartus Prime Handbook*.

8.2.2. Agilex 7 M-Series FPGA EMIF IP Resources

The Agilex 7 M-Series FPGA memory interface IP uses several FPGA resources to implement the memory interface.

8.2.2.1. OCT

You require an OCT calibration block if you are using an Agilex 7 M-Series FPGA OCT calibrated series, parallel, or dynamic termination for any I/O in your design. There are two OCT blocks in an I/O bank, one for each sub-bank.

You must observe the following requirements when using OCT blocks:

- The I/O bank where you place the OCT calibration block must use the same V_{CCIO_PIO} voltage as the memory interface.
- The OCT calibration block uses a single fixed R_{ZQ}. You must ensure that an external termination resistor is connected to the correct pin for a given OCT block.

For specific pin connection requirements, refer to *Specific Pin Connection Requirements*.

8.2.2.2. PLL

When using PLL for external memory interfaces, you must consider the following guidelines:

For the clock source, use the clock input pin specifically dedicated to the PLL that you want to use with your external memory interface. The input and output pins are only fully compensated when you use the dedicated PLL clock input pin.

For specific pin connection requirements, refer to *Specific Pin Connection Requirements*.

8.2.3. Pin Guidelines for Agilex 7 M-Series FPGA EMIF IP

The Agilex 7 M-Series FPGA contains I/O banks on the top and bottom edges of the device, which can be used by external memory interfaces.

Agilex 7 M-Series FPGA I/O banks contain 96 I/O pins. Each bank is divided into two sub-banks with 48 I/O pins in each. Sub-banks are further divided into four I/O lanes, where each I/O lane is a group of twelve I/O ports.

The I/O bank, I/O lane, and pairing pin for every physical I/O pin can be uniquely identified by the following naming convention in the device pin table:

- The I/O pins in a bank are represented as P#X#Y#, where:
  - P# represents the pin number in a bank. It ranges from P0 to P95, for 96 pins in a bank.
  - X# represents the bank number on a given edge of the device. X0 is the farthest bank from the zipper.
  - Y# represents the top or bottom edge of the device. Y0 and Y1 refer to the I/O banks on the bottom and top edge, respectively.
- Because an IO96 bank comprises two IO48 sub-banks, all pins with P# value less than 48 (P# <48) belong to the same I/O sub-bank. All other pins belong to the second IO48 sub-bank.
- The *Index Within I/O Bank* value falls within one of the following ranges: 0 to 11, 12 to 23, 24 to 35, or 36 to 47, and represents one of I/O lanes 0, 1, 2, or 3, respectively.
- To determine whether I/O banks are adjacent, you can refer to the sub-bank-ordering figures for your device family in the *Architecture: I/O Bank* topic. In general, you can assume that I/O banks are adjacent within an I/O edge, unless the I/O bank is not bonded out on the package (indicated by the presence of the " - " symbol in the I/O table), or if the I/O bank does not contain 96 pins, indicating that it is only partially bonded out. If an I/O bank is not fully bonded out in a particular device, it cannot be included within the span of sub-banks for a larger external memory interface. In all cases, you should use the Quartus Prime software to verify that your usage can be implemented.
- The pairing pin for an I/O pin is in the same I/O bank. You can identify the pairing pin by adding 1 to its *Index Within I/O Bank* number (if it is an even number), or by subtracting 1 from its *Index Within I/O Bank* number (if it is an odd number).

8.2.3.1. General Guidelines - LPDDR5

You should follow the recommended guidelines when performing pin placement for all external memory interface pins targeting Agilex 7 M-Series devices, whether you are using the hard memory controller or your own solution.

Note: PHY only, RLDRAM*x*, and QDR*x* are not supported with HPS.

Observe the following general guidelines when placing pins for your Agilex 7 M-Series external memory interface:

1. Ensure that the pins of a single external memory interface reside on the same edge I/O.
2. The address and command pins and their associated clock pins in the address and command bank must follow a fixed pin-out scheme, as defined in the table in the *Address and Command Pin Placement for LPDDR5* topic.
3. Not every byte lane can function as an address and command lane or a data lane. The pin assignment must adhere to the LPDDR5 data width mapping defined in *LPDDR5 Data Width Mapping*.
4. A byte lane must not be used by both address and command pins and data pins.

5. An external memory interface can occupy one or more banks on the same edge. When an interface must occupy multiple banks, ensure that those banks are adjacent to one another.
   - If an I/O bank is shared between two interfaces—meaning that two sub-banks belong to two different EMIF interfaces—then both the interfaces must share the same voltage.
   - Sharing of I/O lanes within a sub-bank for two different EMIF interfaces is not permitted; I/O lanes within a sub-bank can be assigned to one EMIF interface only.
6. Any pin in the same bank that is not used by an external memory interface may not be available for use as a general purpose I/O pin:
   - For fabric EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same sub-bank, pins in an I/O lane that is not assigned to an EMIF interface, can be used as general-purpose I/O pins.
   - For HPS EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same bank, pins in an I/O lane that is not assigned to an EMIF interface cannot be used as general-purpose I/O pins either.
7. All address and command pins and their associated clock pins (CK_t and CK_c) must reside within a single sub-bank. The sub-bank containing the address and command pins is identified as the address and command sub-bank. Refer to the table in *LPDDR5 Data Width Mapping* for the supported address and command and data lane placements for DDR5.
8. The address and command pins and their associated clock pins in the address and command bank must follow a fixed pin-out scheme, as defined in the *Agilex 7 M-Series External Memory Interface Pin Information* file.
9. An external memory interface can occupy one or more banks on the same edge. When an interface must occupy multiple banks, ensure the following:
   - That the banks are adjacent to one another.
   - That you used only the supported data width mapping as defined in the table in *LPDDR5 Data Width Mapping*. Be aware that not every byte lane can be used as an address and command lane or a data lane.
10. An unused I/O lane in the address and command sub-bank can serve to implement a data group, such as a x8 DQS group. The data group must be from the same controller as the address and command signals.
11. An I/O lane must not be used by both address and command pins and data pins.
12. Place read data groups according to the DQS grouping in the pin table and Pin Planner. Read data strobes (such as RDQS_t and RDQS_c) must reside at physical pins capable of functioning as RDQS_t and RDQS_c for a specific read data group size. You must place the associated read data pins (DQ), within the same group.
13. One of the sub-banks in the device (typically the sub-bank within corner bank 3A) may not be available if you use certain device configuration schemes. For some schemes, there may be an I/O lane available for EMIF data group.

- AVST-8 – This is contained entirely within the SDM, therefore all lanes of sub-bank 3A can be used by the external memory interface.
- AVST-16/AVST-32– Lanes 4, 5, 6, and 7 are all effectively occupied and are not usable by the external memory interface.

14. Two memory interfaces cannot share an I/O 48 sub-bank.

8.2.3.2. Specific Pin Connection Requirements

PLL

For LPDDR5, you must constrain the PLL reference clock to the address and command lanes only.

- You must constrain differential reference clocks to pin indices 10 and 11 in lane 2 when placing command address pins in lane 3 and lane 2.
- The sharing of PLL reference clocks across multiple LPDDR5 interfaces is permitted within an I/O bank.

OCT

For LPDDR5, you must constrain the RZQ pin to the address and command lanes only.

- You must constrain RZQ to pin index 2 in lane 3 when placing command address pins in lane 3 and lane 2.
- The sharing of RZQ across multiple LPDDR5 interfaces is permitted within an I/O bank.

RDQS/DQ/DM

For LPDDR5 x8 DQS grouping, the following rules apply:

- You may use pin indices 0, 1, 2, 3, 8, 9, 10, and 11 within a lane for DQ mode pins only.
- You must use pin index 4 for the RDQS_p pin only.
- You must use pin index 5 for the RDQS_n pin only.
- You must ensure that pin index 7 remains unused. Pin index 7 is not available for use as a general purpose I/O.
- You must use pin index 6 for the DM pin only.

8.2.3.3. Command and Address Signals

Command and address signals in SDRAM devices are clocked into the memory device using the CK_t or CK_c signal.

8.2.3.4. Clock Signals

LPDDR5 SDRAM devices use CK_t and CK_c signals to clock the address and command signals into the memory.

The memory uses these clock signals to generate the DQS signal during a read through the DLL inside the memory.

8.2.4. Pin Placements for Agilex 7 M-Series FPGA LPDDR5 EMIF IP

8.2.4.1. Address and Command Pin Placement for LPDDR5

Agilex 7 M-Series FPGA LPDDR5 IP supports fixed address and command pin placement as shown in the following table. The IP supports up to 2 ranks.

Table 170. Address and Command Pin Placement

Address/Command Lane	Index Within Byte Lane	LPDDR5
AC1	11	
	10	
	9	
	8	CS_N[1]
	7	CK_C[0]
	6	CK_T[0]
	5	CS_N[0]
	4	CA[6]
	3	RESET_N
	2	RZQ Site
	1	
	0	
AC0	11	Differential "N-Side" reference clock input site
	10	Differential "P-Side" reference clock input site
	9	CA[5]
	8	CA[4]
	7	WCK_C[1]
	6	WCK_T[1]
	5	WCK_C[0]
	4	WCK_T[0]
	3	CA[3]
	2	CA[2]
	1	CA[1]
	0	CA[0]

8.2.4.2. LPDDR5 Data Width Mapping

The EMIF IP for Agilex 7 M-Series does not support flexible data lane placement.

Only fixed byte lanes within the I/O bank can be used as data lanes. The following table lists the supported address and command and data lane placements in an I/O bank.

Table 171. Component

Controller	Data Width Usage	BL7 [P95: P84]	BL6 [P83: P72]	BL5 [P71: P60]	BL4 [P59: P48]	BL3 [P47: P36]	BL2 [P35: P24]	BL1 [P23: P12]	BL0 [P11: P0]		BL7 [P95: P84]	BL6 [P83: P72]	BL5 [P71: P60]	BL4 [P59: P48]	BL3 [P47: P36]	BL2 [P35: P24]	BL1 [P23: P12]	BL0 [P11: P0]
Primary	LPDDR5 x16	GPIO	GPIO	GPIO	GPIO	AC1 P	AC0 P	DQ[1] P	DQ[0] P									
Primary	LPDDR5 x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	GPIO	GPIO	GPIO	GPIO									
Primary & Secondary	LPDDR5 2ch x16	DQ[1] S	DQS[0] S	AC1 S	AC0 S	AC1 P	AC0 P	DQ[1] P	DQ[0] P									
Primary	LPDDR5 x32	DQ[3] P	DQ[2] P	GPIO	GPIO	AC1 P	AC0 P	DQ[1] P	DQ[0] P									
Primary & Secondary	LP DDR5 4ch x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC0 P	DQ[1] P	DQ[0] P		DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC0 P	DQ[1] P	DQ[0] P

Note:
- **P** Primary controller.
- **S** Secondary controller.

8.2.4.3. LPDDR5 Byte Lane Swapping

The data lane can be swapped when the byte-lanes are utilized as DQ/DQS pins. Byte lane swapping on utilized lanes is allowed when you swap all the DQ/DQS/DM pins in the same byte lane with the other utilized byte lane.

The rules for swapping DQ byte lane are as follows:

- You can only swap between utilized DQ lanes.
- You cannot swap a DQ lane with an AC lane.
- Additional restrictions apply when you use a x16 memory component:
  - You must place DQ group 0 and DQ group 1 on adjacent byte lanes, unless they are separated by AC lanes. These 2 groups must be connected to the same x16 memory component.
  - You must place DQ group 2 and DQ group 3 on adjacent byte lanes, unless they are separated by AC lanes. These 2 groups must be connected to the same x16 memory component.
  - If you use only one byte of the x16 memory component, you must use only the lower byte of the memory component.

Table 172. Component

Controller	Data Width Usage	BL7 P95:P84	BL6 P83:P72	BL5 P71:P60	BL4 P59:P48	BL3 P47:P36	BL2 P35:P24	BL1 P23:P12	BL0 P11:P0
Primary & Secondary	LPDDR5 2ch x16	DQ[1] S	DQS[0] S	AC1 S	AC0 S	AC1 P	AC0 P	DQ[1] P	DQ[0] P
Primary	LPDDR5 x32	DQ[3] P	DQ[2] P	GPIO	GPIO	AC1 P	AC0 P	DQ[1] P	DQ[0] P

Note:
- P Primary controller.
- S Secondary controller.

Example 1: LPDDR5 2 ch x16

DQ[0] and DQ[1] of the primary controller are can swapped with each other. DQ[0] and DQ[1] of the secondary controller can be swapped with each other.

Example 2: LPDDR5 x32

DQ[0] and DQ[1] can be swapped with each other. DQ[2] and DQ[3] can be swapped with each other.

8.3. LPDDR5 Board Design Guidelines

This section provides board layout design recommendations and guidelines for Agilex 7 M-Series FPGAs, with GPIO-B (input/output) silicon implementation to support LPDDR5.

This PCB layout guideline covers various supported LPDDR5 topologies along with maximum supported data rate that you can use for a successful PCB design.

A successful PCB design requires not only following the topology and routing guidelines here, but must also meet PDN design requirements.

For related information, refer also to the Agilex 7 F, I, and M-Series PDN design guidelines and the Agilex 7 high speed transceiver PCB design guidelines, available on the Intel website.

