External Memory Interfaces (EMIF) IP User Guide

Agilex™ 5 FPGAs and SoCs

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

IP Version: **6.1.0**

Online Version
Send Feedback

817467
2024.04.01

Contents

1. About the External Memory Interfaces Agilex™ 5 FPGA IP..... 6
1.1. Release Information.....6
2. Agilex 5 FPGA EMIF IP – Introduction.....7
2.1. Agilex 5 EMIF IP Protocol and Feature Support.....7
2.2. Agilex 5 EMIF IP Design Flow..... 7
2.2.1. Agilex 5 EMIF IP Design Checklist..... 8
3. Agilex 5 FPGA EMIF IP – Product Architecture..... 10
3.1. Agilex 5 EMIF Architecture: Protocol and Maximum Interface Width Support..... 10
3.2. Agilex 5 EMIF Architecture: Introduction..... 11
3.2.1. Agilex 5 EMIF Architecture: I/O Subsystem..... 12
3.2.2. Agilex 5 EMIF Architecture: I/O SSM.....13
3.2.3. Agilex 5 EMIF Architecture: HSIO Bank..... 13
3.2.4. Agilex 5 EMIF Architecture: I/O Lane..... 23
3.2.5. Agilex 5 EMIF Architecture: Input DQS Clock Tree..... 26
3.2.6. Agilex 5 EMIF Architecture: PHY Clock Tree.....27
3.2.7. Agilex 5 EMIF Architecture: PLL Reference Clock Networks.....27
3.2.8. Agilex 5 EMIF Architecture: Clock Phase Alignment.....28
3.2.9. User Clock in Different Core Access Modes.....29
3.3. Agilex 5 EMIF Sequencer..... 30
3.3.1. Agilex 5 Mailbox Structure and Register Definitions.....31
3.4. Agilex 5 EMIF Controller.....46
3.4.1. Hard Memory Controller.....46
3.5. Agilex 5 EMIF IP for Hard Processor Subsystem (HPS)..... 47
4. Agilex 5 FPGA EMIF IP – End-User Signals..... 50
4.1. Agilex 5 FPGA EMIF IP Interfaces for DDR4.....50
4.1.1. ref_clk for EMIF 50
4.1.2. core_init_n for EMIF 50
4.1.3. usr_async_clk for EMIF 51
4.1.4. usr_clk for EMIF 51
4.1.5. usr_rst_n for EMIF 51
4.1.6. s0_axi4 for EMIF.....51
4.1.7. mem for EMIF 54
4.1.8. oct for EMIF 55
4.2. Agilex 5 FPGA EMIF IP Interfaces for LPDDR4..... 55
4.2.1. ref_clk for EMIF 55
4.2.2. core_init_n for EMIF 55
4.2.3. usr_async_clk for EMIF 56
4.2.4. usr_clk for EMIF 56
4.2.5. usr_rst_n for EMIF 56
4.2.6. s0_axi4 for EMIF.....56
4.2.7. mem for EMIF 59
4.2.8. oct for EMIF 59
4.3. Agilex 5 FPGA EMIF IP Interfaces for LPDDR5..... 60
4.3.1. ref_clk for EMIF 60
4.3.2. core_init_n for EMIF 60

4.3.3. usr_async_clk for EMIF 60
4.3.4. usr_clk for EMIF 61
4.3.5. usr_rst_n for EMIF 61
4.3.6. s0_axi4 for EMIF 61
4.3.7. oct for EMIF 64
4.4. Agilex 5 FPGA EMIF IP Interfaces for EMIF Calibration Component 64
4.4.1. s0_axi4lite_clk for EMIF 64
4.4.2. s0_axi4lite_rst_n for EMIF 65
4.4.3. s0_axil for EMIF 65

5. Agilex 5 FPGA EMIF IP – Simulating Memory IP 66
5.1. Simulation Walkthrough 66
5.1.1. Calibration 67
5.1.2. Simulation Scripts 67
5.1.3. Functional Simulation with Verilog HDL 67
5.1.4. Simulating the Design Example 68

6. Intel Agilex 5 FPGA EMIF IP - DDR4 Support 71
6.1. Intel Agilex 5 FPGA EMIF IP Parameters for DDR4 71
6.1.1. Agilex 5 FPGA EMIF IP Parameter for DDR4 71
6.1.2. Intel Agilex 5 FPGA EMIF Memory Device Description IP (DDR4) Parameter Descriptions 76
6.2. Agilex 5 FPGA EMIF IP Pin and Resource Planning 81
6.2.1. Intel Agilex 5 FPGA EMIF IP Resources 82
6.2.2. Pin Guidelines for Intel Agilex 5 FPGA EMIF IP 82
6.2.3. Pin Placements for Intel Agilex 5 FPGA DDR4 EMIF IP 84
6.3. Agilex 5 EMIF Pin Swapping Guidelines 92
6.3.1. DDR4 Byte Lane Swapping 92
6.3.2. DDR4 Address and Command and CLK Lane 93
6.3.3. DDR4 Interface x8 Data Lane 93
6.3.4. DDR4 Interface x4 Data Lane 94
6.4. DDR4 Layout Design Guidelines 95
6.4.1. DDR4 PCB Stackup and Design Considerations 96
6.4.2. DDR4 General Design Considerations 98
6.4.3. DDR4 Routing Guidelines - Memory-Down (Discrete) Topologies 102

7. Intel Agilex 5 FPGA EMIF IP - LPDDR4 Support 109
7.1. Intel Agilex 5 FPGA EMIF IP Parameters for LPDDR4 109
7.1.1. Agilex 5 FPGA EMIF IP Parameter for LPDDR4 109
7.1.2. Intel Agilex 5 FPGA EMIF Memory Device Description IP (LPDDR4) Parameter Descriptions 114
7.2. Intel Agilex 5 FPGA EMIF IP Pin and Resource Planning 118
7.2.1. Intel Agilex 5 FPGA EMIF IP Interface Pins 119
7.2.2. Intel Agilex 5 FPGA EMIF IP Resources 121
7.2.3. Pin Guidelines for Intel Agilex 5 FPGA EMIF IP 122
7.3. LPDDR4 Layout Design Guidelines 128
7.3.1. LPDDR4 PCB Stackup and Design Considerations 128
7.3.2. LPDDR4 General Design Considerations 131
7.3.3. LPDDR4 Interface Design Guidelines 135

8. Intel Agilex 5 FPGA EMIF IP - LPDDR5 Support 139
8.1. Intel Agilex 5 FPGA EMIF IP Parameters for LPDDR5 139

8.1.1. Agilex 5 FPGA EMIF IP Parameter for LPDDR5 139
8.1.2. Intel Agilex 5 FPGA EMIF Memory Device Description IP (LPDDR5) Parameter Descriptions 145
8.2. Agilex 5 FPGA EMIF IP Pin and Resource Planning 150
8.2.1. Agilex 5 FPGA EMIF IP Interface Pins 151
8.2.2. Agilex 5 FPGA EMIF IP Resources 153
8.2.3. Pin Guidelines for Agilex 5 FPGA EMIF IP 154
8.2.4. Pin Placements for Intel Agilex 5 FPGA LPDDR5 EMIF IP 157
8.3. LPDDR5 Layout Design Guidelines 160
8.3.1. LPDDR5 PCB Stackup and Design Considerations 161
8.3.2. LPDDR5 General Design Considerations 164
8.3.3. LPDDR5 Interface Design Guidelines 168

9. Agilex 5 FPGA EMIF IP – Timing Closure 174
9.1. Timing Closure 174
9.1.1. Timing Analysis 174
9.2. Optimizing Timing 175

10. Agilex 5 FPGA EMIF IP – Controller Optimization 177
10.1. Interface Standard 177
10.2. Bank Management Efficiency 178
10.3. Data Transfer 178
10.4. Improving Controller Efficiency 178
10.4.1. Frequency of Operation 178
10.4.2. Series of Reads or Writes 179

11. Agilex 5 FPGA EMIF IP – Debugging 180
11.1. Interface Configuration Performance Issues 180
11.1.1. Interface Configuration Bottleneck and Efficiency Issues 180
11.2. Functional Issue Evaluation 181
11.2.1. Intel IP Memory Model 181
11.2.2. Vendor Memory Model 182
11.2.3. Transcript Window Messages 182
11.3. Timing Issue Characteristics 184
11.3.1. Evaluating FPGA Timing Issues 184
11.3.2. Evaluating External Memory Interface Timing Issues 185
11.4. Verifying Memory IP Using the Signal Tap Logic Analyzer 186
11.5. Generating Traffic with the Test Engine IP 186
11.6. Guidelines for Developing HDL for Traffic Generator 187
11.7. Hardware Debugging Guidelines 188
11.8. Create a Simplified Design that Demonstrates the Same Issue 188
11.9. Measure Power Distribution Network 189
11.10. Measure Signal Integrity and Setup and Hold Margin 189
11.11. Vary Voltage 189
11.12. Operate at a Lower Speed 189
11.13. Determine Whether the Issue Exists in Previous Versions of Software 189
11.14. Determine Whether the Issue Exists in the Current Version of Software 190
11.15. Try A Different PCB 190
11.16. Try Other Configurations 191
11.17. Debugging Checklist 191
11.18. Categorizing Hardware Issues 192
11.19. Signal Integrity Issues 192

11.20. Characteristics of Signal Integrity Issues.....192
11.21. Evaluating Signal Integrity Issues.....192
11.22. Skew.....192
11.23. Crosstalk.....193
11.24. Power System.....193
11.25. Clock Signals.....193
11.26. Address and Command Signals.....193
11.27. Read Data Valid Window and Eye Diagram.....193
11.28. Write Data Valid Window and Eye Diagram.....194
11.29. Hardware and Calibration Issues.....194
11.30. Memory Timing Parameter Evaluation.....194
11.31. Verify that the Board Has the Correct Memory Component or DIMM Installed.....195

12. Document Revision History for External Memory Interfaces (EMIF) IP User Guide... 196

1. About the External Memory Interfaces Agilex™ 5 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.

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.

2. Agilex 5 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 5 FPGA IP* (referred to hereafter as the *Agilex 5 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 5 EMIF IP Protocol and Feature Support

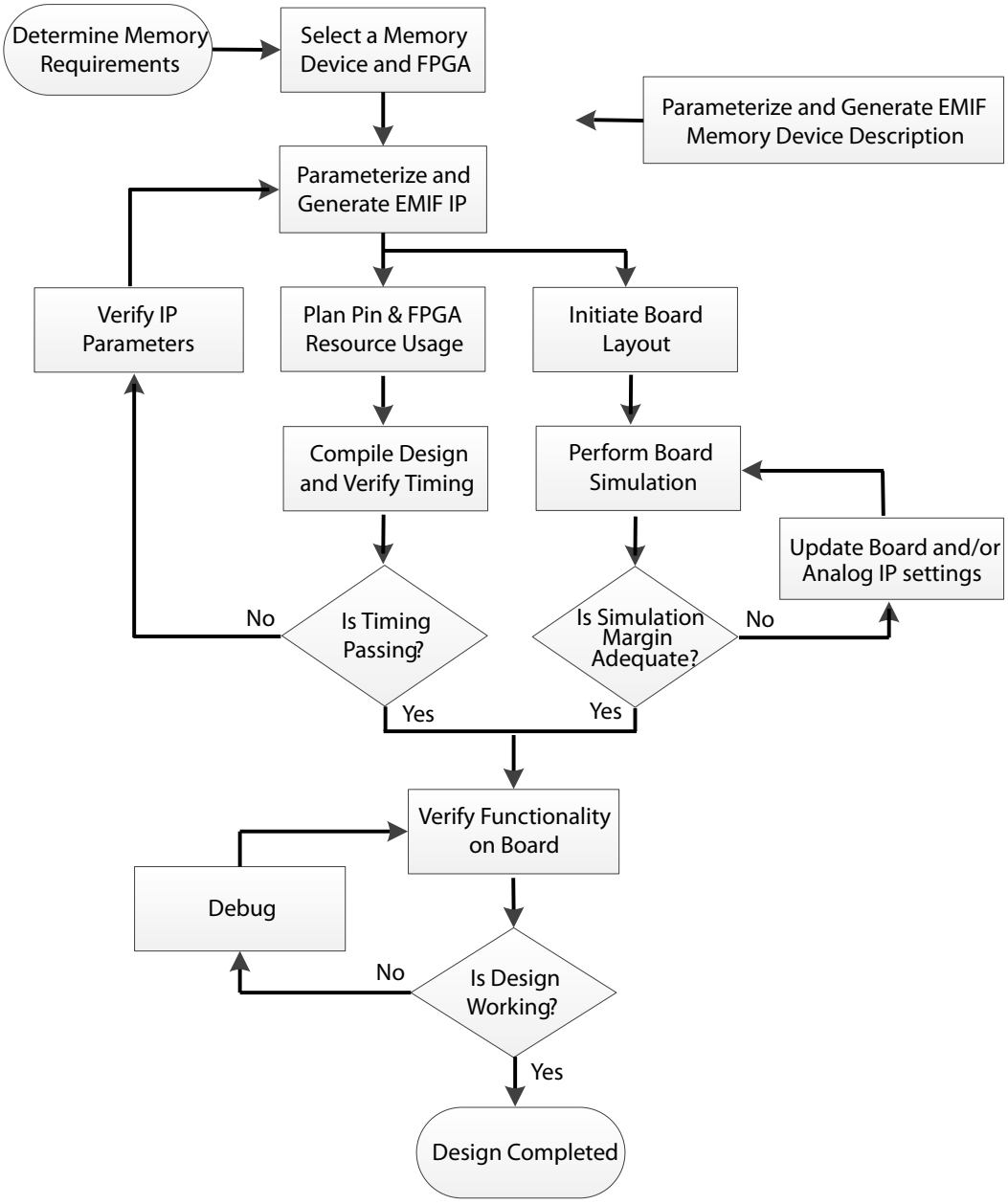
- The Agilex 5 FPGA EMIF IP supports DDR4 with hard memory controller and hard PHY.
- The Agilex 5 FPGA EMIF IP supports LPDDR4 with hard memory controller and hard PHY.
- The Agilex 5 FPGA EMIF IP supports LPDDR5 with hard memory controller and hard PHY.