8.3.1. PCB Stack-up and Design Considerations

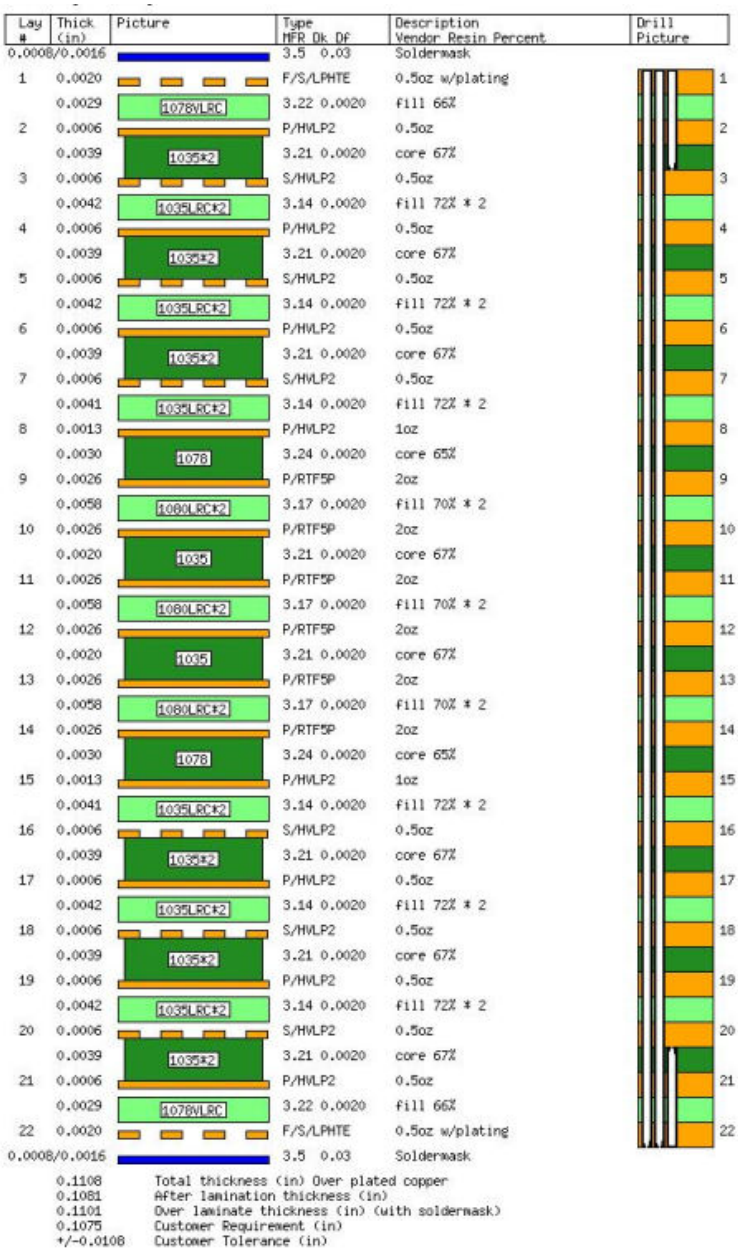
The following figure shows an example of an 18-layer PCB stackup that has been used for LPDDR5 on an Intel platform board. You may use other stackups (thin such as PCIE board, or thick board), provided you follow the recommendations in these guidelines.

Figure 53. 18-Layer Thin Board Type-4 PCB with Micro Via, Stacked Via, Buried Via and Through Via

Layer		Cu Weight	Structure	Proposed Thickness (mils)	Structure	Dk @ 1Ghz	Df @ 1Ghz	Copper type	Ref
	Soldermask			0.50		3.6	0.0195		
L1	Top	1/3oz+Plating		1.60				RTF	L2
	Prepreg			2.20	1035 RC 72	2.9	0.001		
L2	Plane	1/3oz+Plating		0.65				RTF	
				3.11	1078 RC70	2.9	0.001		
L3	Signal	1/3oz+Plating		0.65				RTF	L2&4
				3.30	1078 RC70	2.9	0.001		
L4	Plane	1/3oz+Plating		0.65				RTF	
				3.02	1078 RC70	2.9	0.001		
L5	Signal	1/3oz+Plating		0.80				RTF	L4&6
				3.32	1078 RC70	2.9	0.001		
L6	Plane	Hoz		0.60				HVLP	
				2.00	2.0mil core	2.9	0.001		
L7	Signal	Hoz		0.60				HVLP	L6&8
				2.96	1078 RC70	2.9	0.001		
L8	Plane	Hoz		0.60				RTF	
				2.00	2.0mil core	2.9	0.001		
L9	Plane	2oz		2.40				RTF	
				3.35	2 x 1035 RC 72	2.9	0.001		
L10	Plane	2oz		2.40				RTF	
				2.00	2.0mil core	2.9	0.001		
L11	Plane	Hoz		0.60				RTF	
				2.96	1078 RC70	2.9	0.001		
L12	Signal	Hoz		0.60				HVLP	L11&13
				2.00	2.0mil core	2.9	0.001		
L13	Plane	Hoz		0.60				HVLP	
				3.32	1078 RC70	2.9	0.001		
L14	Signal	1/3oz+Plating		0.80				RTF	L13&15
				3.02	1078 RC70	2.9	0.001		
L15	Plane	1/3oz+Plating		0.65				RTF	
				3.30	1078 RC70	2.9	0.001		
L16	Signal	1/3oz+Plating		0.65				RTF	L15&17
				3.11	1078 RC70	2.9	0.001		
L17	Plane	1/3oz+Plating		0.65				RTF	
	Prepreg			2.20	1035 RC 72	2.9	0.001		
L18	Bottom	1/3oz+Plating		1.60				RTF	L17
	Soldermask			0.50		3.6	0.0195		
Finished Thickness (mils)				65.27					

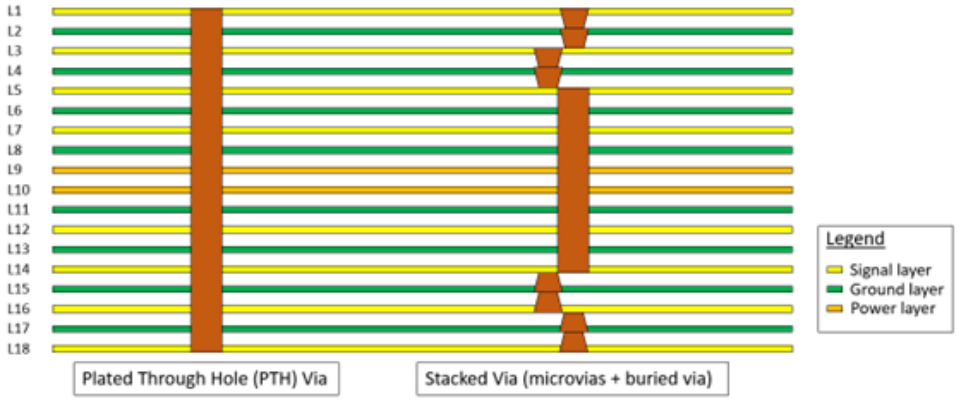
The following figure shows an example of a 22-layer thick PCB stackup, as used with some Intel platform boards and development kits.

Figure 54. 22-Layer Thick Type-4 Board Stack-up, High Performance with Micro Via and PTH, with and without Backdrill



A high-quality type-4 PCB uses not only plated-through-hole (PTH) vias to connect from the top to bottom layer, but also stacked vias, micro vias, and buried vias to connect between layers. For example, a full-height stacked via of an 18 layer PCB consists of a combination of dual-stacked micro vias and buried vias. The following figure shows a cross-sectional comparison of a PTH and stacked via.

Figure 55. Cross-sectional Comparison of Plated-Through Hole Via and Stacked Via



To support maximum data rate operation, LPDDR5 board design requires a high-quality PCB stackup using micro vias, buried vias, or stacked vias to reduce crosstalk for high performance. Reducing the length of signal via is essential to minimizing the crosstalk between signals.

A type-3 PCB with zero-built up layers and PTH vias which is used to implement a DDR4 design can also be used for LPDDR5 designs if backdrill is implemented to reach the maximum supported data rate.

8.3.2. General Design Considerations

Intel recommends that you route all data signals within a specific group on the same layer.

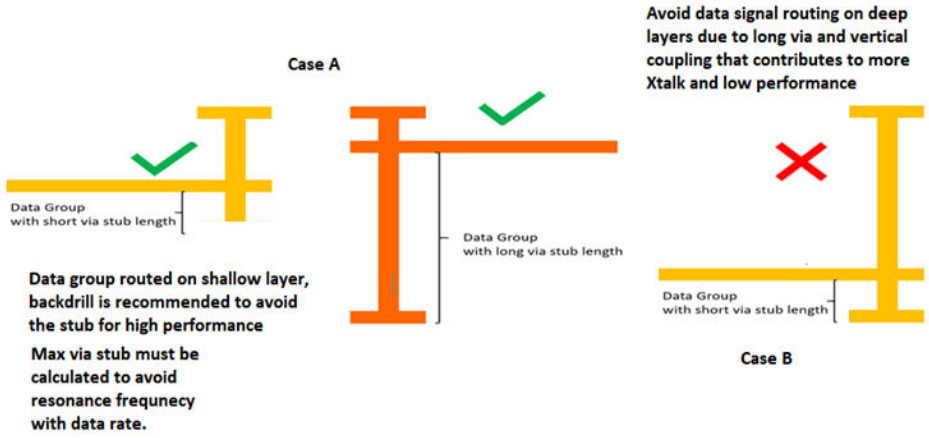
The following figure illustrates a routing example for a type-3 PCB for an LPDDR5 design. Intel recommends that you route Data Group signals such as DQ, DM and DQS on shallow layers as stripline, with the least Z-height via transition to avoid vertical crosstalk for high performance.

The recommended routing layers for Data Group on an 18-layer board using plated-through vias are on the top half of the PCB, such as layers 3, 5, and 7. Other signals such as CA, CTRL, and clock signals can be routed with longer Z-height via transitions on the bottom half of the PCB, such as layers 12, 14, and 16.

Minimal stub effect or back drill is recommended but not mandatory to avoid high reflection for maximum data rate performance for an LPDDR5 interface. Long via stubs will affect the intersymbol interference (ISI) of the channel, but the impact of ISI is less than the impact of crosstalk for maximum performance.

You should avoid stub and use of back drill for LPDDR5 design to reach maximum data rate performance.

Figure 56. Suggested Routings



In the above figure, case A routing is suggested for LPDDR5 Data Group signals over case B, to support maximum data rate. If data signals are routed on deeper layers (as in case B, with long via and short stub), the impact of crosstalk is significant and causes reduced data rate and performance.

To minimize horizontal crosstalk between signals on the same layer, PCB designers must maintain adequate signal trace-to-trace (edge to edge) space with a minimum spacing of 3 x *h* separation distance, where *h* is the dielectric thickness to the closest reference plane, as illustrated below.

Figure 57. Minimum Trace-to-Trace Separation Distance

8.3.3. DDR Differential Signals Routing

DQS and CLK signals in the DDR interface are differential signals and must be routed on the PCB as differential signals unless there is a limitation for PCB routing, such as having a very small pitch at DRAM.

You should have a symmetrical fan-out routing at the FPGA pin field. Non-symmetrical routing for differential signals causes shifting on common-mode voltage and contributes to reduced timing margins at the receiver. The following figures show the recommended differential routing at the FPGA pin field for DQS/CLK signals.

Figure 58. Symmetrical Routing of Differential Signals (DQS/CLK) at FPGA Pin Field, with Length/Skew Matching Between P/N Lanes After FPGA Device Edge

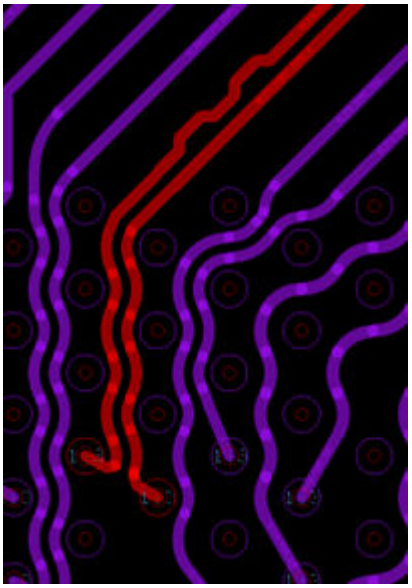
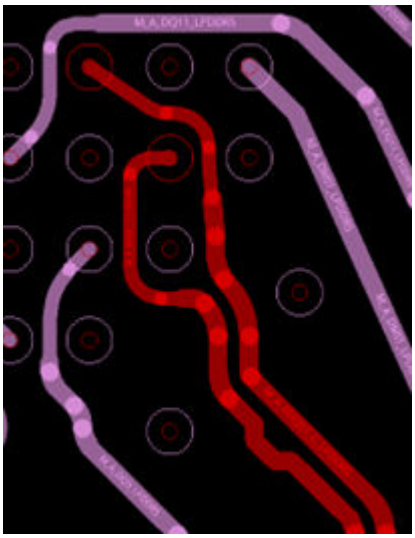


Figure 59. Single-Ended Routing for Differential Signals (DQS/CLK) at DRAM Pin Field with Very Small Pitch and Skew Matching at Edge of DRAM Pin Field



Intel recommends implementing length and skew matching for differential signals immediately after the FPGA device to avoid additional shifting on differential signals common mode voltage.

In cases where very small DRAM device pitch limits the implementation of symmetrical routing at the DRAM pin field for differential signals, it is recommended to route the differential signals as single-ended signals within the DRAM pin field, ensuring to maintain the same impedance while changing from differential to single-ended

configuration. Designers must also keep the same length of routing for each P and N single-ended lane within the DRAM pin field. The skew matching between P and N lanes must be applied before reaching the DRAM pin field.

8.3.4. Ground Plane and Return Path

A continuous and solid ground-reference plane is crucial for data lines, to ensure good signal integrity.

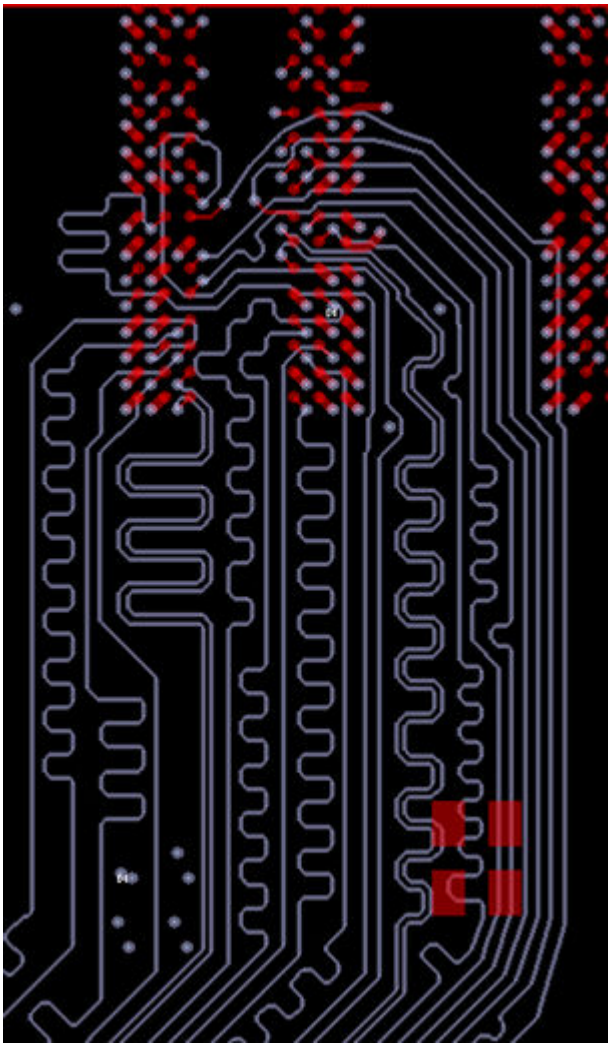
It is important to provide a low-impedance ground return path between the FPGA and DRAM devices, and to keep ground stitching vias within 80 mils from signal transition, for best return path on signal vias and improved signal integrity.