2.2. Agilex 5 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



2.2.1. Agilex 5 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 • LPDDR4 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 (EMIF) IP Design Example User Guide: Agilex 5 FPGAs and SoCs
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 (EMIF) IP Design Example User Guide: Agilex 5 FPGAs and SoCs • Simulating Memory IP
Make pin assignments	For guidance on pin placement, refer to the resources listed in the right column.	• DDR4 Parameter Descriptions • LPDDR4 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.	• 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 (EMIF) IP Design Example User Guide: Agilex 5 FPGAs and SoCs
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 5 FPGA EMIF IP – Product Architecture

This chapter describes the Agilex 5 FPGA EMIF IP product architecture.

3.1. Agilex 5 EMIF Architecture: Protocol and Maximum Interface Width Support

The Agilex 5 FPGA family consists of 2 series: E-Series and D-Series. The following table summarizes the protocol and maximum data width support for E-Series and D-series devices.

Table 3. Protocol and Data Width Support for Fabric EMIF

Protocol	Maximum Data Width		
	E-Series Device Group A	E-Series Device Group B	D-Series
DDR4	x32 + ECC	x32 + ECC	x72 (DIMM) x32 + ECC (Component) x40 (Component)
LPDDR4	4ch x16 2ch x16 1ch x32	4ch x16 2ch x16 1ch x32	4ch x16 2ch x16 1ch x32
DDR5	x32 + ECC	—	2 x36 (UDIMM/SODIMM) 2 x40 (RDIMM) x32 + ECC (Component)
LPDDR5	4ch x16 2ch x16 1ch x32	4ch x16 2ch x16 1ch x32	4ch x16 2ch x16 1ch x32
DIMM Support	No	No	Yes

Table 4. Protocol and Data Width Support for HPS EMIF

Protocol	Maximum Data Width		
	E-Series Device Group A	E-Series Device Group B	D-Series
DDR4	x32 + ECC	x32 + ECC	x32 + ECC
DDR5	x32 + ECC	—	x32 + ECC
LPDDR4	4ch x 16, 2ch x 16, 1ch x 32	4ch x 16, 2ch x 16, 1ch x 32	4ch x 16, 2ch x16, 1ch x 32
LPDDR5	4ch x 16, 2ch x 16, 1ch x 32	4ch x 16, 2ch x 16, 1ch x 32	4ch x 16, 2ch x 16, 1ch x 32
DIMM Support	No	No	Yes

Note:

- E-Series devices support only component interfaces; they do not support DIMMs. For E-Series, DDR5 is supported on Device Group A only.
- Agilex 5 FPGAs do not support DDR4 and DDR5 interface widths of 8.
- The current version of the Agilex 5 External Memory Interface IP supports DDR4, LPDDR4, and LPDDR5 memory protocols.
- The current version of the Quartus Prime software supports only E-Series Device Group B.
- LPDDR5 on Agilex 5 E-Series Device Group B can support only one frequency set point (FSP0).

3.2. Agilex 5 EMIF Architecture: Introduction

The Agilex 5 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 5 FPGA IP supports the DDR4, LPDDR4, and LPDDR5 memory protocols.

The following are key hardware features of the Agilex 5 EMIF architecture:

Hard Sequencer

The sequencer employs a hardened processor, and can perform memory calibration for a wide range of protocols. For Agilex 5 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 5 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 and LPDDR4 memory protocols.

High-Speed PHY Clock Tree

Dedicated high speed PHY clock networks clock the I/O buffers in Agilex 5 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.

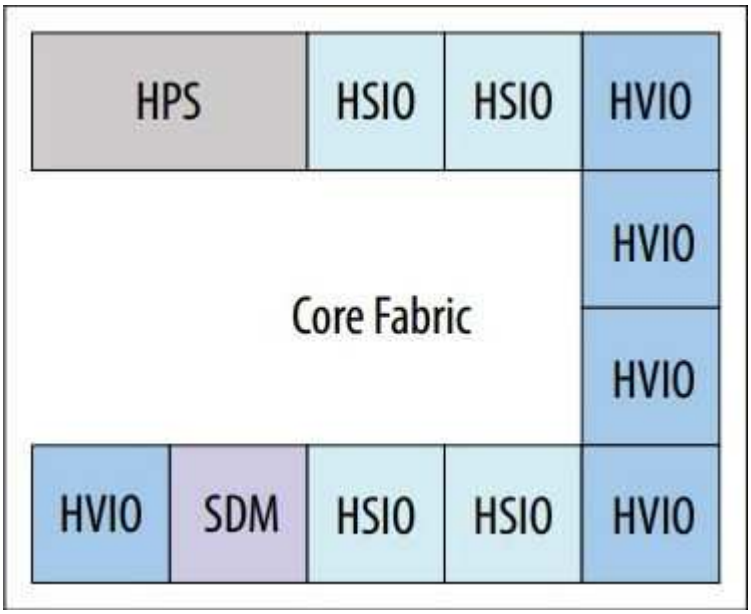
3.2.1. Agilex 5 EMIF Architecture: I/O Subsystem

In Agilex 5 devices, four types of I/O banks are available:

- High Speed I/O (HSIO)
- High Voltage I/O (HVIO)
- Hard Processor System I/O (HPS I/O)
- Secure Device Manager I/O (SDM I/O)

Only HSIO banks can support EMIF interfaces.

Figure 2. Example of Device Layout for Agilex 5 FPGAs



Note:

- The above figure does not show transceiver banks.
- HSIO and HVIO bank availability varies across device packages.

The HSIO subsystem provides the following features:

- General-purpose I/O registers and I/O buffers.
- Compensation block (comp block)
  - On-chip termination (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.

3.2.2. Agilex 5 EMIF Architecture: I/O SSM

Each HSIO 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.

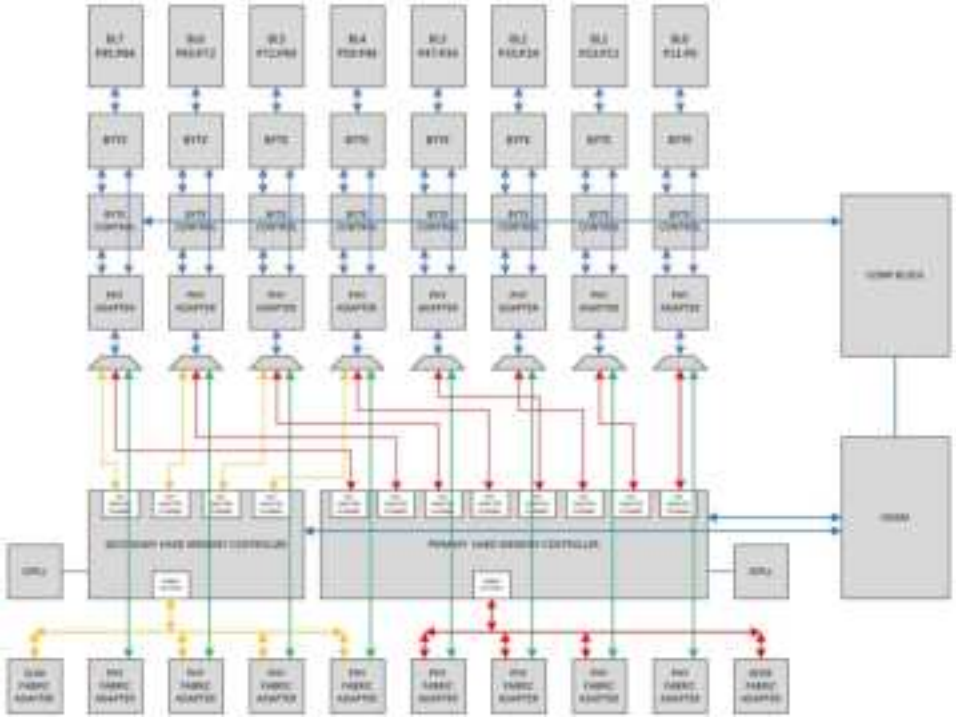
3.2.3. Agilex 5 EMIF Architecture: HSIO Bank

Each I/O row contains up to two HSIO banks; the exact number of banks depends on device size and pin package. EMIF interfaces can be implemented on HSIO banks only.

Each HSIO 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. HSIO Bank Architecture in Agilex 5 Devices



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

Agilex 5 devices have two hard memory controllers: primary and secondary. The primary hard memory controller has access to all 96 pins in an HSIO 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.

Package B18A on Agilex 5 E-Series devices has only one HSIO bank. Only pins on the top sub-bank are bonded out. Due to the limited HSIO pins available, this package does not support EMIF.

3.2.3.1. Pin Placement

Table 6. 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] *			

continued...

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
	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]		
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]

continued...

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
	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
	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]

continued...

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
	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]
	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.

Table 7. LPDDR4 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_DQS_C[3]	MEM_1_MEM_DQS_C[1]
	88	MEM_DQS_T[3]	MEM_1_MEM_DQS_T[1]
	87	MEM_DQ[27]	MEM_1_MEM_DQ[11]
	86	MEM_DQ[26]	MEM_1_MEM_DQ[10]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	85	MEM_DQ[25]	MEM_1_MEM_DQ[9]
	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_DQS_C[2]	MEM_1_MEM_DQS_C[0]
	76	MEM_DQS_T[2]	MEM_1_MEM_DQS_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		
	67		MEM_1_MEM_CK_C
	66		MEM_1_MEM_CK_T
	65		
	64		
	63		MEM_1_MEM_RESET_N
	62		OCT_1_OCT_RZQIN
	61		
	60		
BL4	59		Differential "N-side" reference clock input site
	58		Differential "P-side" reference clock input site
	57		MEM_1_MEM_CS[1]
	56		MEM_1_MEM_CS[0]
	55		MEM_1_MEM_CKE[1]
	54		MEM_1_MEM_CKE[0]
	53		MEM_1_MEM_CA[5]
	52		MEM_1_MEM_CA[4]
	51		MEM_1_MEM_CA[3]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	50		MEM_1_MEM_CA[2]
	49		MEM_1_MEM_CA[1]
	48		MEM_1_MEM_CA[0]
BL3	47		
	46		
	45		
	44		
	43	MEM_CK_C	MEM_0_MEM_CK_C
	42	MEM_CK_T	MEM_0_MEM_CK_T
	41		
	40		
	39	MEM_RESET_N	MEM_0_MEM_RESET_N
	38	RZQ Site	OCT_0_OCT_RZQIN
	37		
	36		
BL2	35	Differential "N-side" reference clock input site	
	34	Differential "P-side" reference clock input site	
	33	MEM_CS[1]	MEM_0_MEM_CS[1]
	32	MEM_CS[0]	MEM_0_MEM_CS[0]
	31	MEM_CKE[1]	MEM_0_MEM_CKE[1]
	30	MEM_CKE[0]	MEM_0_MEM_CKE[0]
	29	MEM_CA[5]	MEM_0_MEM_CA[5]
	28	MEM_CA[4]	MEM_0_MEM_CA[4]
	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_DQS_C[1]	MEM_0_MEM_DQS_C[1]
	16	MEM_DQS_T[1]	MEM_0_MEM_DQS_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_DQS_C[0]	MEM_0_MEM_DQS_C[0]
	4	MEM_DQS_T[0]	MEM_0_MEM_DQS_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]

Table 8. 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]
	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]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	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]
	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

continued...

Lane Number	Pin Index	x32	2 Channel x16
	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]
	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]

continued...

Lane Number	Pin Index	x32	2 Channel x16
	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 (EMIF) IP Design Example User Guide: Agilex 5 FPGAs and SoCs* for more information about 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.

3.2.3.2. HSIO Sub-Bank Usage

The pins in an HSIO 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.

3.2.4. Agilex 5 EMIF Architecture: I/O Lane

An HSIO 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 9. 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 T/DQS C), 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 5 device pin tables.