8.3.5. DRAM Break-in Layout Guidelines

For discrete DRAM components on PCB, you can use either the dog-bone or via in pad at the DRAM for the signal transition from inner layer to DRAM.

If you use dog-bone via transitions, you should separate them with larger pitch, to avoid crosstalk between the signal vias.

Figure 60. Data Signal Group Routing on PCB to Memory Down Configuration



8.3.6. LPDDR5 PCB Layout Guidelines

This section describes PCB layout guidelines for an LPDDR5 interface.

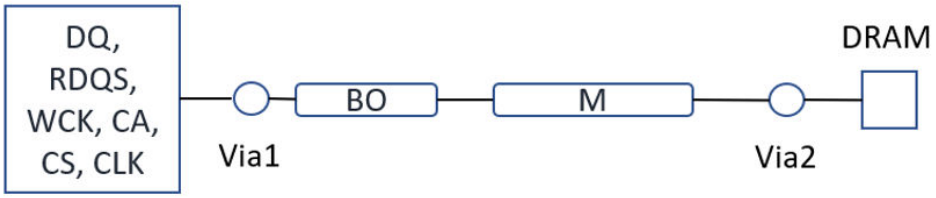
Agilex 7 M-Series devices support LPDDR5 interfaces only for memory down configuration. The LPDDR5 interface supports both thin and thick PCB stackups. The maximum supported data rates vary depending on the selected topology.

8.3.6.1. LPDDR5 Discrete Component/Memory Down Topology (1 Rank or 2 Rank, up to 64 Bit Interface)

LPDDR5 memory down support is available in two configurations: single rank or dual rank, up to 64 bit interface.

There are four DRAM interface signal groupings: Data Group, Command-Address Group, Control Group, and Clock Group. The connection between the FPGA and DRAM uses point-to-point topology for Data, Command/Address, Control, and Clocks, as shown in the following figure.

Figure 61. Point-to-Point Connection for Data, CA, CTRL, and Clock Signals Topology for LPDDR5



The LPDDR5 interface does not support a traditional dual-directional data-strobe architecture. However, two single-directional data strobes such as Write Clock (WCK) for Write Operations and an optional Read Clock (RDQS) for Read Operations are supported.

The following two figures show the connection topology for DQ, WCK signal and CA, CLK, CTRL signals for LPDDR5.

Figure 62. WCK Signals Topology for LPDDR5 Memory Down, T-Line Connection, Depending on EMIF Topology

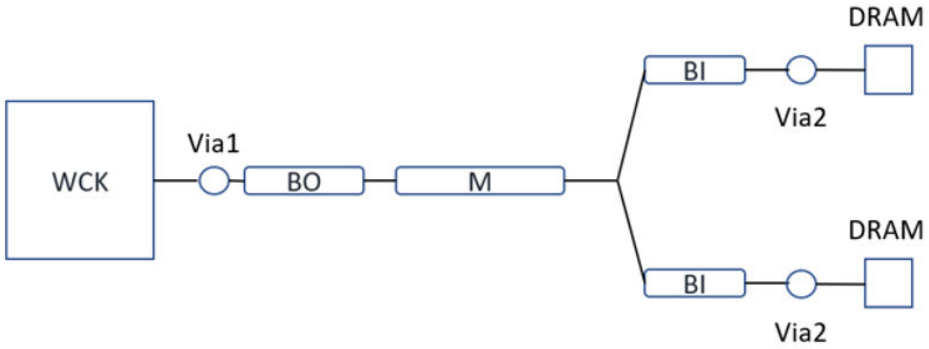
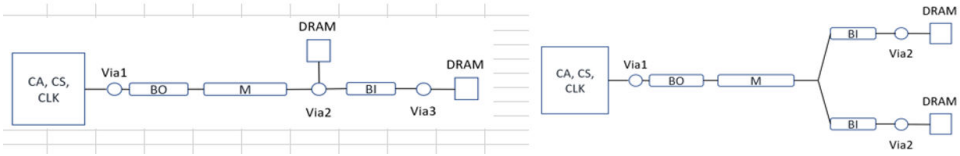


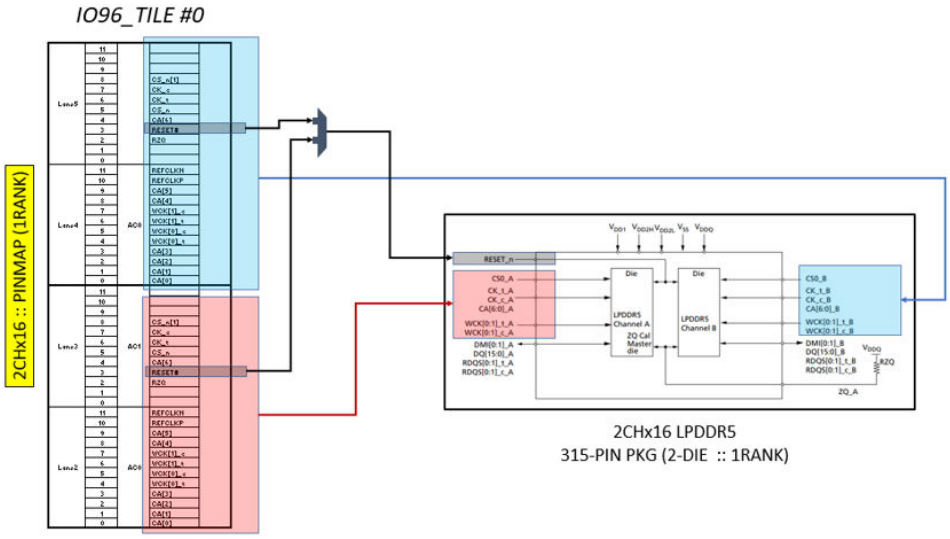
Figure 63. CS, CLK, CTRL Signals Topology for LPDDR5 Memory Down: Daisy or T-Line Connections, Depending on EMIF Topology



8.3.6.2. Supported LPDDR5 Topologies

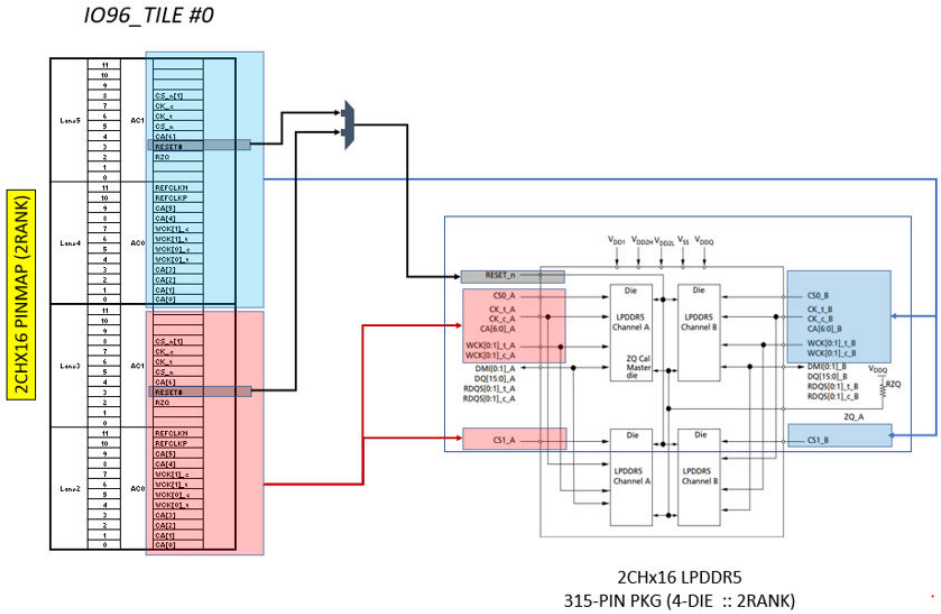
The figures in this topic show various supported LPDDR5 topologies, based on the type of DRAM component, number of supported channels, and ranks.

Figure 64. Dual-Channel Single-Rank x16, Dual-Die DRAM Component



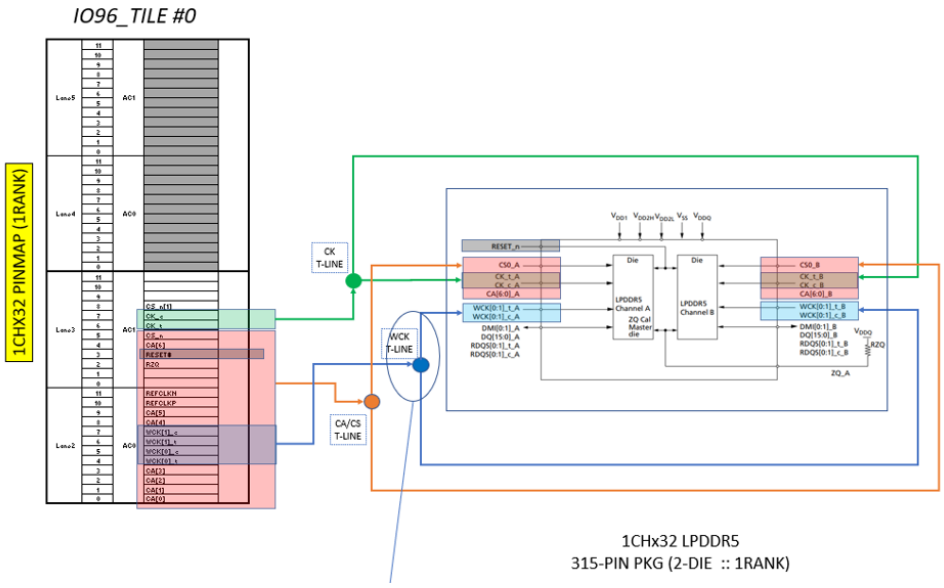
In the above figure, all signals from the controller to DRAM package are point-to-point connections.

Figure 65. Dual-Channel Dual-Rank x16, Using Quad-die DRAM Component



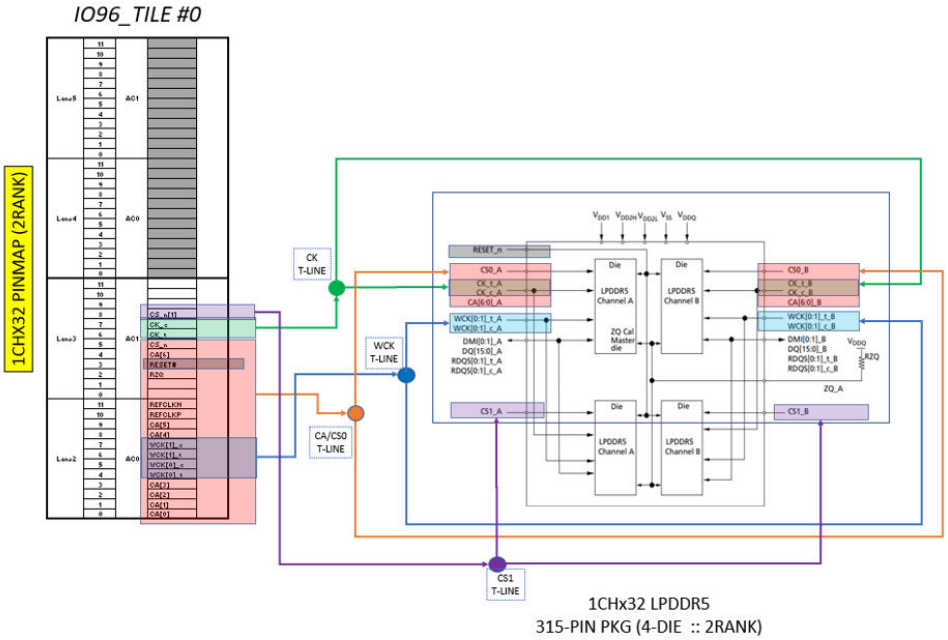
In the above figure, all signals from the controller to DRAM package are point-to-point connections.

Figure 66. Single-Channel Single-Rank x32, Using Dual-die DRAM Component



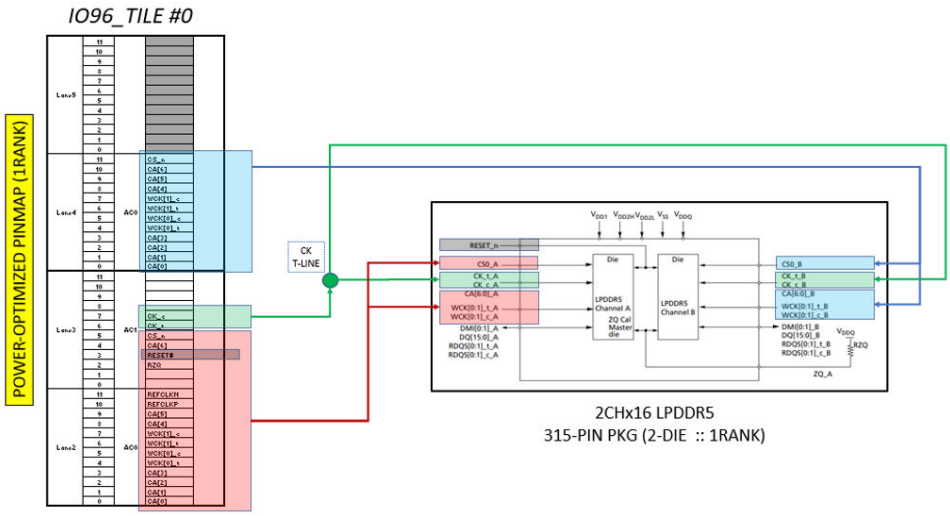
In the above figure, CK, WCK and CA/CS signals from the controller to DRAM are via T-Line connection; each WCK signal from the controller can be connected to both DRAM dies via T-Line, or can be connected to both WCK signals at each DRAM die via T-line.