Table 10. 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 4. x4 Group

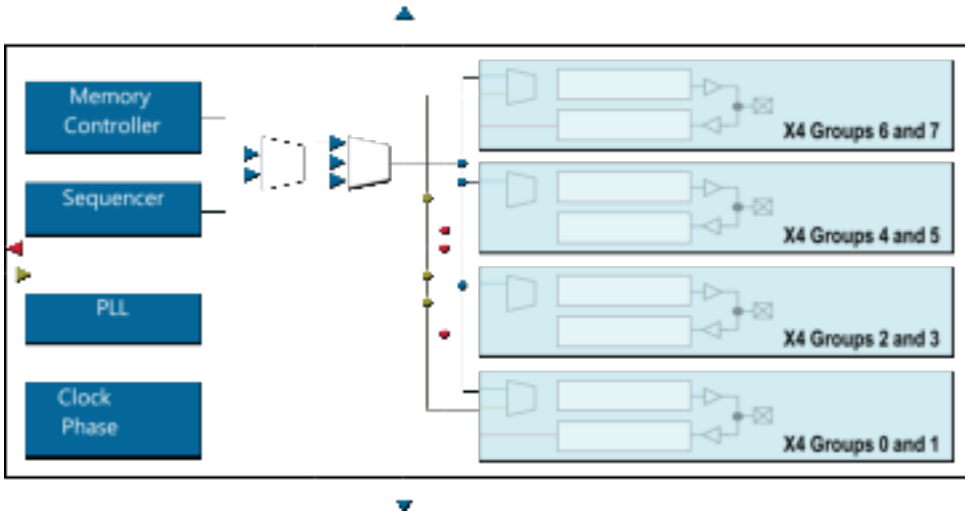


Figure 5. x8 Group

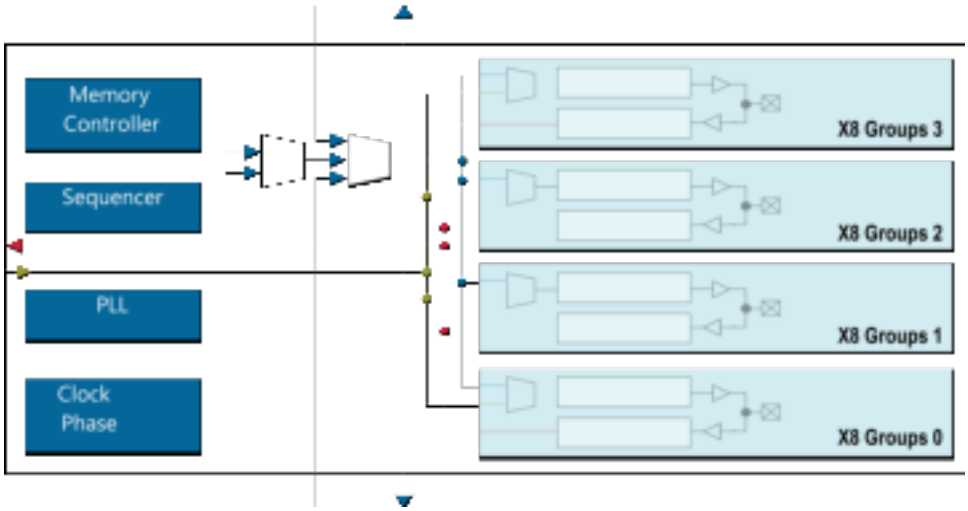
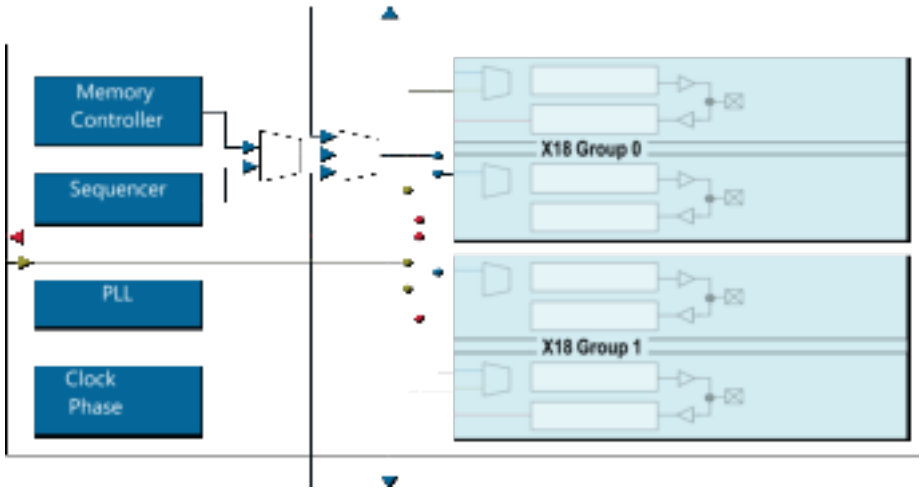


Figure 6. x18 Group



3.2.5. Agilex 5 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 11. 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.

3.2.6. Agilex 5 EMIF Architecture: PHY Clock Tree

Dedicated high-speed clock networks drive I/Os in the Agilex 5 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 5 devices can run as fast as 1.6 GHz. All Agilex 5 external memory interfaces use the PHY clock trees.

3.2.7. Agilex 5 EMIF Architecture: PLL Reference Clock Networks

Each HSIO 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.

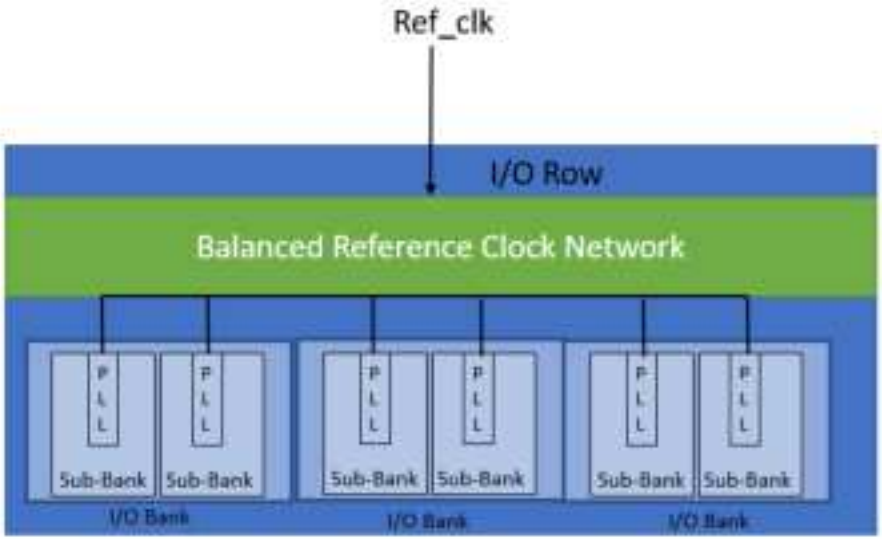
The PLL reference clock must be constrained to the address and command sub-bank only.

Agilex 5 external memory interfaces that span multiple HSIO banks use the PLL in each bank. The Agilex 5 architecture allows for relatively short PHY clock networks, reducing jitter and duty-cycle distortion.

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

- A single PLL reference clock source feeds all HSIO 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 7. PLL Balanced Reference Clock Tree



3.2.8. Agilex 5 EMIF Architecture: Clock Phase Alignment

In Agilex 5 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 5 external memory interfaces employ clock phase alignment circuitry.

Figure 8. Clock Phase Alignment Illustration

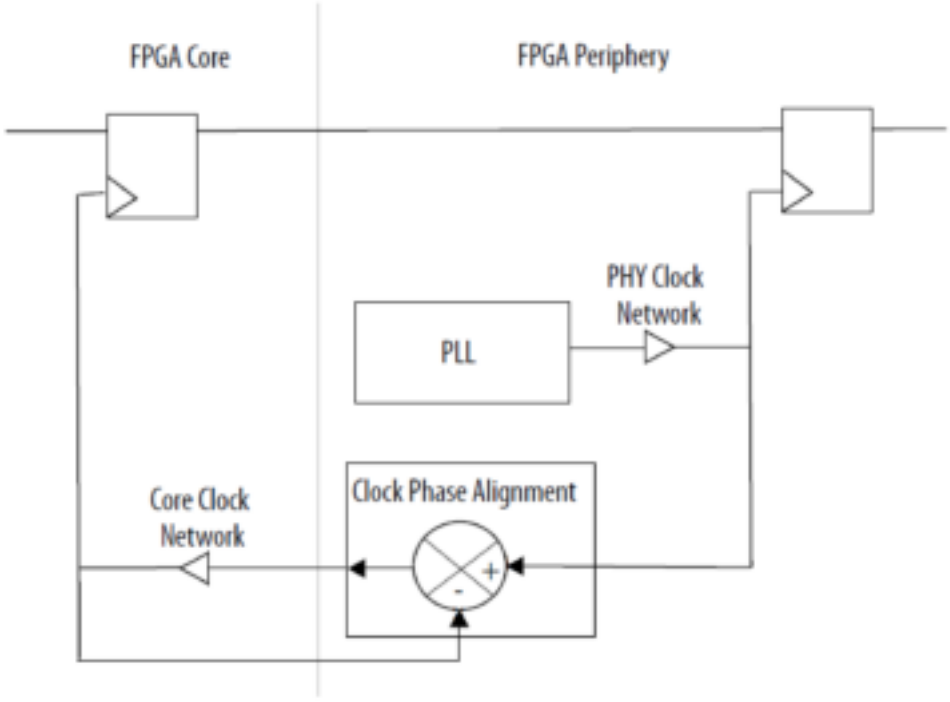
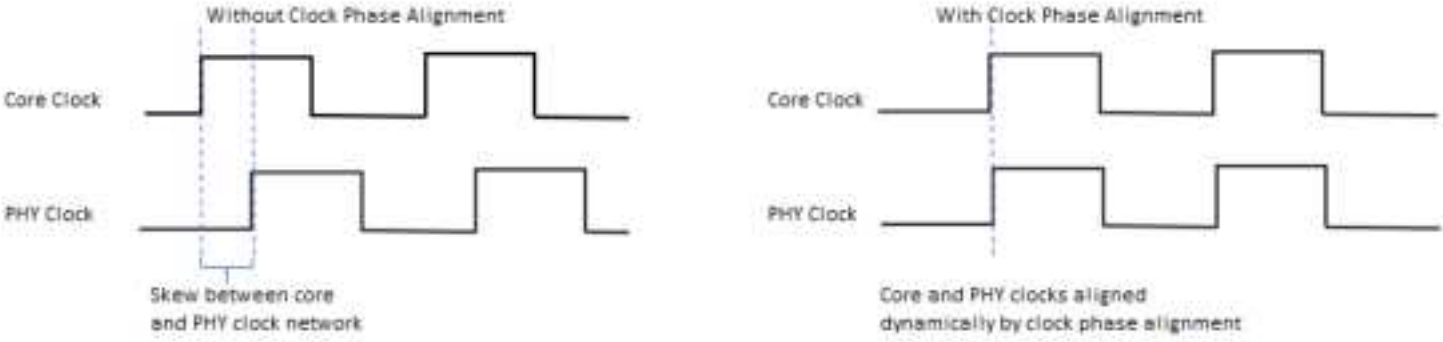


Figure 9. Effect of Clock Phase Alignment



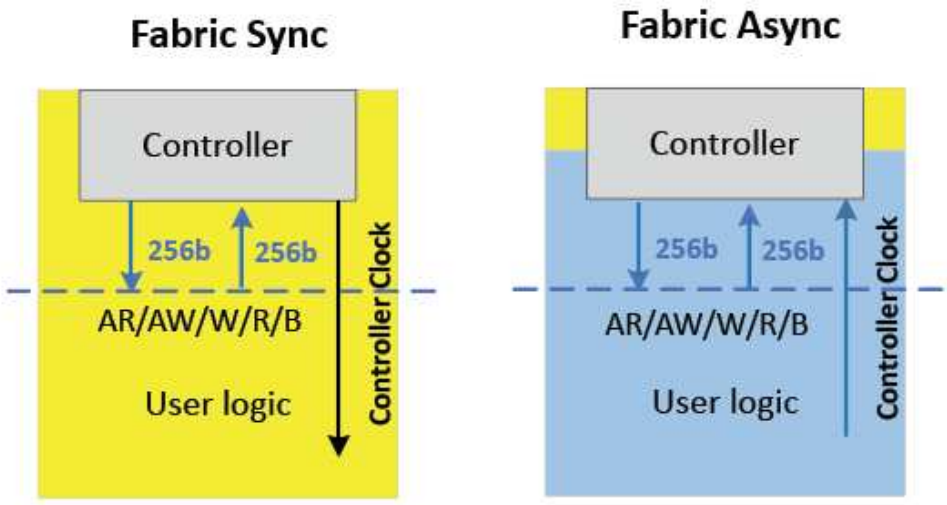
3.2.9. User Clock in Different Core Access Modes

The EMIF IP for Agilex 5 devices supports two 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 300 MHz.
  - In DDR4, the user clock frequency will be one-quarter of the memory clock frequency ((mem_CK)/4).
  - In DDR5, LPDDR5, and LPDDR4, 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).

The following figures illustrate the different clocking styles available for the Agilex 5 EMIF IP.

Figure 10. Access Modes



Benefits of Each Access Mode

- Synchronous fabric clocking is required for DDR4 DIMM.
- Asynchronous fabric access mode has the lowest latency.
- Asynchronous fabric access mode can achieve higher memory clock frequency in some speed grade / protocol combinations.

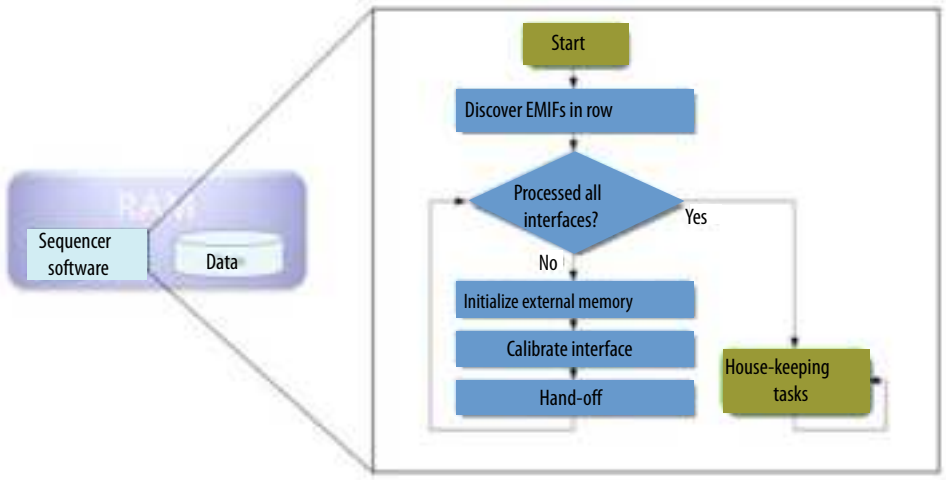
3.3. Agilex 5 EMIF Sequencer

The Agilex 5 EMIF sequencer is fully hardened in silicon, with executable code to handle protocols and topologies. Hardened RAM contains the calibration algorithm.

The Agilex 5 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 11. Agilex 5 EMIF Sequencer Operation



3.3.1. Agilex 5 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 definition.

Table 12. 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.
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.

continued...