Figure 67. Single-Channel Dual-Rank x32, Using a Quad-die DRAM component



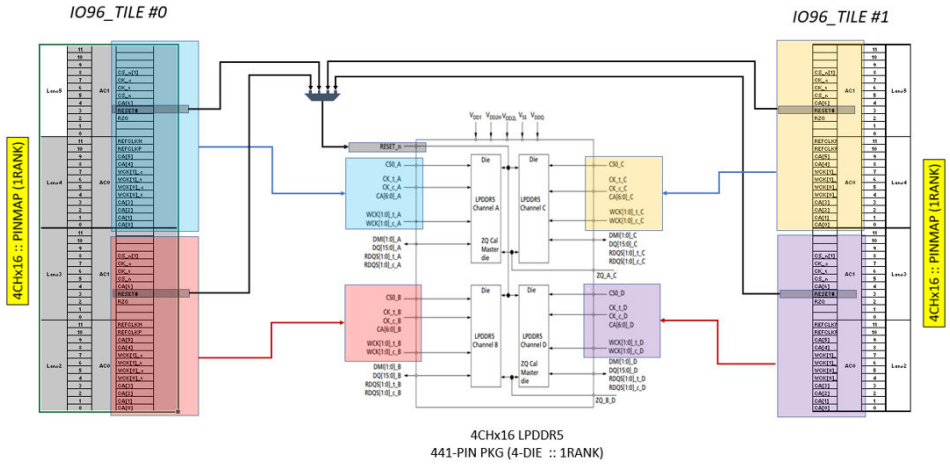
In the above figure, CK, WCK and CA/CS signals from the controller to DRAM are via T-Line connection.

Figure 68. Dual-Channel Single Rank x16, Using a dual-die DRAM Component



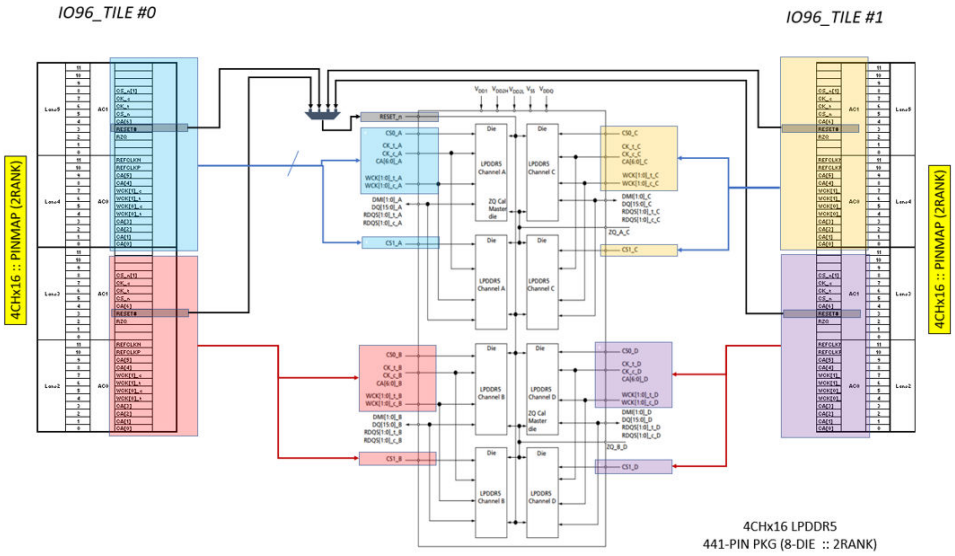
In the above figure, all signals from the controller to DRAM package are point to point connections, except the CK signal which is connected via T-Line in power mode optimization.

Figure 69. Quad-Channel Single Rank x16, Using a Quad-Die DRAM Component



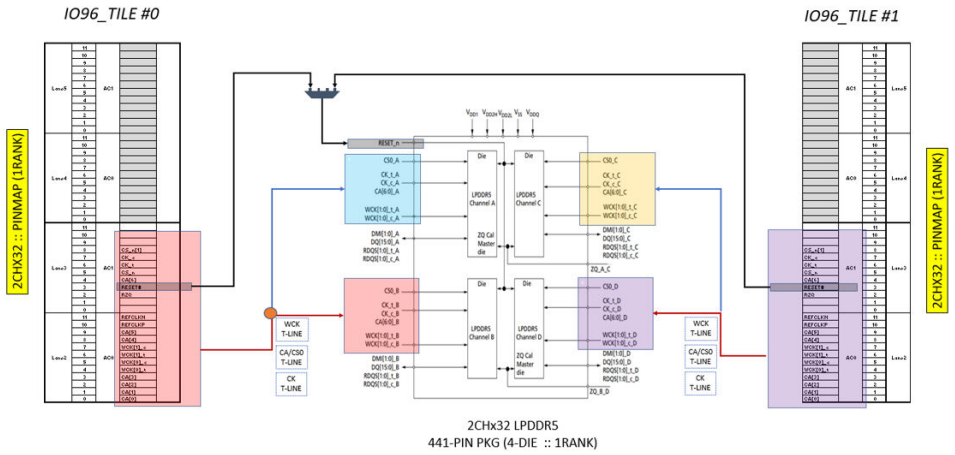
In the above figure, all signals from the controller to DRAM package are point-to-point connections

Figure 70. Quad-Channel Dual-Rank x16, Using an 8-die DRAM Component



In the above figure, all signals from the controller to DRAM package are point-to-point connections.

Figure 71. Dual-Channel-Single -Rank x32, Using a Quad-Die DRAM Component



In the above figure, CK, WCK and CA/CS signals are connected to both DRAM dies via T-Line connection.

The following table provides comprehensive routing guidelines (recommended trace impedance and length) for each LPDDR5 signal, based on a memory down topology. For example, the maximum length of the main trace routing can be derived from total trace length by subtracting the break-out and break-in trace segment lengths.

The signal trace width, and minimum spacing/gaps (in mils) from edge-to-edge of signal traces are based on the default stackup shown in the PCB Stack-up and Design Considerations topic; however, PCB designers can use the target impedance for any other stackups. The *h* value in the table represents the minimum substrate height from signal layer to reference layer.

Table 173. Routing Guidelines for LPDDR5 Memory Down Topology

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohm)	Trace Width, W (mil)	Trace Spacing, S1 (mil): Within Group	Trace Spacing, S2 (mil): CMD/CTRL/CLK to DQ/DQS	Trace Spacing, S3 (mil): DQ Nibble to Nibble	Trace Spacing (mil), Within DIFF pair	Trace Spacing (mil), DQS pair to DQ	Trace Spacing (mil), CLK pair to CMD/CTRL/CKE
			Segment	Total Max								
DQ, RDQS	BO	SL	1000	3000		4	5, 17	17	17	4		
	M	SL			40	5.2	6 (2h)	9 (3h)	9 (3h)	4	9 (3h)	
CA/CS/CLK (Direct Connect)	BO1/BO2	SL	450	4000		3	5, 17	17		4		17
	M	SL			40	5.2		9 (3h)		4		9 (3h)
	BI	SL	50			3		9 (3h)		4		9 (3h)
CA/CS/CLK (Daisy)	BO	SL	500	5000 (to last DRAM for Daisy)		3	5, 17	17		4		17
	M	SL			40	5.2	6 (2h)	9 (3h)		4		9 (3h)
	BI	SL	1000			3	6 (2h)	9 (3h)		4		9 (3h)
WCK	BO	SL	1000	3000		4	5, 17	17		4		
	M	SL			40	5.2	6 (2h)	9 (3h)		4	9 (3h)	
	BI	SL	200 ~ 500			3	6 (2h)	9 (3h)		4		

Memory Down Topology Guidelines	
Reference plane	Continuous Ground Only
Use 3x of Dielectric Height for serpentine routing spacing	

Reset signal routing design also follows the CMD/ADD/CTRL routing design. Maintain at least 5x h edge-to-edge spacing from the Reset signal to other signals on the same layer. There is no requirement to have skew matching between Reset signal and CLK signal.

Skew matching for the LPDDR5 interface consists of both package routing skew and PCB physical routing skew. You must maintain skew matching of CA and CTRL with respect to the clock signals to ensure signals at the receiver are correctly sampled. In addition, there are skew matching requirements for DQ and DQS within a byte group, DQS and CLK.

The following table provides a detailed skew matching guideline to facilitate PCB trace routing efforts. The length matching criteria in the table below represents a default PCB on an Intel platform board design. Skew matching criteria must be always followed in any other stackup.

Table 174. Skew Matching Requirement for LPDDR5 Memory Down Topology

Length Matching Rules	Length	Time (assuming 170ps/in delay)
Length matching between DQ and WCK per x16	-400mil < DQ - WCK < 650mil	-68ps < DQ - WCK < 110ps
Length matching between DQ and RDQS per x8	-120mil < DQ - RDQS < 120mil	-20ps < DQ - RDQS < 20ps
Length matching between WCK and CLK per x16	-200mil < WCK - CLK < 400mil	-34ps < WCK - CLK < 68ps
Length matching between DQ signals per x8	< 160mil	< 27ps
Length matching between channel-to-channel	< 1000mil	< 170ps
Length matching between CLK and CA bits per x16 (CA needs to reference to CLK)	-200mil < CLK - CA < 200mil	-34ps < CLK - CA < 34ps
Length matching between CA bits per x16	< 300mil	< 51ps
Length matching between CLK and CS per x16	-500mil < CLK - CS < 500mil	-85ps < CLK - CS < 85ps
Length matching between CS bits per x16	< 400mil	< 68ps
Length matching between RDQS_N and	< 5mil	< 1ps
Length matching between WCK_N and WCK_P	< 5mil	< 1ps
Length matching between CLK_N and CLK_P	< 5mil	< 1ps
Length matching between WCK Tee segments	< 5mil	< 1ps
Include package length in length matching	Required	Required
Notes		
Keep GND stitching via within 80mil from signal transition which changes reference planes.		
Use 3x of Dielectric Height for serpentine routing spacing		

LPDDR5 eye margin is sensitive to crosstalk, especially when the signals are routed on deep layers in the stackup. The deep-layer vertical transition induces more vertical coupling between signals and hence more crosstalk.

Intel recommends keeping the via transition depth in Z-direction to less than 16mil (routed on shallow layers with backdrill) to achieve high performance on the LPDDR5 interface. The maximum data rate of LPDDR5 depends on the type of PCB and on the DDR memory down configuration as seen in the following table.

Table 175. DQ Routing Summary for LPDDR5 Memory Down

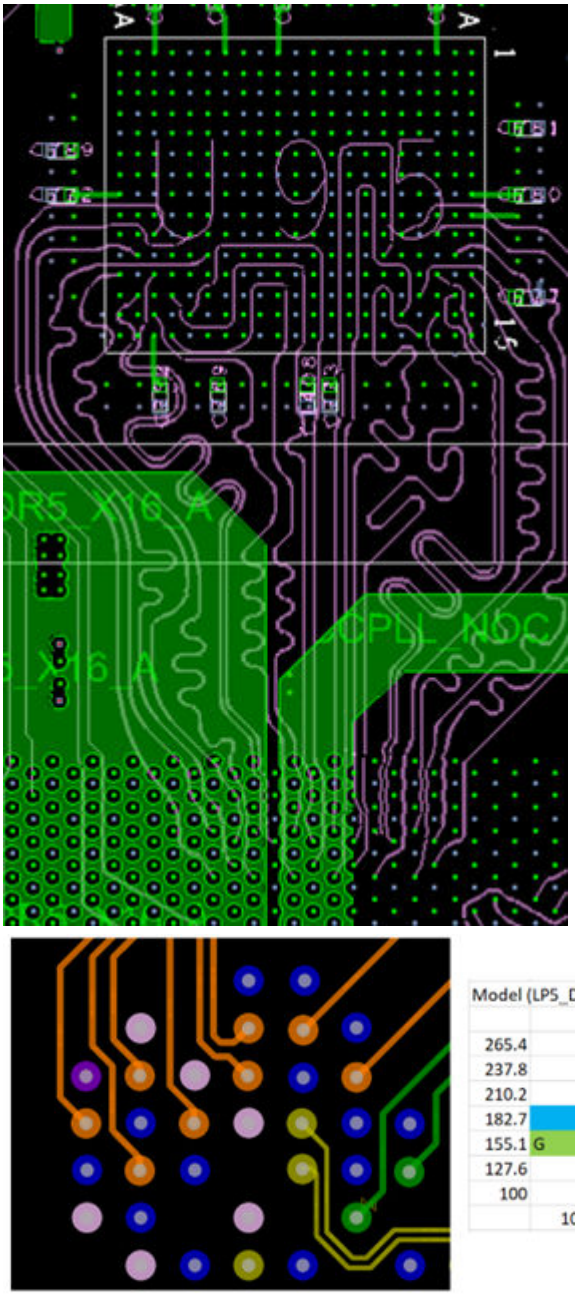
Memory	LPDDR5	
Signal; Group	**DQ**	
Board Thickness (mil)	65mil or 120mil (Routing must be on upper layers, max via transition depth <=16mil)	65mil or 120mil (Routing must be on upper layers, max via transition depth <=16mil)
Maximum Z-transition height (mil)	16	16
PCB Stripline Trace Impedance (Ohms)	40	40
Memory Configuration	Memory Down	Memory Down
# of Rank	1 (x16 bit or x32 bit Double Die)	2 (x 16 or x 32 bit)
Maximum Length Total (Inch)	3.0	3.0
Notes	Maximum package length in FPGA design is shorter than 34mm.	

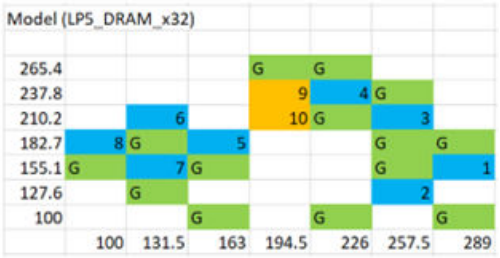
8.3.6.3. Example of an LPDDR5 Layout on an Intel FPGA Platform Board

The following figures show the layout example of a single rank LPDDR5 x 32-bit device with a pitch size of 0.7×0.8mm on an Intel FPGA platform design.

This example has been designed on a thick PCB (120mil stackup) using micro vias and through vias with backdrill. The LPDDR5 signal routing is on upper layers to avoid vertical crosstalk on interface and achieve high performance.

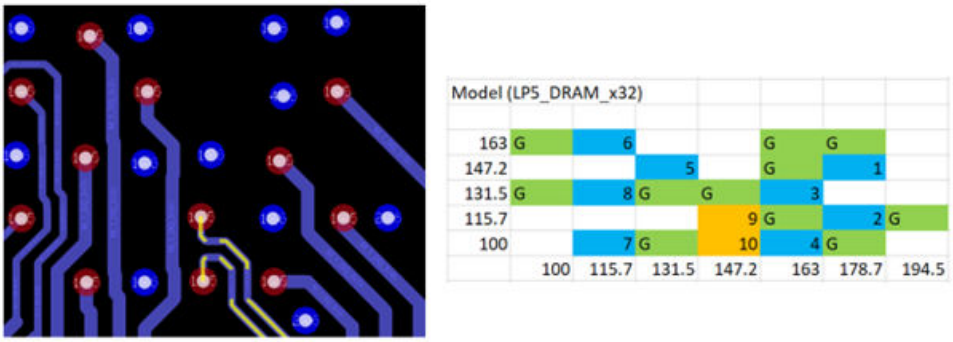
Figure 72. Board Layout and Via Pattern for Single Rank LPDDR5 x32 device on an Intel FPGA Platform Design





In addition, the following figure shows a LPDDR5 64-bit device board routing sample with pitch dimension of 0.4×0.4mm. The microvia has been used for via transitions on this interface.

Figure 73. Board Via Pattern for LPDDR5 64-bit Device



8.3.7. LPDDR5 Simulation Strategy

The simulation strategy has two parts:

- Data Signal signal integrity simulation with respect to their DQS on the worse signal integrity of a data group (considering the longest routing and maximum vertical crosstalk between signals).
- CS/CTRL/CMD signal integrity simulation with respect to their CLK signals on the worst signal integrity of those signals (considering the longest routing and maximum vertical crosstalk between signals).

Intel recommends that the signal integrity engineer review the layout and pick the worst data group (select a victim and surrounded aggressors and DQS in the group) that has the worst signal integrity on the layout (that is, the worst cross talk coupling between deep vertical vias), long trace/PCB routing and maximum reflection on the routing path due to long via stubs if backdrilling is not applied.

Designers must perform signal integrity simulation of the board layout for the selected victim surrounded by aggressor signals.

The channel analysis must be performed in the time domain (using PRBS pattern for I/O signal generator) while the channel is built by using actual per-pin package model at both ends, PCB model in the format of scattering parameter along with I/O buffer model at both ends. I/O buffer IBIS model was used for DDR4 interface SI simulation; however, LPDDR5 requires an IBIS AMI buffer model (due to the equalizations/FFE/DFE at both TX and RX) at both ends to recover the data. Eye diagram is evaluated after the simulation to meet eye specification at both ends.

Note: Currently the FPGA LPDDR5 GPIO-B buffer IBIS AMI model is not available for designers to do the signal integrity simulation. Designers are advised to follow strictly the PCB routing design guideline in this document to meet maximum supported data rate per selected configuration.

9. Agilex 7 M-Series FPGA EMIF IP – Timing Closure

This chapter describes timing analysis and optimization techniques that you can use to achieve timing closure within the FPGA.

Note: At this time, Agilex 7 M-Series device timing models have not been verified by silicon characterization.

9.1. Timing Closure

The following sections describe the timing analysis using the respective FPGA data sheet specifications and the user-specified memory data sheet parameters.

- Core to core (C2C) transfers have timing constraints created and are analyzed by the Timing Analyzer. Core timing does not include user logic timing within core or to and from the EMIF block. The EMIF IP provides the constrained clock to the customer logic.
- Core to periphery (C2P) transfers have timing constraints created and are timing analyzed by the Timing Analyzer.
- Periphery to core (P2C) transfers have timing constraints created and are timing analyzed by the Timing Analyzer.
- Periphery to periphery (P2P) transfers are modeled entirely by a minimum pulse width violation on the hard block, and have no internal timing arc.

To account for the effects of calibration, the EMIF IP includes additional scripts that are part of the `<phy_variation_name>_report_timing.tcl` and `<phy_variation_name>_report_timing_core.tcl` files that determine the timing margin after calibration. These scripts use the setup and hold slacks of individual pins to emulate what is occurring during calibration to obtain timing margins that are representative of calibrated PHYs. The effects considered as part of the calibrated timing analysis include improvements in margin because of calibration, and quantization error and calibration uncertainty because of voltage and temperature changes after calibration.

9.1.1. Timing Analysis

Timing analysis of Agilex 7 M-Series EMIF IP is somewhat simpler than that of some earlier device families, because Agilex 7 M-Series devices have more hardened blocks and fewer soft logic registers to be analyzed, because most are user logic registers.

Your Agilex 7 M-Series EMIF IP includes a Synopsys Design Constraints File (`.sdc`) which contains timing constraints specific to your IP. The `.sdc` file also contains Tool Command Language (`.tcl`) scripts which perform various timing analyses specific to memory interfaces.

9.1.1.1. PHY or Core

Timing analysis of the PHY or core path includes the path from the last set of registers in the core to the first set of registers in the periphery (C2P), or the path from the last set of registers in the periphery to the first of registers in the core (P2C) and the ECC related path if it is enabled.

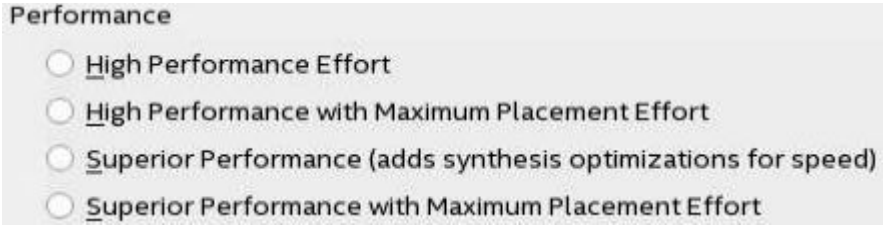
Core timing analysis excludes user logic timing to or from EMIF blocks. The EMIF IP provides a constrained clock (for example: ddr4_usr_clk) with which to clock customer logic; pll_afi_clk serves this purpose.

The PHY or core analyzes this path by calling the `report_timing` command in `<variation_name>_report_timing.tcl` and `<variation_name>_report_timing_core.tcl`.

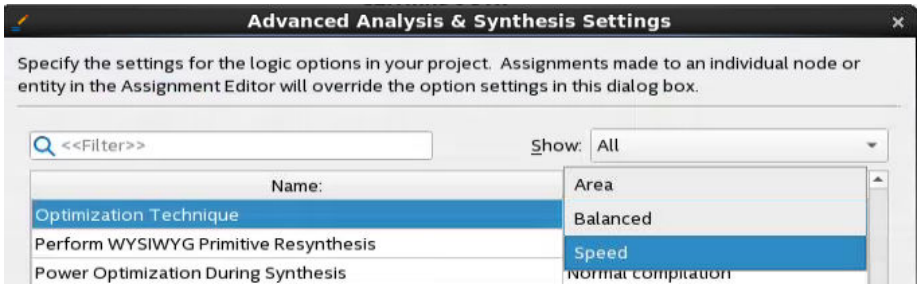
9.2. Optimizing Timing

The Quartus Prime software offers several advanced features that you can use to assist in meeting core timing requirements.

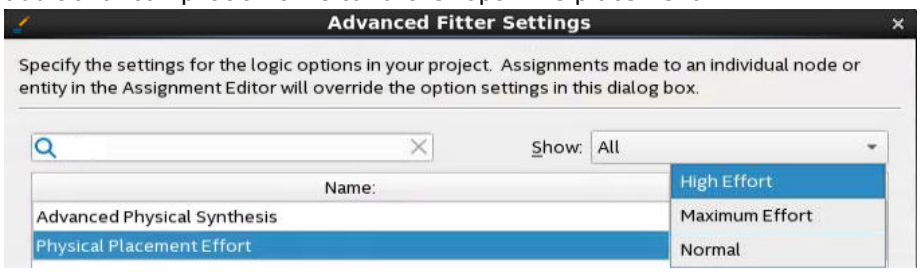
1. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings**. Under **Optimization mode**, select one of the **Performance** options.



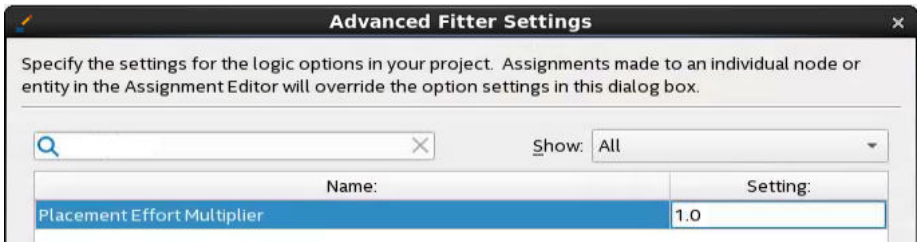
2. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings ➤ Advanced Settings (Synthesis).** For **Optimization Technique**, select **Speed**.



3. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings ➤ Advanced Settings (Fitter).** For **Physical Placement Effort**, select **High Effort** or **Maximum Effort**. The High and Maximum effort settings take additional compilation time to further optimize placement.



4. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings ➤ Advanced Settings (Fitter).** For **Placement Effort Multiplier**, select a number higher than the preset value of 1.0. A higher value increases CPU time, but may improve placement quality.



10. Agilex 7 M-Series FPGA EMIF IP – Controller Optimization

When designing an external memory interface, you should understand the ways available to increase the efficiency and bandwidth of the memory controller.

The following topics discuss factors that affect controller efficiency and ways to increase the efficiency of the controller.

Controller Efficiency

Controller efficiency varies depending on data transaction. The best way to determine the efficiency of the controller is to simulate the memory controller for your specific design.

Controller efficiency is expressed as:

Efficiency = number of active cycles of data transfer/total number of cycles

The total number of cycles includes the number of cycles required to issue commands or other requests.

Note: You calculate the number of active cycles of data transfer in terms of local clock cycles.

10.1. Interface Standard

Complying with certain interface standard specifications affects controller efficiency.

When interfacing the memory device to the memory controller, you must observe timing specifications and perform the following bank management operations:

- **Activate**

  Before you issue any read (RD) or write (WR) commands to a bank within an SDRAM device, you must open a row in that bank using the activate (ACT) command. After you open a row, you can issue a read or write command to that row based on the t_{RCD} specification. Reading or writing to a closed row has negative impact on the efficiency as the controller has to first activate that row and then wait until t_{RCD} time to perform a read or write.

- **Precharge**

  To open a different row in the same bank, you must issue a precharge command. The precharge command deactivates the open row in a particular bank or the open row in all banks. Switching a row has a negative impact on the efficiency as you must first precharge the open row, then activate the next row and wait t_{RCD} time to perform any read or write operation to the row.

- **Device CAS latency**

  The memory device has its own read latency, and the higher the CAS latency, the less efficient an individual access. The higher the operating frequency, the longer the CAS latency is in number of cycles.

- **Refresh**

  A refresh, in terms of cycles, consists of the precharge command and the waiting period for the auto refresh.

10.2. Bank Management Efficiency

Bank management operation affects controller efficiency.

When a read operation reads changes from a row in a bank, it has an impact on efficiency, relative to the row in the bank remaining unchanged.

When a row in the bank is unchanged, the controller does not need to issue precharge and activate commands; by not issuing precharge and activate commands, the speed of the read operation is increased, resulting in better efficiency.

Similarly, if you do not switch between read and write frequently, the efficiency of your controller improves significantly.

10.3. Data Transfer

The following methods of data transfer reduce the efficiency of your controller:

- Performing individual read or write accesses is less efficient.
- Switching between read and write operation reduces the efficiency of the controller.
- Performing read or write operations from different rows within a bank or in a different bank—if the bank and a row you are accessing is not already open—also affects the efficiency of your controller.

10.4. Improving Controller Efficiency

You can use the following methods to improve the efficiency of your controller.

- Frequency of Operation
- Series of Reads or Writes

The following sections discuss these methods in detail.

10.4.1. Frequency of Operation

Certain frequencies of operation give you the best possible latency based on the memory parameters. The memory parameters you specify through the parameter editor are converted to clock cycles and rounded up.

In most cases, the frequency and parameter combination is not optimal. If you are using a memory device that has t_{RCD} = 15 ns and are running the interface at 1200 MHz, you get the following results:

- For quarter-rate implementation (t_{Ck} = 3.33 ns):

  t_{RCD} convert to clock cycle = 15/3.33 = 4.5, rounded up to 5 clock cycles or 16.65 ns.

10.4.2. Series of Reads or Writes

Performing a series of reads or writes from the same bank and row increases controller efficiency.

For best performance, minimize random reads and random writes. When you perform reads and writes to random locations, the operations require row and bank changes. To change banks, the controller must precharge the previous bank and activate the row in the new bank. Even if you change the row in the same bank, the controller has to close the bank (precharge) and reopen it again just to open a new row (activate). Because of the precharge and activate commands, efficiency can decrease by as much as 3–15%, as the controller needs more time to issue a read or write.

If you must perform a random read or write, use additive latency and bank interleaving to increase efficiency.

Controller efficiency depends on the method of data transfer between the memory device and the FPGA, the memory standards specified by the memory device vendor, and the type of memory controller.

11. Agilex 7 M-Series FPGA EMIF IP – Debugging

This chapter discusses issues and strategies for debugging your external memory interface IP.

11.1. Interface Configuration Performance Issues

There are many interface combinations and configurations possible in an Intel design, therefore it is impractical for Intel to explicitly state the achievable f_{MAX} for every combination.

Intel seeks to provide guidance on typical performance, but this data is subject to memory component timing characteristics, interface widths, depths directly affecting timing deration requirements, and the achieved skew and timing numbers for a specific PCB.

FPGA timing issues should generally not be affected by interface loading or layout characteristics. In general, the Intel performance figures for any given device family and speed-grade combination should usually be achievable.

To resolve FPGA (PHY and PHY reset) timing issues, refer to the *Timing Closure* chapter.

Achievable interface timing (address and command, half-rate address and command, read and write capture) is directly affected by any layout issues (skew), loading issues (deration), signal integrity issues (crosstalk timing deration), and component speed grades (memory timing size and tolerance). Intel performance figures are typically stated for the default (single rank, unbuffered DIMM) case. Intel provides additional expected performance data where possible, but the f_{MAX} is not achievable in all configurations. Intel recommends that you optimize the following items whenever interface timing issues occur:

- Improve PCB layout tolerances
- Use a faster speed grade of memory component
- Ensure that the interface is fully and correctly terminated
- Reduce the loading (reduce the deration factor)

11.1.1. Interface Configuration Bottleneck and Efficiency Issues

Depending on the transaction types, efficiency issues can exist where the achieved data rate is lower than expected. Ideally, these issues should be assessed and resolved during the simulation stage because they are sometimes impossible to solve later without rearchitecting the product.

Any interface has a maximum theoretical data rate derived from the clock frequency, however, in practice this theoretical data rate can never be achieved continuously due to protocol overhead and bus turnaround times.