Register Name	Byte Offset	Width (bits)	Access	Description
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. Do 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. 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 (that is, Is the response data ready?). CMD_RESPONSE_DATA_SHORT field in this register can be used for 16-bit response data.

Table 13. 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

3.3.1.1. Mailbox Supported Commands

Table 14. 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
	ECC_INTERRUPT_MASK	0x0105
	ECC_GET_SBE_INFO	0x0107
	ECC_GET_DBE_INFO	0x0108
	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

continued...

CMD_TYPE Enum	CMD_OPCODE Enum	Value
	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

3.3.1.2. Mailbox Command Definitions

Table 16. 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 17. CMD_REQ Definition

Bit	Name	Description	Access	Reset
[31:29]	CMD_TARGET_IP_TYPE	Indicates the type of IP, as follows: • 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.	Read-Write	0x0
[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 18. 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

continued...

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
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 19. 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 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.

continued...

CMD_REQ	Description
	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. '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.

continued...

CMD_REQ	Description
	'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. 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.

continued...

CMD_REQ	Description
	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. 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_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.

continued...

CMD_REQ	Description
	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
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 is 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

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_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 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 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 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**	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

continued...

CMD_REQ	Description
CMD_REQ [15:0]: CMD_OPCODE = **ECC_SCRUB_MODE_1_START**	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 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 = **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 encompass a minimum of 2 bursts. Only used if BIST_FULL_MEM is 'b0.

continued...

CMD_REQ	Description
	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** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_RESULTS_STATUS**	Command to get BIST results 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 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 is 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.

continued...

CMD_REQ	Description
	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 '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>	Command to get BIST memory initialization status for the previously initiated BIST operation for memory interface specified using the instance ID [Inputs] N/A

continued...

CMD_REQ	Description
CMD_REQ [23:16]: CMD_TYPE = **CMD_TRIG_CONTROLLER_OP** *CMD_REQ [15:0]:* CMD_OPCODE = **BIST_MEM_INIT_STATUS**	[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_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_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] 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 enters '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)

continued...

CMD_REQ	Description
	'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** *CMD_REQ [15:0]:* CMD_OPCODE = **LP_MODE_STATUS**	Command to get the Current Low Power State of the Interface. [Inputs] N/A [Outputs] 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

3.4. Agilex 5 EMIF Controller

3.4.1. Hard Memory Controller

The Agilex 5 hard memory controller is designed for high speed, high performance, high flexibility, and area efficiency. The Agilex 5 hard memory controller supports the DDR4 and LPDDR4 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).

3.4.1.1. Hard Memory Controller Features

Table 20. Features of the Agilex 5 Hard Memory Controller

Feature	Description
Memory standards support	Supports DDR4, LPDDR4, DDR5, and LPDDR5 SDRAM.
Interface protocols support	• Supports the AXI4 interface.
Burst devices support	Supports the following memory devices: • Discrete (DDR4,LPDDR4,DDR5,LPDDR5) • UDIMM (DDR4, DDR5) • SODIMM (DDR4,DDR5) • RDIMM (DDR4,DDR5)
Burst length support	• DDR4: BL8 • DDR5: BL16 • LPDDR4: BL16 • LPDDR5: BL16

continued...

Feature	Description
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.
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.

3.5. Agilex 5 EMIF IP for Hard Processor Subsystem (HPS)

The Agilex 5 FPGA EMIF IP can enable the Agilex 5 FPGA hard processor subsystem (HPS) to access external DRAM memory devices.

To enable connectivity between the HPS and the Agilex 5 EMIF IP, you must create and configure an instance of the EMIF for HPS IP, and connect it to the Agilex 5 FPGA hard processor subsystem instance in your system.

Restrictions on I/O Bank Usage for Agilex 5 EMIF IP with HPS

The following restrictions apply to the I/O bank usage:

- Only the two IO96 banks adjacent to the HPS MPFE can be used for HPS-EMIF. (Banks 3A and 3B.)
- If only one IO96 bank is to be used by HPS-EMIF, it must be the one adjacent to the HPS MPFE. (Bank 3A.)
- 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 must be left unconnected; you cannot use them as general-purpose I/O pins.
- HPS-EMIF and AVSTx16/x32 configuration mode cannot be used simultaneously, because both use bank 3A.
- 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.

Table 22. HPS EMIF Mapping

Protocol	Channel	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR4	Single	x16	–	–	–	DQ[1]	AC2	AC1	AC0	DQ[0]
	Single	x16+ECC	–	–	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
	Single	x32	–	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	Single	x32+ECC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	Single	x64	Not supported							
	Single	x64+ECC	Not supported							
DDR5	Single	x16	–	–	–	–	AC1	AC0	DQ[0]	DQ[1]
	Single	x16	DQ[1]	DQ[0]	AC1	AC0	–	–	–	–
	Dual	x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[0]	DQ[1]
	Single	x16+ECC	–	–	–	DQ[ECC]	AC1	AC0	DQ[0]	DQ[1]
	Single	x32	–	–	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	Single	x32+ECC	–	DQ[ECC]	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
LPDDR4	Single	x16	–	–	–	–	AC1	AC0	DQ[1]	DQ[0]
	Single	x16	DQ[1]	DQ[0]	AC1	AC0	–	–	–	–
	Dual	x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	Single	x32	DQ[3]	DQ[2]	–	–	AC1	AC0	DQ[1]	DQ[0]

continued...

Protocol	Channel	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
	Quad [1]	x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	[1] Quad x16 spans two IO96 banks adjacent to the HPS MPFE.									
LPDDR5	Single	x16	–	–	–	–	AC1	AC0	DQ[1]	DQ[0]
	Single	x16	DQ[1]	DQ[0]	AC1	AC0	–	–	–	–
	Dual	x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	Single	x32	DQ[3]	DQ[2]	–	–	AC1	AC0	DQ[1]	DQ[0]
	Quad [2]	x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	[2] Quad x16 spans two IO96 banks adjacent to the HPS MPFE.									

4. Agilex 5 FPGA EMIF IP – End-User Signals

The following sections describe each of the interfaces and their signals, by protocol, for the Agilex 5 EMIF IP.

4.1. Agilex 5 FPGA EMIF IP Interfaces for DDR4

The interfaces in the Agilex 5 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 23. 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 24. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	input	PLL reference clock input

4.1.2. core_init_n for EMIF

An input to indicate that core configuration is complete

Table 25. 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 26. 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 27. 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 28. 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 29. 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 30. 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 31. Interface: oct

Interface type: conduit

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

4.2. Agilex 5 FPGA EMIF IP Interfaces for LPDDR4

The interfaces in the Agilex 5 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 32. 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

4.2.1. ref_clk for EMIF

PLL reference clock input

Table 33. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	input	PLL reference clock input

4.2.2. core_init_n for EMIF

An input to indicate that core configuration is complete

Table 34. 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 35. 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 36. 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 37. 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 38. 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.

4.2.7. mem for EMIF

Interface between FPGA and external memory

Table 39. 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_reset_n	output	Asynchronous Reset
mem_cs	output	Chip Select
mem_ca	output	Command/Address Bus
mem_dq	bidir	Data (read/write)
mem_dqs_t	bidir	Data Strobe (true)
mem_dqs_c	bidir	Data Strobe (complement)
mem_dmi	bidir	Data Mask/Data Inversion

4.2.8. oct for EMIF

On-Chip Termination (OCT) interface

Table 40. Interface: oct

Interface type: conduit

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

4.3. Agilex 5 FPGA EMIF IP Interfaces for LPDDR5

The interfaces in the Agilex 5 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 41. 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
oct	conduit	On-Chip Termination (OCT) interface

4.3.1. ref_clk for EMIF

PLL reference clock input

Table 42. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	input	PLL reference clock input

4.3.2. core_init_n for EMIF

An input to indicate that core configuration is complete

Table 43. 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 44. 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 45. 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 46. 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 47. 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.

continued...

Port Name	Direction	Description
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		
s0_axi4_araddr	input	Read address.
s0_axi4_arburst	input	Read burst type.

continued...

Port Name	Direction	Description
		• '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. oct for EMIF

On-Chip Termination (OCT) interface

Table 48. Interface: oct

Interface type: conduit

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

4.4. Agilex 5 FPGA EMIF IP Interfaces for EMIF Calibration Component

The interfaces in the Agilex 5 EMIF IP each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 49. 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 50. 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 51. Interface: s0_axi4lite_rst_n

Interface type: reset

Port Name	Direction	Description
s0_axi4lite_rst_n	input	Axilite reset

4.4.3. s0_axil for EMIF

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

Table 52. Interface: s0_axil

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
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 5 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
  - 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 5 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 5 EMIF IP supports functional simulation through the design example using the traffic generator IP.

To perform functional simulation for an Agilex 5 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 5 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 5 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 various third party simulation tools, such as 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 run a simulation, navigate to the simulation directory `<example_design_directory>/sim/ed_sim/` and run the simulation script of your choice.

For ModelSim* SE and Siemens* EDA QuestaSim*- Intel FPGA Edition Simulators

1. At the command prompt, change the working directory to the following:

   ```
   <example_design_directory>/sim/ed_sim/mentor
   ```

2. Invoke vsim by typing:

   ```
   vsim
   ```

   The system launches a terminal window where you can run the commands described in the following steps.

3. Run the following command in the terminal window:

   ```
   source msim_setup.tcl
   ```

4. Run the following command in the terminal window:

   ```
   ld_debug
   ```

5. To select a signal to observe, right-click and select **Add Wave** from the context menu.

6. To run the simulation, type:

```
run -all
```

Upon successful completion, the simulation displays the following message:

```
Simulation stopped due to successful completion!
```

For VCS Simulator

At the command prompt, change the working directory to the following: *<example_design_directory>*/sim/ed_sim/synopsys/vcs

Non-interactive Mode

To run a simulation in non-interactive mode, proceed as follows:

1. Type the following command on a single line:

```
sh vcs_setup.sh USER_DEFINED_COMPILE_OPTIONS="" USER_DEFINED_ELAB_OPTIONS="-
xlrm\ uniq_prior_final" USER_DEFINED_SIM_OPTIONS=""
```

The system performs the simulation and displays the following message upon successful completion:

```
Simulation stopped due to successful completion!
```

Interactive Mode

To run a simulation in interactive mode, proceed as described below.

Note: If you have already generated a `simv` executable in non-interactive mode, delete the `simv` and `simv.diadir` files within the `vcs` folder.

1. Open the `vcs_setup.sh` file in an editor and add a `-debug_access+r` command, as highlighted in the figure below:

2. Compile the design example by typing:

```
sh vcs_setup.sh USER_DEFINED_ELAB_OPTIONS="-xlrm\ uniq_prior_final"
SKIP_SIM=1
```

3. To start the simulation in interactive mode, type the following command in the terminal console:

```
simv -gui&
```

6. Intel Agilex 5 FPGA EMIF IP - DDR4 Support

This chapter contains IP parameter descriptions and pin planning information for Intel Agilex 5 FPGA external memory interface IP for DDR4.

6.1. Intel Agilex 5 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 you must set it manually.

6.1.1. Agilex 5 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 53. 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)
Hydra Remote Access	Specifies whether the Hydra control and status registers are accessible via JTAG, exported to the fabric, or just disabled (Identifier: EX_DESIGN_HYDRA_REMOTE)

Table 54. 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)
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)
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. 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.

continued...

Display Name	Description
	(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 55. 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 56. Group: General IP Parameters / Advanced Parameters / PHY / Topology

Display Name	Description
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 57. 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 58. 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 59. 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 60. 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 61. 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 62. 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 (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 63. 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 64. 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 65. 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 66. 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)

6.1.2. Intel Agilex 5 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 67. Group: Configuration Save

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

Table 68. 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)
Chip ID Width	Specifies the number of chip ID pins. Only applicable to LRDIMMs or RDIMMs that use 3DS memory devices. Note: This parameter can be auto-computed. (Identifier: DDR4_MEM_DEVICE_CHIP_ID_WIDTH)
Density of Each Memory Component	Specifies the density of each memory component in Gb. (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 69. 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.

continued...

Display Name	Description
	(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 70. 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 71. 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)
Address-Command Latency Mode	Specifies whether address-command latency is supported, and if enabled, the latency in cycles. (Identifier: DDR4_MEM_DEVICE_AC_PARITY_LATENCY_MODE)

Table 72. 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. 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)

continued...

Display Name	Description
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)
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)

continued...

Display Name	Description
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. (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)

continued...

Display Name	Description
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)
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)

6.2. Agilex 5 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.

6.2.1. Intel Agilex 5 FPGA EMIF IP Resources

The Intel Agilex 5 FPGA memory interface IP uses several FPGA resources to implement the memory interface.

6.2.1.1. OCT

You require an OCT calibration block if you are using an Intel Agilex 5 FPGA OCT calibrated series, parallel, or dynamic termination for any I/O in your design. There are two OCT blocks in an HSIO 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 VCCIO_PIO voltage as the memory interface.
- The OCT calibration block uses a single fixed RZQ. You must ensure that an external termination resistor is connected to the correct pin for a given OCT block.

6.2.1.2. 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. Intel Agilex 5 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.

6.2.2. Pin Guidelines for Intel Agilex 5 FPGA EMIF IP

The Intel Agilex 5 FPGA contains HSIO banks on the top and bottom edges of the device, which can be used by external memory interfaces.

Intel Agilex 5 FPGA HSIO 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.

Intel Agilex 5 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 Intel Agilex 5 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 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 byte lanes 0, 1, 2, or 3, respectively.
- To determine whether HSIO banks are adjacent, you can refer to *Architecture: HSIO Bank* in the *Product Architecture* chapter. In general, the two sub-banks within an HSIO 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 HSIO 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).

6.2.2.1. Intel Agilex 5 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 for your external memory interface are known, you can begin planning your system.