Simulate your desired configuration to ensure that you have specified a suitable external memory family and that your chosen controller configuration can achieve your required bandwidth.

Efficiency can be assessed in several different ways, and the primary requirement is an achievable continuous data rate. The local interface signals combined with the memory interface signals and a command decode trace should provide adequate visibility of the operation of the IP to understand whether your required data rate is sufficient and the cause of the efficiency issue.

To show if under ideal conditions the required data rate is possible in the chosen technology, follow these steps:

1. Use the memory vendor's own testbench and your own transaction engine.
2. Use either your own driver, or modify the provided example driver, to replicate the transaction types typical of your system.
3. Simulate this performance using your chosen memory controller and decide if the achieved performance is still acceptable.

Observe the following points that may cause efficiency or bottleneck issues at this stage:

- Identify the memory controller rate (full, half, or quarter) and commands, which may take two or four times longer than necessary
- Determine whether the memory controller is starved for data by observing the appropriate request signals.
- Determine whether the memory controller processor transactions at a rate sufficient to meet throughput requirements by observing appropriate signals, including the local ready signal.

Consider using either a faster interface, or a different memory type to better align your data rate requirements to the IP available directly from Intel.

11.2. Functional Issue Evaluation

Functional issues occur at all frequencies (using the same conditions) and are not altered by speed grade, temperature, or PCB changes. You should use functional simulation to evaluate functional issues.

The Intel FPGA IP includes the option to autogenerate a testbench specific to your IP configuration, which provides an easy route to functional verification.

The following issues should be considered when trying to debug functional issues in a simulation environment.

11.2.1. Intel IP Memory Model

Intel memory IP autogenerates a generic simplified memory model that works in all cases. This simple read and write model is not designed or intended to verify all entered IP parameters or transaction requirements.

The Intel-generated memory model may be suitable to evaluate some limited functional issues, but it does not provide comprehensive functional simulation.

11.2.2. Vendor Memory Model

Contact the memory vendor directly, because many additional models are available from the vendor's support system.

When using memory vendor models, ensure that the model is correctly defined for the following characteristics:

- Speed grade
- Organization
- Memory allocation
- Maximum memory usage
- Number of ranks on a DIMM
- Buffering on the DIMM
- ECC

Note: Refer to the **readme.txt** file supplied with the memory vendor model, for more information about how to define this information for your configuration. Also refer to Transcript Window messages, for more information.

Note: Intel does not provide support for vendor-specific memory models.

During simulation vendor models output a wealth of information regarding any device violations that may occur because of incorrectly parameterized IP.

11.2.3. Transcript Window Messages

When you are debugging a functional issue in simulation, vendor models typically provide much more detailed checks and feedback regarding the interface and their operational requirements than the Intel generic model.

In general, you should use a vendor-supplied model whenever one is available. Consider using second-source vendor models in preference to the Intel generic model.

Many issues can be traced to incorrectly configured IP for the specified memory components. Component data sheets usually contain settings information for several different speed grades of memory. Be aware data sheets specify parameters in fixed units of time, frequencies, or clock cycles.

The Intel generic memory model always matches the parameters specified in the IP, as it is generated using the same engine. Because vendor models are independent of the IP generation process, they offer a more robust IP parameterization check.

During simulation, review the transcript window messages and do not rely on the Simulation Passed message at the end of simulation. This message indicates only that the example driver successfully wrote and then read the correct data for a single test cycle.

Even if the interface functionally passes in simulation, the vendor model may report operational violations in the transcript window. These reported violations often explain why an interface appears to pass in simulation, but fails in hardware.

Vendor models typically perform checks to ensure that the following types of parameters are correct:

- Burst length
- Burst order
- tMRD
- tMOD
- tRFC
- tREFPDEN
- tRP
- tRAS
- tRC
- tACTPDEN
- tWR
- tWRPDEN
- tRTP
- tRDPDEN
- tINIT
- tXPDLL
- tCKE
- tRRD
- tCCD
- tWTR
- tXPR
- PRECHARGE
- CAS length
- Drive strength
- AL
- tDQS
- CAS_WL
- Refresh
- Initialization
- tIH
- tIS
- tDH
- tDS

If a vendor model can verify that all these parameters are compatible with your chosen component values and transactions, it provides a specific insight into hardware interface failures.

11.3. Timing Issue Characteristics

The PHY and controller combinations automatically generate timing constraint files to ensure that the PHY and external interface are fully constrained and that timing is analyzed during compilation. Nevertheless, timing issues can still occur. This topic discusses how to identify and resolve any timing issues that you may encounter.

Timing issues typically fall into two distinct categories:

- FPGA core timing reported issues
- External memory interface timing issues in a specific mode of operation or on a specific PCB

The Timing Analyzer reports timing issues in two categories: core-to-core and core-to-IOE transfers. These timing issues include the PHY and PHY reset sections in the Timing Analyzer Report DDR subsection of timing analysis. External memory interface timing issues are reported specifically in the Timing Analyzer Report DDR subsection, excluding the PHY and PHY reset. The Report DDR PHY and PHY reset sections only include the PHY, and specifically exclude the controller, core, PHY-to-controller and local interface. Quartus Prime timing issues should always be evaluated and corrected before proceeding to any hardware testing.

PCB timing issues are usually Quartus Prime timing issues, which are not reported in the Quartus Prime software, if incorrect or insufficient PCB topology and layout information is not supplied. PCB timing issues are typically characterized by calibration failure, or failures during user mode when the hardware is heated or cooled. Further PCB timing issues are typically hidden if the interface frequency is lowered.

11.3.1. Evaluating FPGA Timing Issues

Usually, you should not encounter timing issues with Intel-provided IP unless your design exceeds Intel's published performance range or you are using a device for which the Quartus Prime software offers only preliminary timing model support. Nevertheless, timing issues can occur in the following circumstances:

- The **.sdc** files are incorrectly added to the Quartus Prime project
- Quartus Prime analysis and synthesis settings are not correct
- Quartus Prime Fitter settings are not correct

For all of these issues, refer to the correct user guide for more information about recommended settings, and follow these steps:

1. Ensure that the IP generated **.sdc** files are listed in the Quartus Prime Timing Analyzer files to include in the project window.
2. Configure the Settings as follows, to help close timing in the design:

- a. On the **Assignments** menu click **Settings**.
  b. In the **Category** list, click **Compiler Settings**.
  c. Select **Optimization mode ➤ Performance ➤ High Performance Effort**.
- a. On the **Assignments** menu click **Settings**.
  b. In the **Category** list, click **Compiler Settings ➤ Advanced Settings (Synthesis).**.
  c. For **Optimization Technique**, select **Speed**.
- a. On the **Assignments** menu click **Settings**.
  b. In the **Category** list, click **Compiler Settings ➤ Advanced Settings (Fitter)**.
  c. For **Physical Placement Effort**, select **High Effort/Maximum Effort**.

3. Use **Timing Analyzer Report Ignored Constraints**, to ensure that **.sdc** files are successfully applied.
4. Use **Timing Analyzer Report Unconstrained Paths**, to ensure that all critical paths are correctly constrained.

More complex timing problems can occur if any of the following conditions are true:

- The design includes multiple PHY or core projects
- Devices where the resources are heavily used
- The design includes wide, distributed, maximum performance interfaces in large die sizes

Any of the above conditions can lead to suboptimal placement results when the PHY or controller are distributed around the FPGA. To evaluate such issues, simplify the design to just the autogenerated example top-level file and determine if the core meets timing and you see a working interface. Failure implies that a more fundamental timing issue exists. If the standalone design passes core timing, evaluate how this placement and fit is different than your complete design.

Use Logic Lock regions or design partitions to better define the placement of your memory controllers. When you have your interface standalone placement, repeat for additional interfaces, combine, and finally add the rest of your design.

Additionally, use fitter seeds and increase the placement and router effort multiplier.

11.3.2. Evaluating External Memory Interface Timing Issues

External memory interface timing issues usually relate to the FPGA input and output characteristics, PCB timing, and memory component characteristics.

The FPGA input and output characteristics are usually fixed values, because the IOE structure of the devices is fixed. Optimal PLL characteristics and clock routing characteristics do have an effect. Assuming the IP is correctly constrained with autogenerated assignments, and you follow implementation rules, the design should reach the stated performance figures.

Memory component characteristics are fixed for any given component or DIMM. Consider using faster components or DIMMs in marginal cases when PCB skew may be suboptimal, or your design includes multiple ranks when deration may cause read

capture or write timing challenges. Using faster memory components often reduces the memory data output skew and uncertainty easing read capture, and lowering the memory's input setup and hold requirement, which eases write timing.

Increased PCB skew reduces margins on address, command, read capture and write timing. If you are narrowly failing timing on these paths, consider reducing the board skew (if possible), or using faster memory. Address and command timing typically requires you to manually balance the reported setup and hold values with the dedicated address and command phase in the IP.

Refer to the respective IP user guide for more information.

Deration because of increased loading, or suboptimal layout may result in a lower than desired operating frequency meeting timing. You should close timing in the Timing Analyzer software using your expected loading and layout rules before committing to PCB fabrication.

Ensure that any design with an Intel PHY is correctly constrained and meets timing in the Timing Analyzer software. You must address any constraint or timing failures before testing hardware.

For more information about timing constraints, refer to the Timing Analysis chapter.

11.4. Verifying Memory IP Using the Signal Tap Logic Analyzer

The Signal Tap logic analyzer shows read and write activity in the system.

For more information about using the Signal Tap logic analyzer, refer to the *Quartus Prime Pro Edition User Guide: Debug Tools*.

To add the Signal Tap logic analyzer, follow these steps:

1. On the Tools menu click **Signal Tap Logic Analyzer** .
2. In the **Signal Configuration** window next to the **Clock** box, click **...** (Browse Node Finder).
3. Type the memory interface system clock in the **Named** box, for **Filter** select **Signal Tap: presynthesis** and click **Search**.
4. Select the memory interface clock that is exposed to the user logic.
5. Click **OK**.
6. Under Signal Configuration, specify the following settings:
   - For **Sample depth**, select **512**
   - For **RAM type**, select **Auto**
   - For **Trigger flow control**, select **Sequential**
   - For **Trigger position**, select **Center trigger position**
   - For **Trigger conditions** , select **1**

11.5. Debugging with the External Memory Interface Debug Toolkit

The External Memory Interface Debug Toolkit for Agilex 7 M-Series FPGAs provides access to data collected by the sequencer during memory calibration, as well as analysis tools to evaluate the stability of the calibrated interface and assess hardware conditions.

The debug toolkit provides the following types of reports:

- Interface and memory configuration, such as external memory protocol and interface width.
- Calibration results, including calibration status (pass or fail), calibration failure stage (if applicable), delay settings and margins, and VREF settings and margins..
- Traffic test results (pass or fail).
- Calibration report, providing detailed information about the margins observed during calibration.
- Debug prints, which consist of delay settings and margins, as well as VREF settings and margins in each calibration. Debug Prints are printed to a text file for further analysis.

The available task and analysis capabilities include the following:

- Ability to request recalibration of the memory interface.
- Ability to run the test engine in the design example.
- Ability to view the delay setting on any pin in the selected interface and update it if necessary.
- Ability to run VREF margining on the interface.
- Ability to run driver margining on the interface.

11.5.1. Prerequisites for Using the EMIF Debug Toolkit

You must complete the following prerequisites before using the EMIF Debug Toolkit.

1. You must configure your design to use the EMIF Debug Toolkit, as described in the following topics.
2. You must compile your design.
3. You must program the target device with the resulting SRAM Object File (`.sof`).

After completing the above steps, you are ready to run the EMIF Debug toolkit.

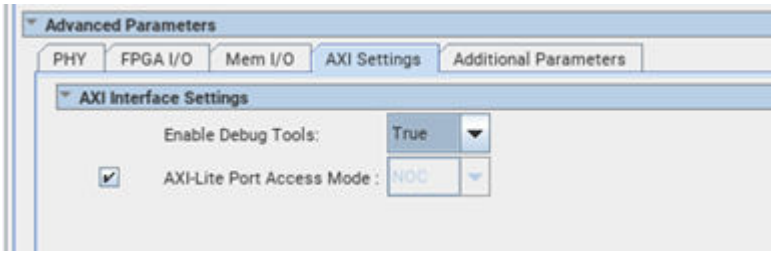
Related Information

Launching the EMIF Debug Toolkit on page 228

11.5.2. Configuring Your Design to Use the EMIF Debug Toolkit

To configure your design for use with the toolkit, you must generate a design example with the debug toolkit.

1. Navigate to the **Advanced Parameters** section of the **General IP Parameters** tab.
2. Select the **AXI Settings** tab.
3. Set **Enable Debug Tools** to **True**.



4. After you have fully parameterized the interface, click **Generate Example Design**. The system generates a design example with the debug toolkit enabled and all necessary components connected as required for a single interface.

Related Information

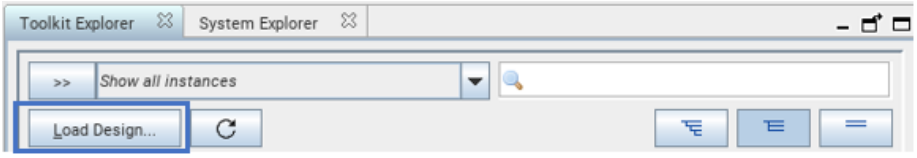
- Design Example Quick Start Guide
  External Memory Interfaces Agilex 7 M-Series FPGA IP Design Example User Guide.
- Launching the EMIF Debug Toolkit on page 228

11.5.3. Launching the EMIF Debug Toolkit

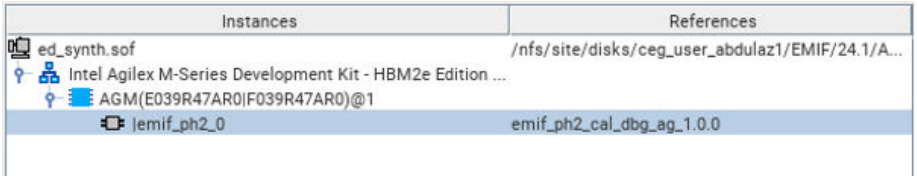
Before launching the EMIF Debug Toolkit, ensure that you have configured your device with a programming file that has the EMIF Debug Toolkit enabled.