6.2.2.1.1. Intel Agilex 5 FPGA EMIF IP Interface Pins

All HSIO banks in Intel Agilex 5 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.

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.

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 requirements.

6.2.2.1.2. 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 HSIO banks required to implement the memory interface, given that an HSIO 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.

6.2.2.1.3. 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.

Timing closure depends on device resource and routing utilization. For related information, refer to *Quartus Prime Pro Edition User Guide: Design Optimization*.

6.2.3. Pin Placements for Intel Agilex 5 FPGA DDR4 EMIF IP

Intel Agilex 5 FPGA DDR4 IP supports fixed address and command pin placement, and fixed data lanes placement.

6.2.3.1. Address and Command Pin Placement for DDR4

Table 73. 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]
	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]

continued...

Address/ Command Lane	Index Within Byte Lane	DDR4				
		Scheme 1	Scheme 1A	Scheme 2	Scheme 3	Scheme 3A
	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 5 FPGA DDR4 IP supports fixed Address and Command pin placement as shown in the preceding table. Note that E-series devices support only component interfaces.

The IP supports up to 2 ranks for the following schemes:

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

6.2.3.2. DDR4 Data Width Mapping

Agilex 5 devices do not support flexible data lanes placement. Only fixed byte lanes within the HSIO bank can be used as data lanes. The following table lists the supported address and command and data lane placements in an HSIO bank.

Table 74. 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
	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]

Note: 1. This configuration is not supported on E-Series devices. DDR4 x40 is not available in the current version of the Quartus Prime software. 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.
4. ES0 silicon supports address/command pin placement for DDR4 on the bottom sub-bank (BL0-BL3) only.

6.2.3.3. General Guidelines

Observe the following general guidelines when placing pins for your Intel Agilex 5 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 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 the *DDR4 Data Width Mapping* topic.
4. A byte lane must not be used by both address and command pins and data pins.
5. An HSIO bank cannot be used for more than one interface – meaning that two sub-banks belonging to two different EMIF interfaces are not permitted.
6. You may not share byte lanes within a sub-bank for two different interfaces; 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.
8. All address and command pins and their associated clock pins (CK_t and CK_c) must reside within a single sub-bank. Refer to the table in the *DDR4 Data Width Mapping* topic for the supported address and command and data lane placements.
9. 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 – 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.
   - AVST-32– Byte lanes 4, 5, 6, and 7 are all effectively occupied and are not usable by the external memory interface.

Note: EMIF IP pin-out requirements for the Intel Agilex 5 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.

6.2.3.4. 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: Intel Agilex 5 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.

6.2.3.5. 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 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.

6.2.3.6. Clock Signals

DDR4 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. The SDRAM data sheet specifies the following timings:

- tDQSCK is the skew between the CK_T or CK_C signals and the SDRAM-generated DQS signal.
- tDSH is the DQS falling edge from CK_T rising edge hold time.
- tDSS is the DQS falling edge from CK_T rising edge setup time.
- tDQSS is the positive DQS latching edge to CK_T rising edge.

SDRAM devices have a write requirement (tDQSS) 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_Tand CK_C 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_T, is aligned with the DQS write to satisfy tDQSS.

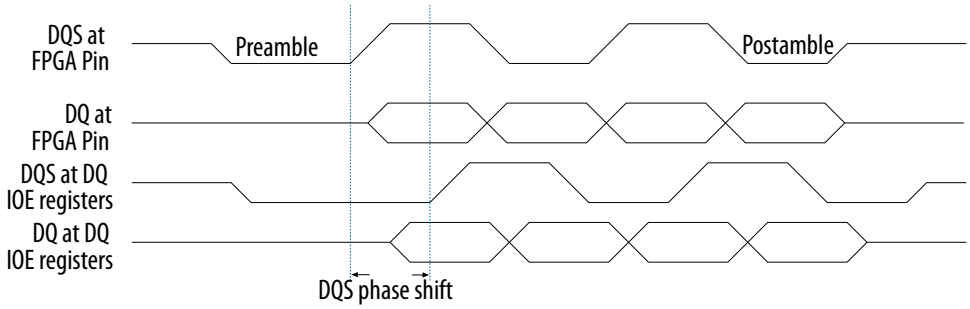
6.2.3.7. Data, Data Strobes, DM/DBI, and Optional ECC Signals

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.

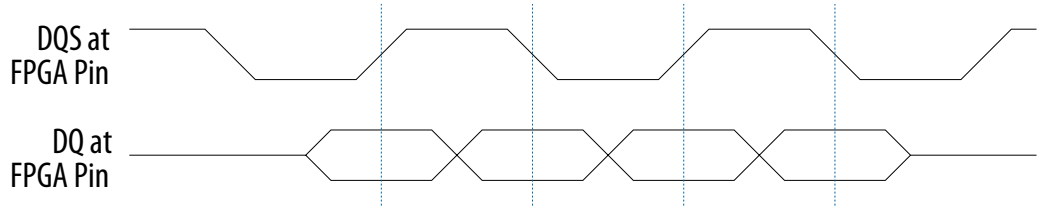
The DQ signals are edge-aligned with the DQS signal during a read from the memory and are center-aligned with the DQS signal during a write to the memory. The memory controller shifts the DQ signals by –90 degrees during a write operation to center align the DQ and DQS signals. The PHY IP delays the 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 14. 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 15. 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 5 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. Agilex 5 EMIF Pin Swapping Guidelines

In Agilex 5 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).

6.3.1. 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 lanes. The rules for swapping DQ byte lanes 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, you cannot swap the highest-indexed DQ byte 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 75. 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]	DQ3	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]

Example 1: DDR4 x 32 +ECC implemented with AC Scheme 2 using x8 memory component

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.

6.3.2. 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 CLK_T and CLK_C cannot be swapped with each other, nor can the T and C lanes for the DQS_T and DQS_C be swapped with each other.

6.3.3. 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 76. 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

continued...

Pin Index Within Byte Lane	DDR4 x8 Data Lane Function	Swap Consideration
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

6.3.4. DDR4 Interface x4 Data Lane

Agilex 5 FPGAs support only x4 components on JEDEC-compliant DIMMs. 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 77. 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	

continued...

Pin Index Within Byte Lane	DDR4 x4 Data Lane Function	Swap Consideration	
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 5 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 and DQS_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 and DQS_C on pin index 4 and pin index 5 and associated DQs are within the corresponding byte lane.
    - DQS_T and DQS_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].

6.4. DDR4 Layout Design Guidelines

This section provides PCB layout design recommendations and guidelines for Agilex 5 E-Series Group B FPGA devices with GPIO-B (Input/Output) silicon implementation.

A successful DDR design on PCB requires not only following the topology and routing guidelines provided here, but also must meet PDN design requirements. For power delivery network (PDN) design guideline information, refer to Agilex 5 Power Distribution Network Design Guidelines, available on the Intel website. For high-speed transceiver PCB layout guidelines, refer to Agilex 5 High Speed PCB Layout Design Guidelines, availabel on the Intel website.

6.4.1. DDR4 PCB Stackup and Design Considerations

The following figures show an example of a PCB stackup with 14 layers that has been used on PCB design for an Intel platform board. You may use other stackups (thin such as PCIE board or thick board) if you meet the recommendations in this guideline.

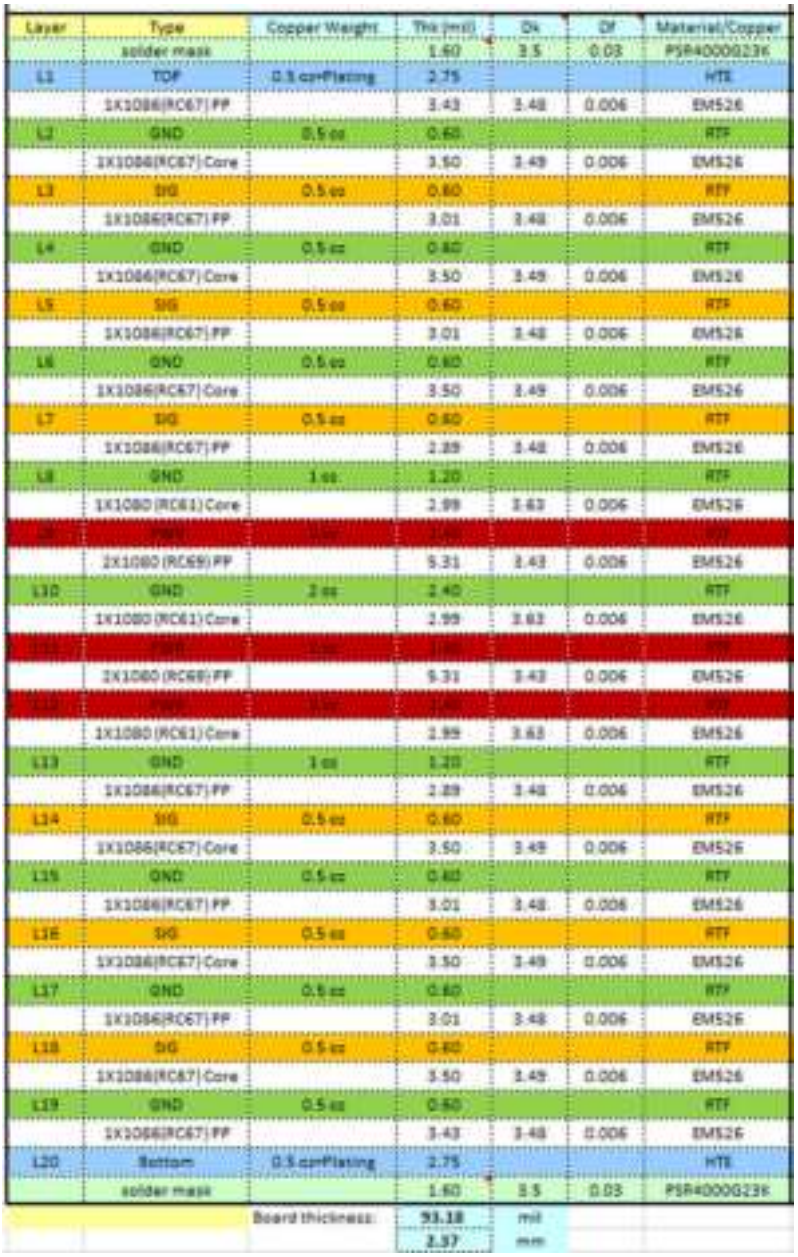
The figure below shows a 14L thin board, high performance Type-IV PCB with micro vias, stacked vias, buried vias and through vias.

Figure 16. 14L Thin Board, High Performance Type-IV PCB Stackup

Layer	Structure	Material	Copper TYPE	Process thickness (um)	Process thickness (mil)
	Soldermask	PSR4000 Series		15	0.59
1	Copper	1/3OZ+Plating	HVLP	30	1.18
	Prepreg	1x1035 RC75%		64	2.52
2	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
3	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		74	2.91
4	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
5	Copper	1/3OZ+Plating	HVLP	25	0.98
	Prepreg	1x1078 RC66%		75	2.95
6	Copper	H OZ	HVLP	15	0.59
	Core	0.130mm H/2		130	5.12
7	Copper	2 OZ	RTF	62	2.44
	Prepreg	3x1078 RC66%		200	7.87
8	Copper	2 OZ	RTF	62	2.44
	Core	0.130mm H/2		130	5.12
9	Copper	H OZ	HVLP	15	0.59
	Prepreg	1x1078 RC66%		75	2.95
10	Copper	1/3OZ+Plating	HVLP	25	0.98
	Prepreg	1x1078 RC66%		63	2.48
11	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		74	2.91
12	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
13	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1035 RC75%		65	2.56
14	Copper	1/3OZ+Plating	HVLP	30	1.18
	Soldermask	PSR4000 Series		15	0.59

The figure below shows a 20L thick Type-III Board stack-up (high performance with PTH with/without backdrill example used at Intel platform boards and development kits.

Figure 17. 20L Thick Type III Board Stackup

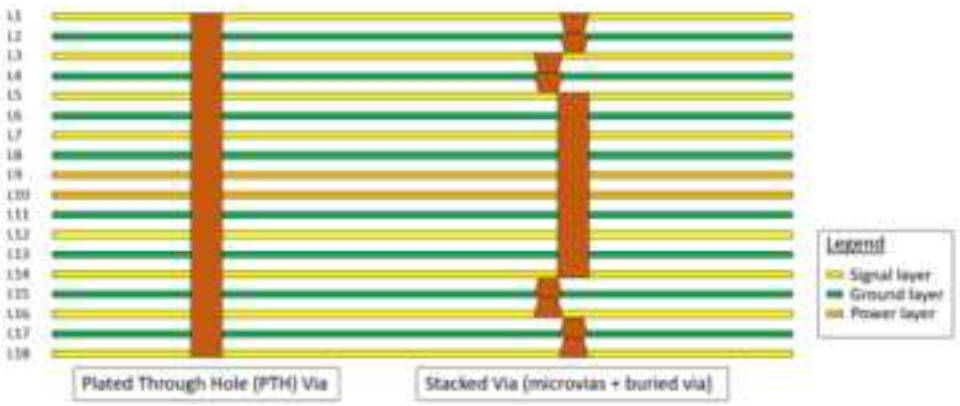


Layer	Type	Copper Weight	Thk (mil)	Dk	Df	Material/Copper
	solder mask		1.60	3.5	0.03	PSR4000G23K
L1	TOP	0.5 oz+Plating	2.75			HTE
	1X1086(RC67) PP		3.43	3.48	0.006	EM526
L2	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L3	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L4	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L5	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L6	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L7	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		2.89	3.48	0.006	EM526
L8	GND	1 oz	1.20			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L9	PWR	1 oz	1.20			RTF
	2X1080 (RC69) PP		5.31	3.43	0.006	EM526
L10	GND	2 oz	2.40			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L11	PWR	1 oz	1.20			RTF
	2X1080 (RC69) PP		5.31	3.43	0.006	EM526
L12	PWR	1 oz	1.20			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L13	GND	1 oz	1.20			RTF
	1X1086(RC67) PP		2.89	3.48	0.006	EM526
L14	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L15	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L16	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L17	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L18	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L19	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.43	3.48	0.006	EM526
L20	Bottom	0.5 oz+Plating	2.75			HTE
	solder mask		1.60	3.5	0.03	PSR4000G23K
		Board thickness:	93.18	mil		
			2.37	mm		

A type-IV PCB is a precise and high-quality PCB. This type-IV PCB utilizes not only PTH vias to connect from top to bottom layers, but also stacked vias, micro vias and buried vias to connect between layers. For example, a full-height stacked via of a 14-layer PCB is made up of a combination of dual-stacked micro vias and buried vias.

The following figure depicts a cross-sectional comparison of a PTH and a stacked via. A type-III PCB board with PTH vias, which is used to implement DDR4 designs and can also be used for LPDDR5 designs.

Figure 18. Cross-sectional Comparison Between PTH and Stacked Via



6.4.2. DDR4 General Design Considerations

General DDR Signal Routing Guideline on PCB

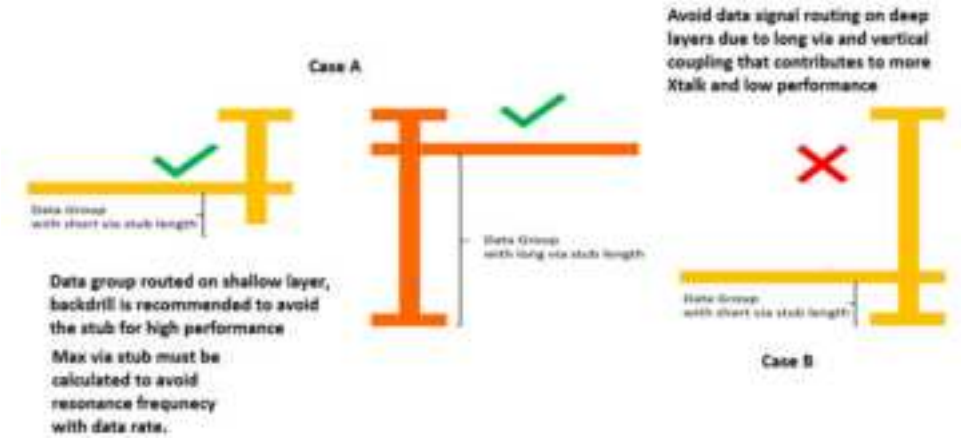
Intel recommends to route all data signals within a specific group on the same layer.

The figure below illustrates a routing example for a type-III PCB board for a DDR4 design. Data Group signals such as DQ, DM and DQS signals should be routed on shallow layers as stripline with the least Z-height via transition to avoid vertical crosstalk to achieve high performance.

For example, the recommended routing layers for data group on a 20-layers board and using PTH via will be on the top half of the PCB such as L3, L5 and L7. Other signals such as CA, CTRL and clock signals can be routed with longer Z-height via transition on the bottom half of the PCB such as L14, L16 and L18.

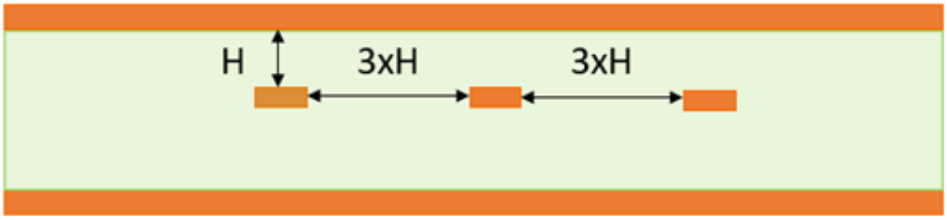
Minimal stub effect or back drill is recommended, but not mandatory, to avoid high reflection for maximum data rate performance. Long via stubs will affect the ISI of channel but the impact of ISI is less than impact of crosstalk for the max data rate performance.

Figure 19. Case A Routing is Suggested for Data Group Signals Over Case B



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

Figure 20. Minimum Trace-to-Trace Separation Distance



DDR FPGA Break Out Routing

Agilex 5 devices come with various pitch sizes for different FPGA pins. The GPIO pin pitch is very small, the device BGA pad stack is also very small, therefore, it is highly recommended to use a dog bone configuration for inner GPIO pins fanout along with stripline routing; however, for the GPIO pins on the edge of device, it is recommended to use microstrip routing. Both microstrip and stripline routing guidelines are presented in this document for all supported EMIF interfaces and topologies.

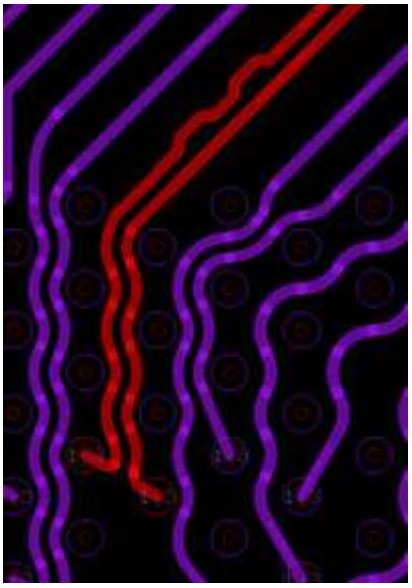
DDR Differential Signals Routing

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 area.

Intel recommends a symmetrical fan-out routing at the FPGA pin field. Non-symmetrical routing for differential signals will cause 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.

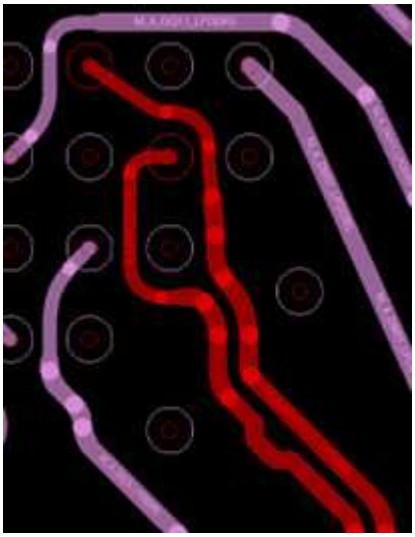
The figure below shows the symmetrical routing of differential signals (DQS/CLK) at FPGA pin field along with length/skew matching between P/N lanes right after FPGA device edge.

Figure 21. Symmetrical Routing of Differential Signals (DQS/CLK) at FPGA Pin Field



The figure below shows the single-ended routing for differential signals (DQS/CLK) at DRAM pin field if the pitch is very small along with skew matching right at edge of DRAM pin field.

Figure 22. Single-ended Routing for Differential Signals (DQS/CLK) at DRAM



Intel recommends to implement length/skew matching for differential signals (if there is) right after FPGA device to avoid additional shifting on differential signals common mode voltage.

In case of having limitation for implementing symmetrical routing at DRAM pin field for differential signals due to very small pitch, Intel recommends to route the differential signals as single-ended signals within the DRAM pin field, ensuring to keep 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/N lanes must be applied before reaching the DRAM pin field.

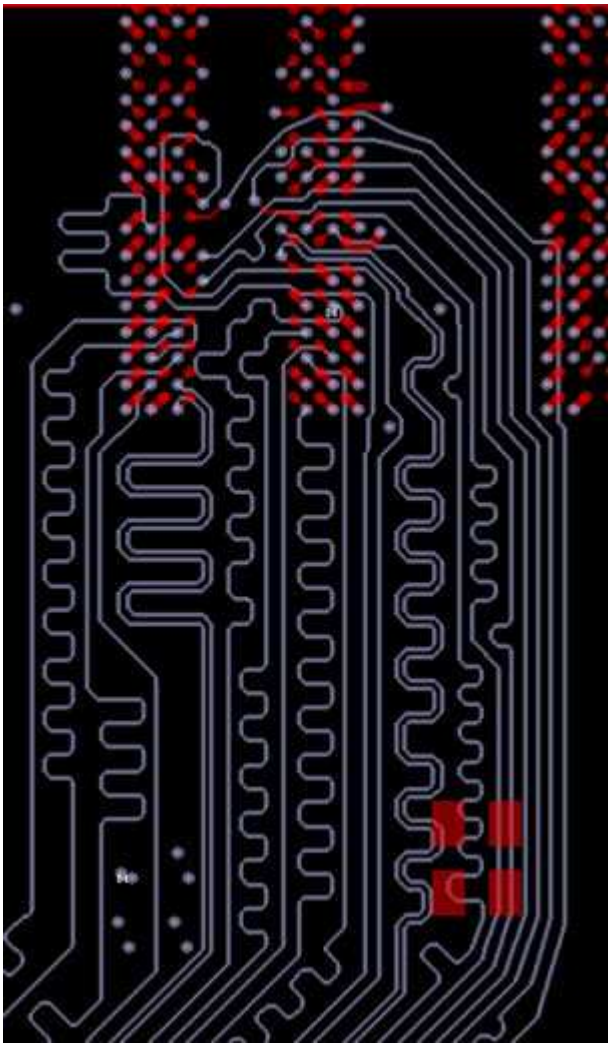
Ground Plane and Return Path

A continuous and solid ground reference plane is crucial for data lines to ensure good signal integrity performance. Low impedance ground return path from the FPGA to DRAM devices should be provisioned. In addition, it is desirable to keep ground stitching vias within 80 mils from signal transition for better return path on signal via and better signal integrity performance.

DRAM Break Out in Layout Guideline

For discrete DRAM components on PCB, you can either use the dog-bone or via in pad at DRAM for the signal transition from inner layer to DRAM. If dog-bone via transition is used, it is recommended to separate them with larger pitch to avoid crosstalk between signal vias.

Figure 23. Data Signal Group Routing on PCB for Memory Down Configuration



6.4.3. DDR4 Routing Guidelines - Memory-Down (Discrete) Topologies

Agilex 5 E-Series Group B FPGAs support only discrete DDR4 components down on mother board, in both thin and thick PCB stackups. The maximum supported data rates depend on the selected topology and thickness of board. There are two topologies for memory-down configurations covered in this section:

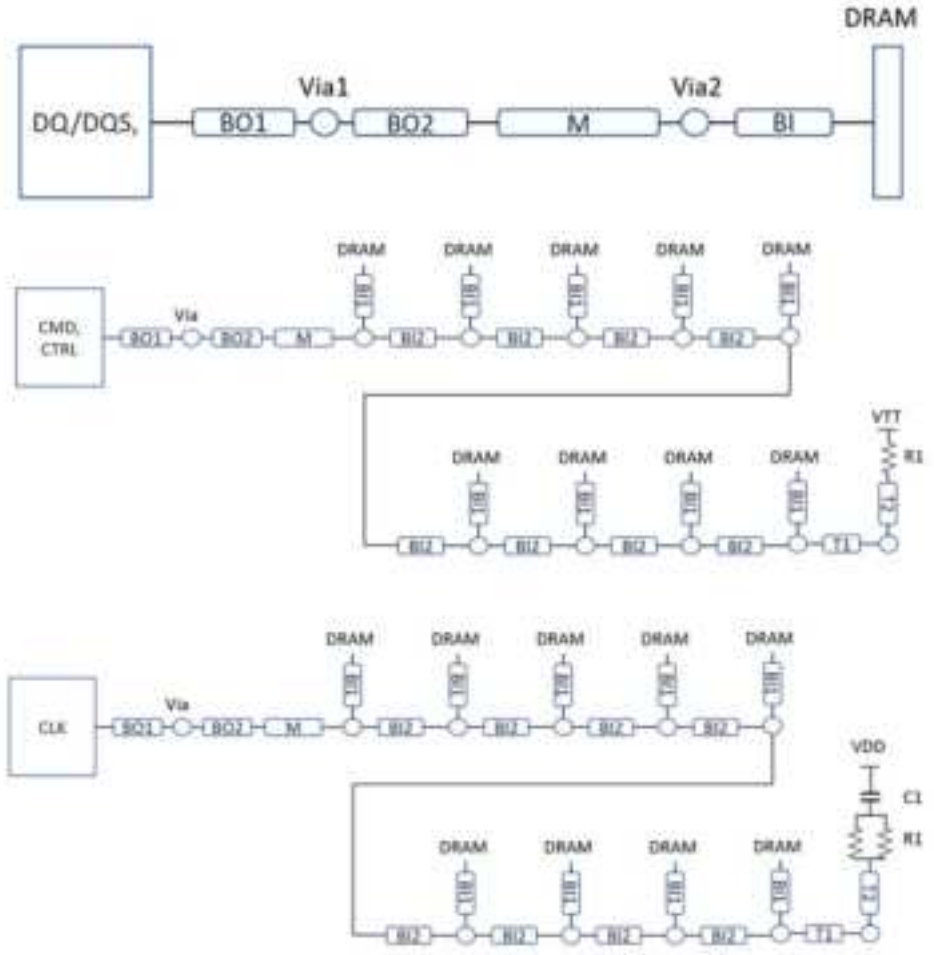
- DDR4 1 RANK x 8 interface.
- DDR4 1 Rank x 16 interface.

6.4.3.1. 1 Rank x 8 Discrete (Memory Down) Topology

A single channel with 1 rank and x8 memory devices, this interface covers data bytes (DQ/DQS), dddress signals, command signals (BA, BG, RAS, CAS, WE, ACT, PAR), control signals (CKE, CS, ODT) and clocks (CLK)s.