To launch the debug toolkit, follow these steps:

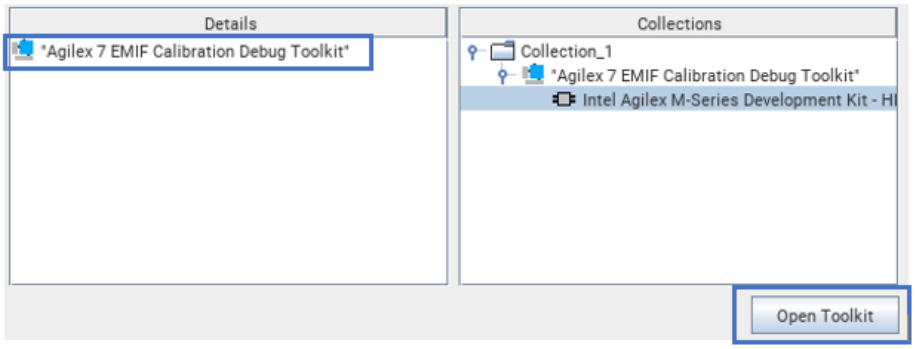
1. In the Quartus Prime software, open the System Console by clicking **Tools ➤ System Debugging Tools ➤ System Console**.
2. In the System Console, load the SRAM Object File (`.sof`) with which you have programmed the board. (Refer to the *Prerequisites for Using the EMIF Debug Toolkit* topic.)



3. Select the correct instance that you want to debug.

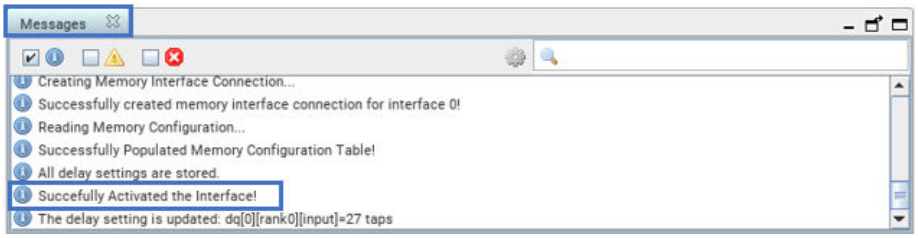


4. Select **Agilex 7 EMIF Calibration Debug Toolkit** from the **Details** section. Click **Open Toolkit** to open the main view of the toolkit.



5. Click **Activate Interface** to allow the toolkit to read the parameters and status for the selected interface, and to perform analysis tasks.

6. Successful activation of the Debug Toolkit is indicated by a message in the Messages tab.



Related Information

- Configuring Your Design to Use the EMIF Debug Toolkit on page 227
- Prerequisites for Using the EMIF Debug Toolkit on page 227

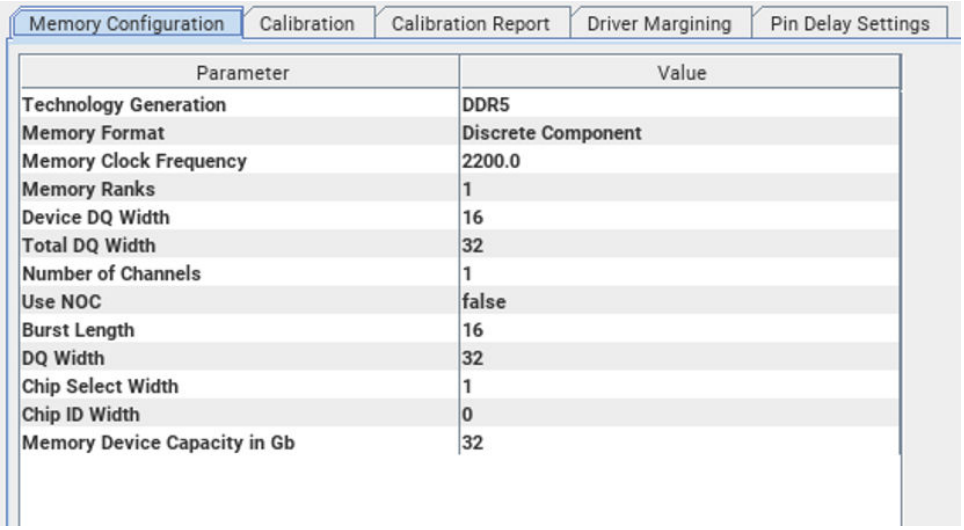
11.5.4. Using the EMIF Debug Toolkit

The main view of the EMIF Debug Toolkit contains the **Memory Configuration**, **Calibration**, **Calibration Report**, **Driver Margining**, **VREF Margining**, and **Pin Delay Settings** tabs.

Memory Configuration Tab

The **Memory Configuration** tab shows the IP settings, which you defined when you parameterized the EMIF IP.

Figure 77. Memory Configuration Tab



Parameter	Value
Technology Generation	DDR5
Memory Format	Discrete Component
Memory Clock Frequency	2200.0
Memory Ranks	1
Device DQ Width	16
Total DQ Width	32
Number of Channels	1
Use NOC	false
Burst Length	16
DQ Width	32
Chip Select Width	1
Chip ID Width	0
Memory Device Capacity in Gb	32

Calibration Tab

The **Calibration** tab allows you to rerun calibration and the test engine.

Figure 78. Calibration Tab

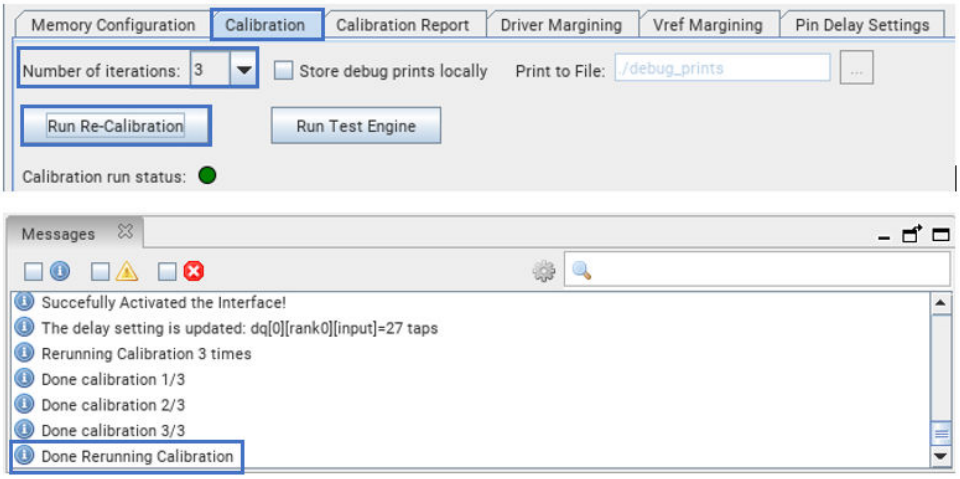
11.5.4.1. Rerunning Calibration

The **Calibration** tab lets you specify a number of iterations by which to rerun calibration.

To rerun calibration, follow these steps:

1. Select the desired number of iterations from the **Number of Iterations** pull-down menu.
2. Click **Run Re-Calibration** to repeat calibration the specified number of times. The system reports **Done Rerunning Calibration** in the **Messages** window upon completion.

   A green dot in the **Calibration run status** indicator signifies that the calibration passed for all the iterations, while a red dot indicates that the calibration failed in at least one of the iterations.



Note: For multichannel interfaces, recalibrating an interface resets all channels associated with that EMIF.

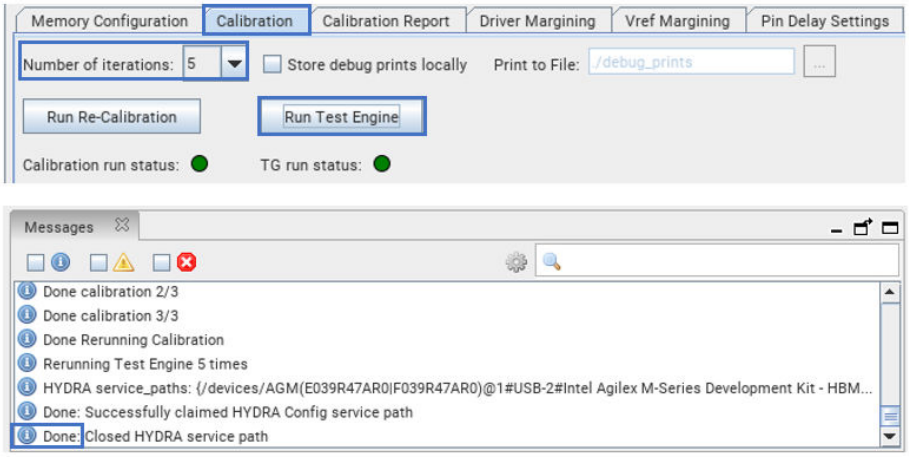
11.5.4.2. Rerunning the Test Engine

You can use the debug toolkit to rerun the test engine.

To rerun the test engine, follow these steps:

1. Select the desired number of iterations that you want to rerun the test engine from the **Number of Iterations** pulldown menu.
2. Click **Run Test Engine** to rerun the traffic generator with each iteration. A **Done** status on the **Messages** window indicates a complete run.

   A green dot in the **Calibration run status** indicator signifies a pass for all iterations of the test engine, while a red dot indicates a failure in at least one iteration.



11.5.4.3. Saving Debug Print

You can save the calibration log to a Debug Print text file.

To save a Debug Print file, follow these steps:

1. On the **Calibration** tab, enable **Store debug prints locally**.
2. In the **Print to File** field, specify a path and filename for the file that you want to create.

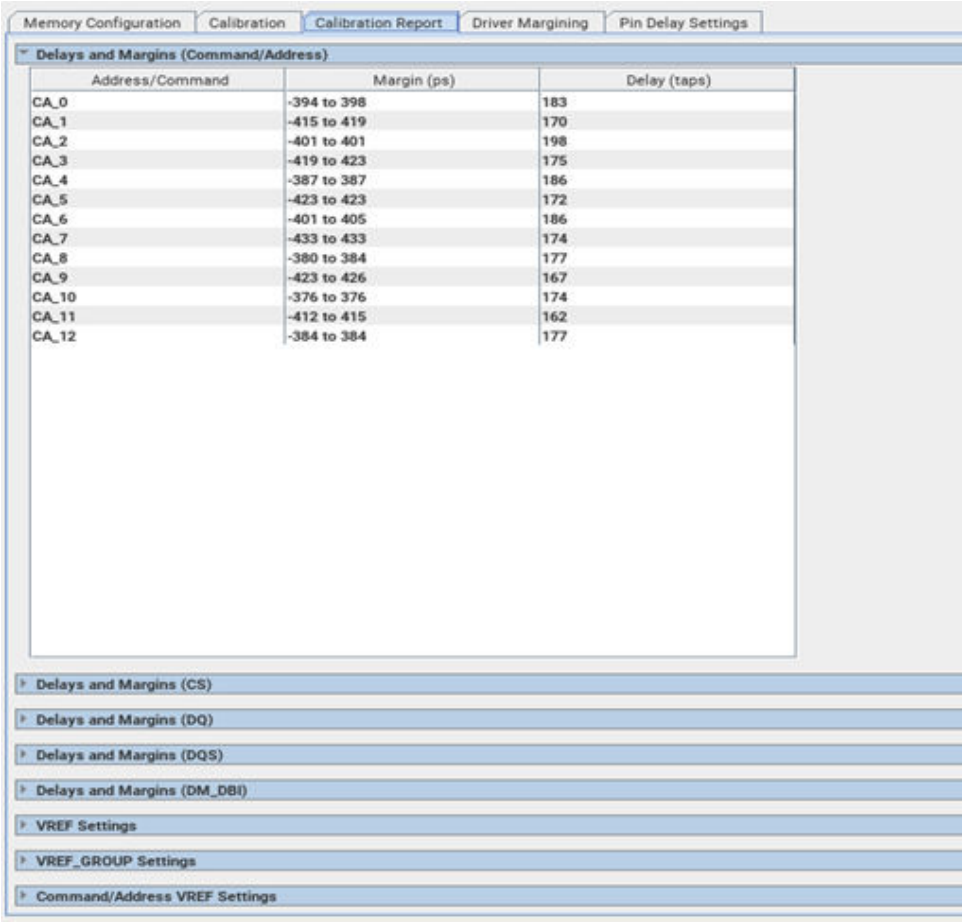
In the example illustrated in the above diagram, clicking **Run Re-Calibration** would generate 5 debug print files, named `debug_prints_1`, `debug_prints_2`, and so forth. Each file stores the debug prints information for one iteration of calibration.

11.5.4.4. Calibration Reports

Calibration reports provide detailed information about the margins observed during calibration and the settings applied on the calibration bus during calibration.

To view the margins, click on the respective section for `Address/Command`, `CS`, `DQ`, `DQS`, `DM_DBI`, `VREF Settings` or `VREF_GROUP Settings`.

Figure 79. Calibration Report Tab



11.5.4.5. Driver Margining Tab

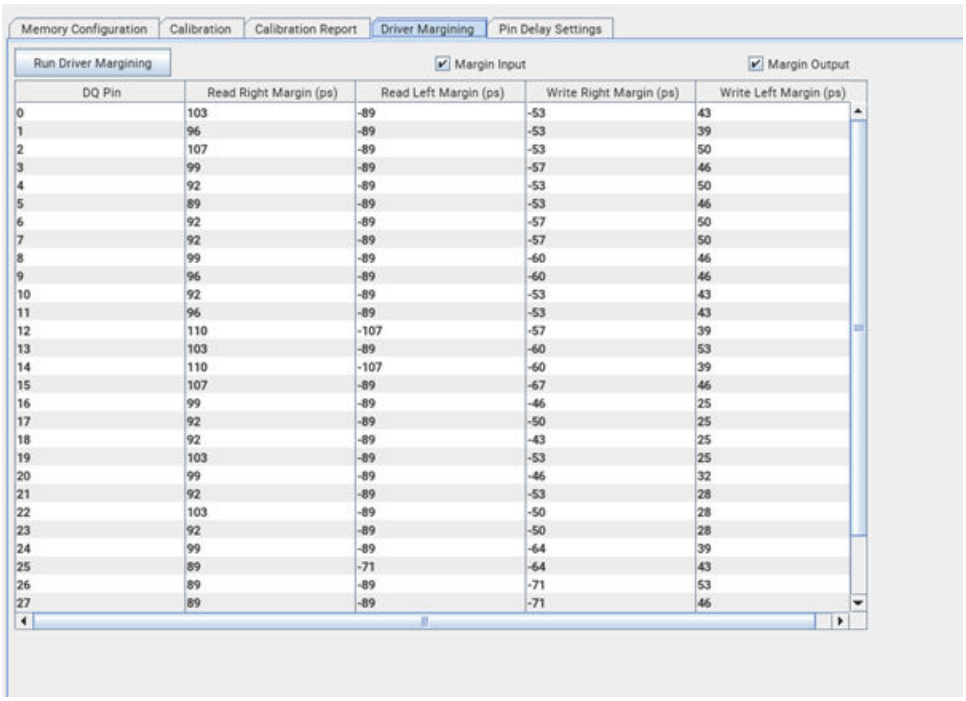
The driver margining feature lets you measure margins on your memory interface using a driver with predefined traffic patterns.

Margins measured with this feature are expected to be smaller than margins measured during calibration, because this traffic pattern is longer than those run during calibration, and is intended to stress the interface.

To use driver margining, press **Run Driver Margining**, at the top-left of the tab. The toolkit then measures margins for DQ read, DQ write, and DM. The process usually takes a few minutes, depending on the margin size, the interface size, and the duration of the driver tests.

The system displays the test results in the table when the test has completed.

Figure 80. Driver Margining Tab



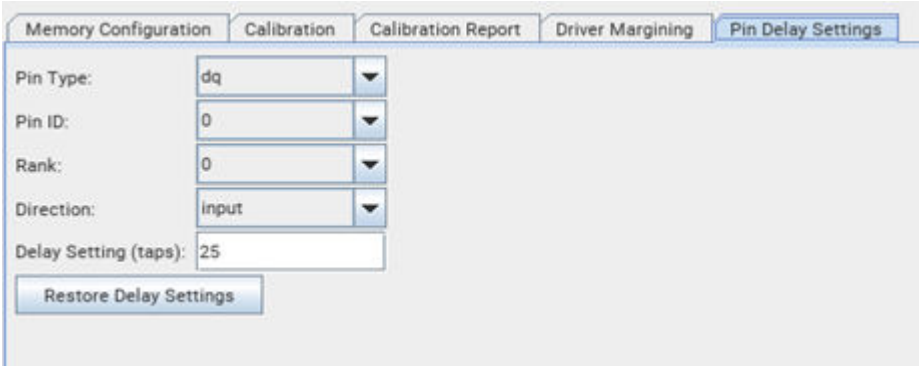
DQ Pin	Read Right Margin (ps)	Read Left Margin (ps)	Write Right Margin (ps)	Write Left Margin (ps)
0	103	-89	-53	43
1	96	-89	-53	39
2	107	-89	-53	50
3	99	-89	-57	46
4	92	-89	-53	50
5	89	-89	-53	46
6	92	-89	-57	50
7	92	-89	-57	50
8	99	-89	-60	46
9	96	-89	-60	46
10	92	-89	-53	43
11	96	-89	-53	43
12	110	-107	-57	39
13	103	-89	-60	53
14	110	-107	-60	39
15	107	-89	-67	46
16	99	-89	-46	25
17	92	-89	-50	25
18	92	-89	-43	25
19	103	-89	-53	25
20	99	-89	-46	32
21	92	-89	-53	28
22	103	-89	-50	28
23	92	-89	-50	28
24	99	-89	-64	39
25	89	-71	-64	43
26	89	-89	-71	53
27	89	-89	-71	46

11.5.4.6. Pin Delay Settings Tab

The **Pin Delay Settings** tab lets you view and change delay values on specific pins.

You can select the **Pin Type**, **Pin ID**, **Rank**, and **Direction** values of any delay that you are interested in, and the toolkit displays the delay value in the **Delay Setting (taps)** field.

Figure 81. Pin Delay Settings Tab



11.6. Generating Traffic with the Test Engine IP

Every Agilex 7 M-Series FPGA EMIF design example includes an instance of the software-driven programmable AXI traffic generator, known as the Test Engine IP.

You can view the Test Engine IP software within the following Python scripts:

- A `main.py` file that parses the `.qsys` file and selects the traffic program to run during execution.
- A `traffic_patterns.py` file that contains many different tutorial programs and functional tests that you can refer to when writing your own traffic patterns.

For the EMIF design example, the hard-coded traffic program selected when you generate a design is the `emif_tg_emulation` traffic program, which provides these features:

- Single write and read (with AxLEN=*axlen_a*[1])
- Single write and read (with AxLEN=*axlen_b*[2])
- Sequential address[3] block of 512 writes and 512 reads (with AxLEN=*axlen_a*[1])
- Sequential address[3] block of 512 writes and 512 reads (with AxLEN=*axlen_b*[2])
- Random address[4] block of 512 writes and 512 reads (with AxLEN= *axlen_a*[1])

1. The *axlen_a* value is dependent on the memory technology:
   - For DDR4: 0
   - For DDR5: 1
   - For LPDDR5: 3
2. The *axlen_b* value is dependent on the memory technology:
   - For DDR4: 0
   - For DDR5: 0 (results in Read-Modify-Write or Data-Masking on the memory side).
   - For LPDDR5: 3
3. Sequential Address pattern starts at address=0, and increments by (AXI_DATA_WIDTH/8)*(AxLEN +1) on each transaction.
4. Random Address pattern starts at address=0, and uses pseudo-random addresses.

11.7. Guidelines for Developing HDL for Traffic Generator

If you are not getting the expected response on the AXI bus when using your own traffic generator to test your EMIF IP on hardware, ensure that your traffic generator meets the following guidelines.

1. The traffic generator issues transactions only after calibration has completed successfully. You can check the calibration status by using the AXI-Lite interface. In the EMIF example design, the `cal_done_rst_n port` on the `ed_synth_axil_driver_0` corresponds to the calibration status. A value of `cal_done_rst_n=1` indicates that the calibration has completed and passed.

   Your traffic generator can begin to issue AXI-compliant transactions only after `cal_done_rst_n=1`.

Figure 82. `cal_done_rst_n` in `ed_synth_axil_driver_0`

```
149   ed_synth_axil_driver_0 axil_driver_0 (
150      .axil_driver_clk     (user_pll_outclk1_clk),
151      .axil_driver_rst_n   (~rst_controller_reset_out_reset),
152      .axil_driver_awaddr  (axil_driver_0_axil_driver_axi4_lite_awaddr),
153      .axil_driver_awvalid (axil_driver_0_axil_driver_axi4_lite_awvalid),
154      .axil_driver_awready (axil_driver_0_axil_driver_axi4_lite_awready),
155      .axil_driver_wdata   (axil_driver_0_axil_driver_axi4_lite_wdata),
156      .axil_driver_wstrb   (axil_driver_0_axil_driver_axi4_lite_wstrb),
157      .axil_driver_wvalid  (axil_driver_0_axil_driver_axi4_lite_wvalid),
158      .axil_driver_wready  (axil_driver_0_axil_driver_axi4_lite_wready),
159      .axil_driver_bresp   (axil_driver_0_axil_driver_axi4_lite_bresp),
160      .axil_driver_bvalid  (axil_driver_0_axil_driver_axi4_lite_bvalid),
161      .axil_driver_bready  (axil_driver_0_axil_driver_axi4_lite_bready),
162      .axil_driver_araddr  (axil_driver_0_axil_driver_axi4_lite_araddr),
163      .axil_driver_arvalid (axil_driver_0_axil_driver_axi4_lite_arvalid),
164      .axil_driver_arready (axil_driver_0_axil_driver_axi4_lite_arready),
165      .axil_driver_rdata   (axil_driver_0_axil_driver_axi4_lite_rdata),
166      .axil_driver_rresp   (axil_driver_0_axil_driver_axi4_lite_rresp),
167      .axil_driver_rvalid  (axil_driver_0_axil_driver_axi4_lite_rvalid),
168      .axil_driver_rready  (axil_driver_0_axil_driver_axi4_lite_rready),
169      .axil_driver_awprot  (axil_driver_0_axil_driver_axi4_lite_awprot),
170      .axil_driver_arprot  (axil_driver_0_axil_driver_axi4_lite_arprot),
171      .cal_done_rst_n      (axil_driver_0_cal_done_rst_n_reset)
172   );
```

2. Ensure that all the AXI ports on the EMIF IP are driven by registers. To prevent registers from being merged and synthesized away, add the *don't merge* and *preserve* attributes to the registers driving the AXI port in your HDL.

Figure 83. Specifying *dont_merge* and *preserve* Attributes to all Registers Driving AXI Port

```
module ed_synth_traffic_generator_kg_simple(
input  wire         driver0_axi4_awready,
output reg [43:0]   driver0_axi4_awaddr    /* synthesis dont_merge syn_preserve = 1 */ ,
output reg          driver0_axi4_awvalid   /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [6:0]    driver0_axi4_awid      /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [7:0]    driver0_axi4_awlen     /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [2:0]    driver0_axi4_awsize    /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [1:0]    driver0_axi4_awburst   /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [0:0]    driver0_axi4_awlock    /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [3:0]    driver0_axi4_awcache   /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [2:0]    driver0_axi4_awprot    /* synthesis dont_merge syn_preserve = 1 */ ,
output reg [13:0]   driver0_axi4_awuser    /* synthesis dont_merge syn_preserve = 1 */ ,
input  wire         driver0_axi4_arready,
output reg          driver0_axi4_arvalid   /* synthesis dont_merge syn_preserve = 1 */ ,
```

12. Document Revision History for External Memory Interfaces Agilex 7 M-Series FPGA IP User Guide

Document Version	Quartus Prime Version	IP Version	Changes
2024.04.01	24.1	6.1.0	• In the *Product Architecture* chapter: — Expanded the note after the table in the *LPDDR5 Pin Placement* topic. — Added two rows to the table in the *Mailbox Supported Commands* topic. — Added two new rows to the *Command Definitions* table in the *Mailbox Command Definitions* topic. • In the *Agilex 7 FPGA EMIF IP Parameter for DDR4* topic in the *DDR4 Support* chapter: — Repositioned the *Group: Example Design / Example Design* table. — Added a row to the *Group: General IP Parameters / High-Level Parameters* table. — Added the *Group: Example Design / Traffic Generator Program* table. • In the *DDR5 Support* chapter: — In the *Agilex 7 FPGA EMIF IP Parameter for DDR5 topic*: • Repositioned the *Group: Example Design / Example Design* table. • Added the *Group: Example Design / Traffic Generator Program* table. — In the *Agilex 7 FPGA EMIF Memory Device Description IP (DDR5) Parameter Descriptions* topic, added one row to the *Group: Memory Timing Parameters / Timing Parameters* table. • In the *LPDDR5 Support* chapter: — In the *Agilex 7 FPGA EMIF IP Parameter for LPDDR5* topic: • Repositioned the *Group: Example Design / Example Design* table. • Added the *Group: Example Design / Traffic Generator Program* table. — In the *Agilex 7 FPGA EMIF Memory Device Description IP (LPDDR5) Parameter Descriptions* topic: • In the *Group: High-Level Parameters* table: — Changed the *Number of Channels in Memory Package* row to *Number of Channels*. — Changed the *Number of Ranks per Channel* row to *Memory Ranks*. — Changed the *Number of Individual DRAM Components per Rank* row to *Number of Components per Rank*. • In the *Debugging* chapter: — Removed the final sentence from the *Interface Configuration Bottleneck and Efficiency Issues* topic. — In the *Debugging with the External Memory Interface Debug Toolkit* section, implemented assorted updates and figure replacements in the following topics: • *Debugging with the External Memory Interface Debug Toolkit*. • *Configuring Your Design to Use the EMIF Debug Toolkit*. • *Launching the EMIF Debug Toolkit*. • *Using the EMIF Debug Toolkit*. • *Rerunning Calibration*. • *Rerunning the Test Engine*.

Document Version	Quartus Prime Version	IP Version	Changes
2023.12.04	23.4	6.0.0	• In the *Architecture* chapter: — In the *Mailbox Command Definitions* topic, modified the *CMD_TARGET_IP_TYPE* description in the *CMD_REQ Definition* table, and added the *CMD_TARGET_IP_TYPE Definition* table. • In the *End-User Signals* chapter, updated interfaces and signals descriptions. • In the *DDR4 Support*, *DDR5 Support*, and *LPDDR5 Support* chapters, updated the parameter description sections. • In the *LPDDR5 Support* chapter: — In the *LPDDR5 Data Width Mapping* topic, modified the *Component* table. — Added the *LPDDR5 Byte Lane Swapping* topic. • In the *Debugging* chapter: — Modified the bulleted lists in the *Debugging with the External Memory Interface Debug Toolkit* topic. — Updated figures in the *Rerunning the Traffic Generator* and *Saving Debug Print* topics. — Added *Calibration Reports*, *Driver Margining Tab*, and *Pin Delay Settings Tab* topics, to the *Debugging with the External Memory Interface Debug Toolkit* section. — Added the *LPDDR5 Byte Lane Swapping* topic.
2023.10.02	23.3	5.0.0	• In the *Architecture* chapter: — Added the *Lockstep Configuration* topic. — Added information to the table in the *Mailbox Supported Commands* topic. — Added three tables to the *Mailbox Command Definitions* topic. • In the *DDR4 Support* chapter: — Added lockstep configuration information to the *DDR4 Data Width Mapping* topic. — Added lockstep configuration information to the *General Guidelines - DDR4* topic. — Added lockstep configuration information to the *DDR4 Byte Lane Swapping* topic. • In the *End-User Signals* chapter, updated the *s0_axi4 for EMIF* topic in the *DDR4 Interfaces*, *DDR5 Interfaces*, and *LPDDR5 Interfaces* sections.

continued...

Document Version	Quartus Prime Version	IP Version	Changes
			• In the *DDR5 Support* chapter, added the *Board Design Guidelines* section. • In the *LPDDR5 Support* chapter, added the *Board Design Guidelines* section. • In the *Debugging* chapter, added the *Debugging with the External Memory Interface Debug Toolkit* section.
2023.06.26	23.2	4.0.0	• In the *Architecture* chapter: — Added the *User Clock in Different Core Access Modes* topic. — Added the *Mailbox Supported Commands* and *Mailbox Command Definitions* topics. — Added the *Intel Agilex 7 M-Series EMIF IP for Hard Processor Subsystem (HPS)* topic. • In the *End-User Signals* chapter, updated the signals description topics. • In the *DDR4 Support* chapter: — Updated the parameter description topics. — Modified the *DDR4 Byte Lane Swapping* topic. • In the *DDR5 Support* chapter: — Updated the parameter description topics. — Added the *Address and Command Pin Placement for DDR5* and *DDR5 Data Width Mapping* topics. — Added the *Pin Swapping Guidelines* section. • In the *LPDDR5 Support* chapter: — Updated the parameter description topics. — Added the *Address and Command Pin Placement for LPDDR5* and *LPDDR5 Data Width Mapping* topics.
2023.04.03	23.1	3.0.0	Initial release.