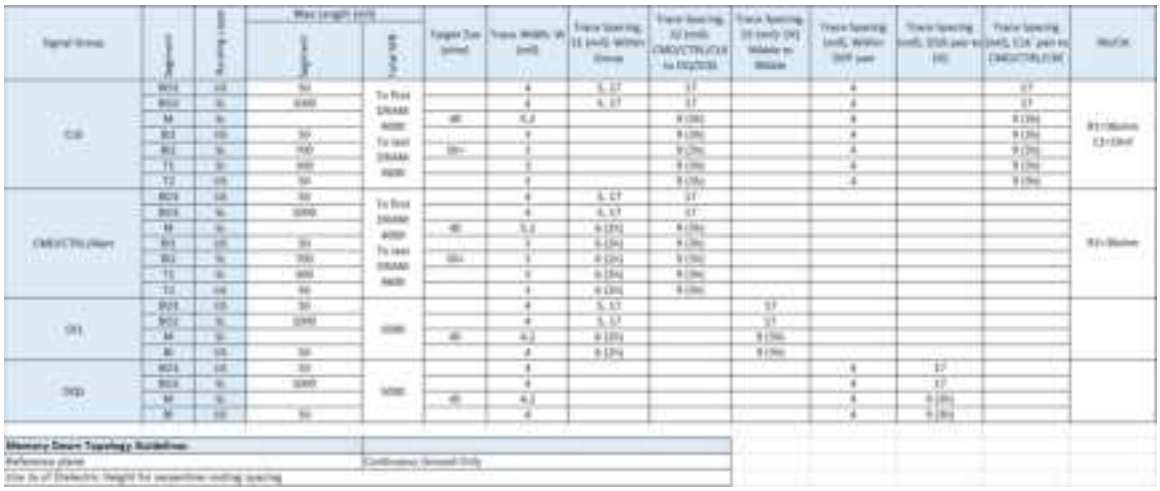
The following figure illustrates the signal connection topology for the 1 rank x 8 memory down configuration.

Figure 24. Signals Connections for Supported Signals in 1 Rank x 8 Discrete Topology



The following table shows specific routing guidelines for 1 rank x 8 discrete memory topology.

Table 78. Stripline Routing Guidelines for 1 Rank x8 Discrete Memory Topology

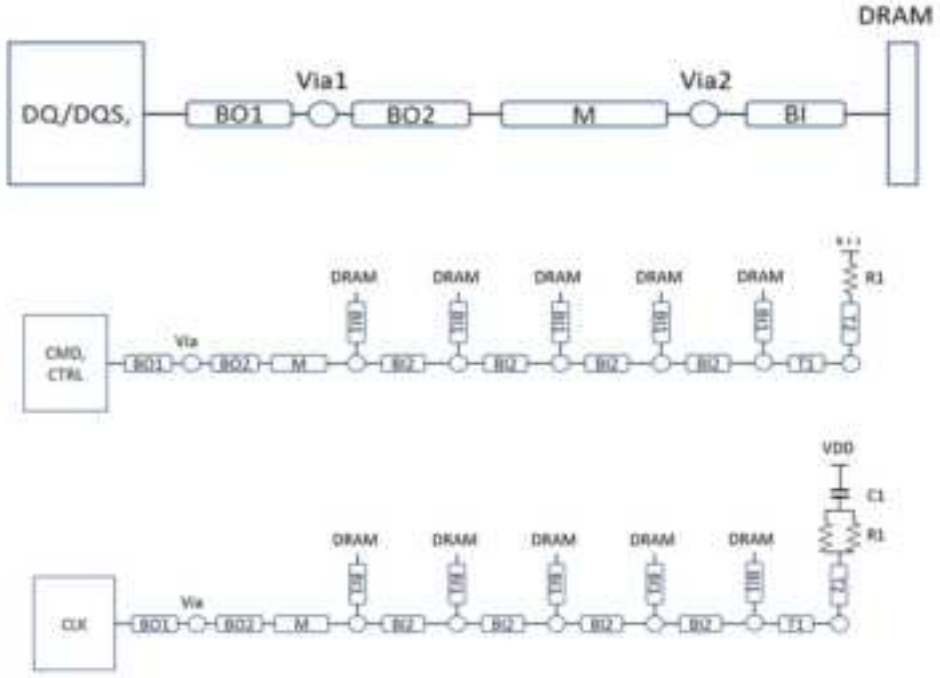


6.4.3.2. 1 Rank x 16 Discrete (Memory Down) Topology

A single channel with 1 rank and x16 memory devices, this 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).

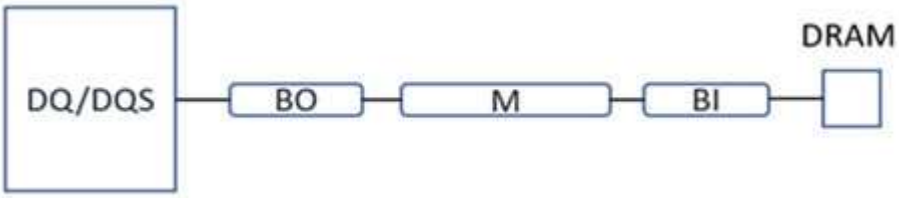
The following figure illustrates the stripline routing for inner pins, for the signal connection topology for 1 rank x 16 memory down configuration.

Figure 25. Stripline Routing for DDR4 1 Rank x 16 Discrete Topology



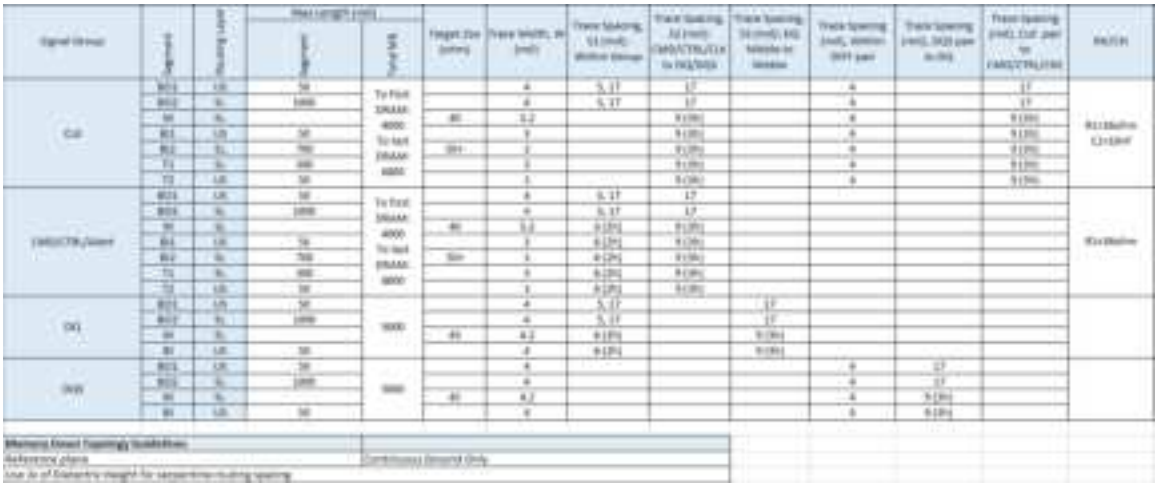
The following figure illustrates the microstrip routing for edge pins, per byte.

Figure 26. Microstrip Routing for DDR4 1 Rank x 16 Discrete Topology



The following table shows the stripline routing guideline for inner pins.

Table 79. Stripline Routing Guideline for GPIO Inner Pins



The following table shows the microstrip routing guideline for edge pins,per byte.

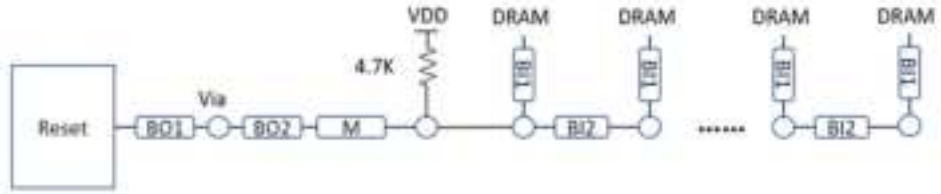
Table 80. Microstrip Routing Guideline for GPIO Edge Pins

6.4.3.3. VREF_CA/RESET Signal Routing Guidelines for 1 Rank x 8 and 1 Rank x 16 Discrete (Memory Down) Topology

The following figure shows the reset routing scheme and setting, which you can apply to both 1 rank x 8 and 1 rank x 16 discrete topologies.

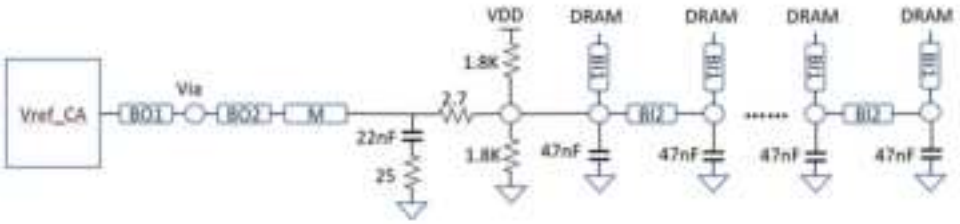
The target impedance for the reset signal is 50 ohms. The reset signal shall have at least 3 x *h* (where *h* represents trace to nearest reference plane height or distance) spacing to other nearby signals on the same layer. The end-to-end reset trace length is not limited but shall not exceed more than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

Figure 27. Reset Scheme and Setting for Memory Down (Discrete) Topology



The following figure shows a VREF_CA routing scheme and setting, which you can apply to both 1 rank x 8 and 1 rank x 16 discrete topologies.

Figure 28. VREF_CA Scheme and Setting for Memory Down (Discrete) Topology



Intel recommends that you use at least a 10 mil trace width for VREF_CA routing on the PCB. The VREF_CA signal should have at least 3 x *h* (where *h* represents trace to nearest reference plane height or distance) spacing to other nearby signals on the same layer. The 1.8K ohm voltage divider circuitry shall be replaced by a 0.9K ohm resistor, pulled up to 0.6 V from the voltage regulator.

6.4.3.4. Skew Matching Guidelines for DDR4 (Memory Down) Discrete Configurations

These skew matching guidelines apply to 1 rank x 8 and 1 rank x 16 memory down topologies.

For skew matching In the DDR4 discrete topology, board designers must follow these rules:

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

The following table shows skew matching guidelines for DDR4 down-memory topology.

Table 81. Skew Matching Guidelines for DDR Memory Down Topology

Length Matching Rules	Length	Time (assuming 170ps/in delay)
Length Matching between DQS and CLK	-500mil < CLK - DQS < 5000mil	-85ps < CLK - DQS < 935ps
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 CMD/ADDR/CTRL and Clock	-120mil < CLK - CMD/ADDR/CTRL < 120mil	-20ps < CLK - CMD/ADDR/CTRL < 20ps
Length matching among CMD/ADDR/CTRL within each channel	< 120mil	< 20ps
Include package length in length matching	Required	Required
Notes		
Keep GND stitching via within 80mil from signal transition which changes reference planes.		
Length matching on ALERT is not required.		

6.4.3.5. Power Delivery Recommendation for DDR4 Discrete Configurations

This section describes PDN design guidelines for the memory side in discrete topology.

The total number of decoupling capacitors is based on single channel; if multiple channels are sharing the same power rail, the number of decoupling capacitors at memories for all channels should be scaled accordingly. You should use smaller decoupling capacitors in memory PDN design to minimize area, inductance, and resistance on the PDN path.

The following table shows the required quantity and value of decoupling capacitors on the PCB, specifically for memory side.

Table 82. PDN Design Guidelines for the Memory Side in Discrete Topology

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

6.4.3.6. DDR4 Simulation Strategy

The simulation strategy is divided into 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 max 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 max 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 performance on the layout, e.g the worst crosstalk (coupling between deep vertical vias), long trace routing and maximum reflection on routing path due to long via stubs if backdrilling is not applied

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

You must perform the channel analysis in the time domain, using a pseudorandom binary sequence (PRBS) pattern for I/O signal generation, while the channel is built by using actual per-pin package models at both ends, including PCB model in the format of scattering parameter along with I/O buffer model at both ends. An I/O buffer IBIS model is used for DDR4 interface signal integrity simulation. Evaluate the eye diagram after the simulation, to ensure the eye specification is met at both ends.

7. Intel Agilex 5 FPGA EMIF IP - LPDDR4 Support

This chapter contains IP parameter descriptions and pin planning information for Intel Agilex 5 FPGA external memory interface IP for LPDDR4.

7.1. Intel Agilex 5 FPGA EMIF IP Parameters for LPDDR4

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 you must set it manually.

7.1.1. Agilex 5 FPGA EMIF IP Parameter for LPDDR4

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 83. 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)
Hydra Remote Access	Specifies whether the Hydra control and status registers are accessible via JTAG, exported to the fabric, or just disabled (Identifier: EX_DESIGN_HYDRA_REMOTE)

Table 84. 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)
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 85. 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 86. Group: General IP Parameters / Advanced Parameters / PHY / Topology

Display Name	Description
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 87. 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 88. 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 89. 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 90. 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 91. 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 92. 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)
Drive Strength	Specifies the termination to be used when driving read data from memory (Identifier: GRP_MEM_ODT_DQ_X_RON)

Table 93. 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 94. 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 (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 95. Group: General IP Parameters / Advanced Parameters / Mem I/O / Memory I/O Settings / Address/Command Bus Reference Voltage (Vref)

Display Name	Description
VrefCA Range	Specifies which of the memory protocol defined ranges will be used (Identifier: GRP_MEM_VREF_CA_X_CA_RANGE)
VrefCA Value	Specifies the initial VrefCA value to be used (Identifier: GRP_MEM_VREF_CA_X_CA_VALUE)

Table 96. 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 97. 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

Display Name	Description
	(Identifier: USER_EXTRA_PARAMETERS)

Table 98. 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)

7.1.2. Intel Agilex 5 FPGA EMIF Memory Device Description IP (LPDDR4) 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 99. Group: Configuration Save

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

Table 100. Group: High-Level Parameters

Display Name	Description
Memory Format	Specifies the packaging format of the memory device (Identifier: MEM_FORMAT)
Number of Channels in Memory Package	Number of Channels in Memory Package. This value must be consistent with the same MEM_NUM_CHANNELS parameter in the top-level EMIF IP GUI. (Identifier: LPDDR4_MEM_DEVICE_NUM_CHANNELS)
Number of Ranks Per Channel	Number of Ranks Per Channel in Memory Package. This value must be consistent with the same MEM_NUM_RANKS parameter in the top-level EMIF IP GUI. (Identifier: LPDDR4_MEM_DEVICE_NUM_RANKS_PER_CHANNEL)
Number of Individual DRAM Components Per Rank	Number of Individual DRAM Components Per Rank in Memory Package. This value must be consistent with the same MEM_COMPS_PER_RANK parameter in the top-level EMIF IP GUI. (Identifier: LPDDR4_MEM_DEVICE_NUM_COMPS_PER_RANK)
Density of Each Memory Die per Channel	Specifies the density of each channel in the memory die in Gb (Identifier: LPDDR4_MEM_DEVICE_DENSITY_GBITS)
Enable Write Data Bus Inversion	Enables Write Data Bus Inversion (Identifier: LPDDR4_MEM_DEVICE_WR_DBI_EN)
Enable Data Mask	Enables Data Masking for write operations (Identifier: LPDDR4_MEM_DEVICE_DM_EN)

Table 101. Group: Memory Interface Parameters / Data Bus

Display Name	Description
DQ Width per DRAM Component	Specifies the DQ width of each LPDDR4 DRAM component. As byte mode is not supported, this value is always 16. To form x32 LP4 interfaces, select 2 components per rank at the EMIF IP level. (Identifier: LPDDR4_MEM_DEVICE_DQ_WIDTH)
Total DQ Width Per Channel	Total DQ Width Per Channel. For LPDDR4 packages, this is the product of the per-DRAM DQ Width and Number of Individual DRAM Components per Rank. (Identifier: LPDDR4_MEM_DEVICE_TOTAL_DQ_WIDTH_PER_CHANNEL)
Burst Length	Burst Length (Identifier: LPDDR4_MEM_DEVICE_BURST_LENGTH)

Table 102. Group: Memory Interface Parameters / Device Topology

Display Name	Description
Device Row Address Width	Specifies the row address width of this LPDDR4 DRAM component. This value is auto-derived from the specified component density. (Identifier: LPDDR4_MEM_DEVICE_ROW_ADDR_WIDTH)
Device Bank Address Width	Specifies the bank address width. This value is fixed as per the JEDEC standard and cannot be changed. (Identifier: LPDDR4_MEM_DEVICE_BANK_ADDR_WIDTH)
Device Column Address Width	Specifies the column address width of this LPDDR4 DRAM component. This value is fixed for all component densities as per the JEDEC standard and cannot be changed. (Identifier: LPDDR4_MEM_DEVICE_COL_ADDR_WIDTH)

Table 103. Group: Memory Timing Parameters / Timing Parameters

Display Name	Description
Memory Clock Frequency	Specifies the Write Clock Frequency Note: This parameter can be auto-computed. (Identifier: PHY_MEMCLK_FREQ_MHZ)
Memory Speedbin	Maximum Data Rate for which this memory device is rated for (Identifier: LPDDR4_MEM_DEVICE_SPEEDBIN)
Memory Write Latency Set	Selects the Write Latency Set for this device. Selection affects auto-calculation of Write Latency. (Identifier: LPDDR4_MEM_DEVICE_WLS)
Memory Read Latency	Read Latency of the memory device in clock cycles Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_CL_CYC)
Memory Write Latency	Write Latency in clock cycles Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_CWL_CYC)

Table 104. Group: Memory Timing Parameters / Pre- and Post-Amble Options

Display Name	Description
Read Preamble Cycles	Duration of read preamble in cycles (Identifier: LPDDR4_MEM_DEVICE_RD_PREAMBLE_CYC)
Read Postamble Cycles	Duration of read postamble in cycles

continued...

Display Name	Description
	(Identifier: LPDDR4_MEM_DEVICE_RD_POSTAMBLE_CYC)
Write Postamble Cycles	Duration of write postamble in cycles (Identifier: LPDDR4_MEM_DEVICE_WR_POSTAMBLE_CYC)

Table 105. Group: Memory Timing Parameters / Advanced Timing Parameters

Display Name	Description
tSR	Minimum duration (Entry to Exit) of Self Refresh in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TSR_NS)
tXSR	Self-Refresh Exit to Next Valid Command Delay Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TXSR_NS)
tXP	Exit Power-Down to Next Valid Command Delay Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TXP_NS)
tCCD	CAS-to-CAS Delay in memory clock cycles Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCCD_CYC)
tRTP	Internal READ to PRECHARGE Command Delay Time in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRTP_NS)
tRCD	RAS-to-CAS Delay in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRCD_NS)
tRPpb	Per-Bank Precharge Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRPPB_NS)
tRPab	All-Bank Precharge Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRPAB_NS)
tRAS	Row Active Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRAS_NS)
tWR	Write Recovery Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TWR_NS)
tWTR	Write-to-Read Delay in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TWTR_NS)
tRRD	RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRRD_NS)
tPPD	Precharge-to-precharge delay in memory clock cycles Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TPPD_CYC)
tFAW	Four-bank ACTIVE window time in ns

continued...

Display Name	Description
	Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TFAW_NS)
tRC	Activate-to-Activate command period (same bank) in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRC_NS)
tREFW	Refresh window time in milliseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TREFW_MS)
Min Number of Refs Reqd	Minimum Number of Refreshes Required Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_MINNUMREFSREQ)
tREFI	Refresh Interval Time in us Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TREFI_NS)
tRFCab	All-Bank Refresh Cycle Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRFCAB_NS)
tRFCpb	Per-Bank Refresh Cycle Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TRFCPB_NS)
tCKE	CKE Minimum Pulse Width time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCKE_NS)
tCMDCKE	Delay from valid command to power-down-entry (CKE low) in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCMDCKE_NS)
tCKELCK	Valid clock requirement after power-down-entry in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCKELCK_NS)
tCSCKE	Valid CS requirement before power-down-entry (CKE low) in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCSCKE_NS)
tCKCKEH	Valid clock requirement before power-down-exit in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCKCKEH_NS)
tCSCKEH	Valid CS requirement before power-down-exit (CKE high) in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCSCKEH_NS)
tMRWCKEL	Delay from MRW command to power-down-entry (CKE low) in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TMRWCKEL_NS)
tZQCKE	Delay from ZQCal Start command to power-down-entry (CKE low) in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TZQCKE_NS)
tMRR	Mode Register Read Command Period Time in memory clock cycles Note: This parameter can be auto-computed.

continued...

Display Name	Description
	(Identifier: LPDDR4_MEM_DEVICE_TMRR_CYC)
tMRW	Mode Register Write Command Period Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TMRW_NS)
tMRD	Mode Register Set Command Period Time in ns Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TMRD_NS)
tESCKE	Delay from SRE command to CKE Input low Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TESCKE_NS)
tZQCAL	ZQ calibration time in microseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TZQCAL_US)
tZQLAT	ZQcal Latch time in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TZQLAT_NS)
tDQSCK_MAX	Maximum DQS output access time from CK in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TDQSCK_MAX_PS)
tDQSCK_MIN	Minimum DQS output access time from CK in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TDQSCK_MIN_PS)
tCKCKEL	Clock valid requirements after power-down-entry (CKE low) in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCKCKEL_NS)
tCKELCMD	Valid command requirement after CKE input low in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCKELCMD_NS)
tCKEHCMD	Valid command requirement after CKE input high in nanoseconds Note: This parameter can be auto-computed. (Identifier: LPDDR4_MEM_DEVICE_TCKEHCMD_NS)

7.2. Intel Agilex 5 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 for your external memory interface are known, you can begin planning your system.

7.2.1. Intel Agilex 5 FPGA EMIF IP Interface Pins

Any I/O banks that do not support transceiver operations in Intel Agilex 5 FPGAs support external memory interfaces.

However, DQS (data strobe), 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.

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.

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. LPDDR4 Component Options

The table and figure below illustrate the pin placement and routing recommendation for a single 32-bit channel, and two 16-bit channels, respectively.

Note: Always consult your memory vendor's data sheet to verify pin placement and routing plans.

Table 106. Pin Options for LPDDR4 x32 and 2 x16

Pins	1 CH x32	2 CH x16	
Data	DQ[15:0]_A DQ[15:0]_B	DQ[15:0]_A	DQ[15:0]_B
Data mask	DMI[1:0]_A DMI[1:0]_B	DMI[1:0]_A	DMI[1:0]_B

continued...

Pins	1 CH x32	2 CH x16	
Data strobe	DQS[1:0]_t_A DQS[1:0]_c_A DQS[1:0]_t_B DQS[1:0]_t_B	DQS[1:0]_t_A DQS[1:0]_c_A	DQS[1:0]_t_B DQS[1:0]_c_B
Address/Command	CA[5:0]_A CS0_A CA[5:0]_B CS0_B	CA[5:0]_A CS0_A	CA[5:0]_B CS0_B
Clock	CK_t_A CK_c_A CK_t_B CK_c_B	CK_t_A CK_c_A	CK_t_B CK_c_B
Reset	RESET_n	RESET_n (Resistor jumper to select from mem_0 or mem_1.)	
Clock Enable	CKE	CKE	

Figure 29. Single-Channel x32 LPDDR4, Single Rank

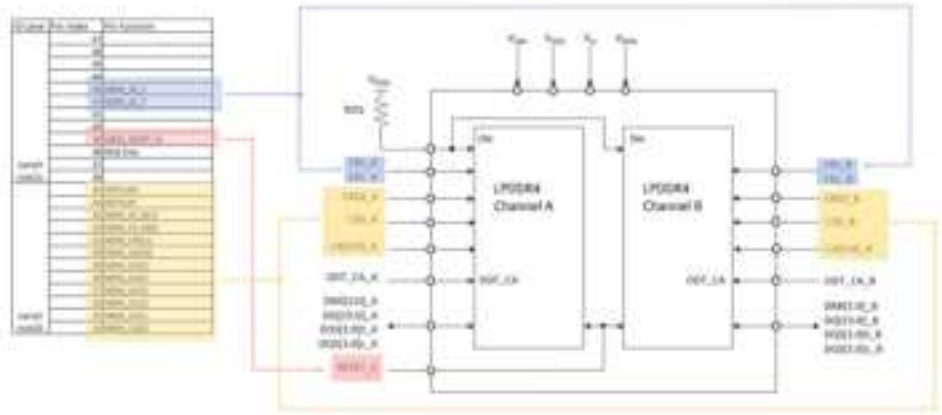
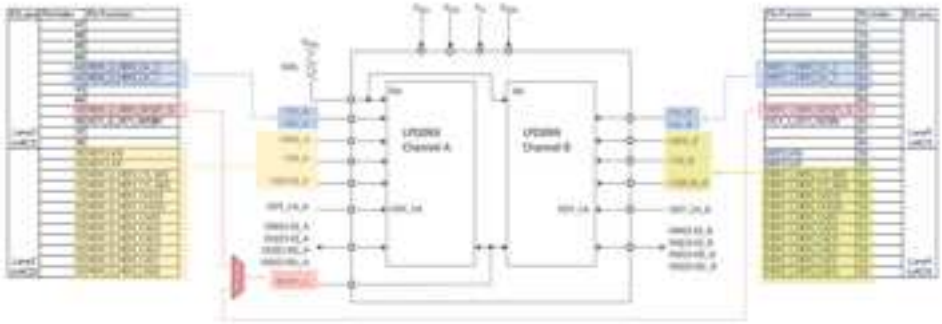


Figure 30. Dual-Channel x16 LPDDR4, Single Rank



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 Intel Agilex 5 devices, consult the EMIF Device Selector on www.intel.com.

Timing closure depends on device resource and routing utilization. For related information, refer to the *Quartus Prime Pro Edition User Guide: Design Optimization*.

7.2.2. Intel Agilex 5 FPGA EMIF IP Resources

The Intel Agilex 5 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 Intel Agilex 5 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 VCCIO_PIO voltage as the memory interface.
- The OCT calibration block uses a single fixed RZQ. 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. Intel Agilex 5 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.

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

7.2.3. Pin Guidelines for Intel Agilex 5 FPGA EMIF IP

The Intel Agilex 5 FPGA contains I/O banks on the top and bottom edges of the device, which can be used by external memory interfaces.

Intel Agilex 5 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.

Intel Agilex 5 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.

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#, 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 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 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).

7.2.3.1. General Guidelines

Observe the following general guidelines when placing pins for your Intel Agilex 5 external memory interface.

1. Ensure that the pins of a single external memory interface reside on the same edge I/O.
2. The LPDDR4 x32 or 2x16 implementation should be confined within the same I/O bank.
3. Two different external memory interfaces cannot share a sub-bank.
4. 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 LPDDR4* topic.
5. Not every byte lane can function as an address and command lane or a data lane. The pin assignment must adhere to the LPDDR4 data width mapping defined in the *Pin Placement for Agilex 5 FPGA EMIF IP for LPDDR4* topic.
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. Pins in any lane in the same IO96 bank that are 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.
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 *LPDDR4 Pin Placement* table in the *Pin Placement* topic in the *Product Architecture* chapter.
9. The address and command for LPDDR4 would utilize 2 IO lanes in the sub-bank. The 2 unused I/O lane in the address and command sub-bank would serve to implement data groups. The data groups must be from the same controller as the address and command signals.
10. An I/O lane must not be used by both address and command pins and data pins.
11. 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.
12. 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

Note: EMIF IP pin-out requirements for the Intel Agilex 5 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.

7.2.3.2. Specific Pin Connection Requirements

PLL

- For LPDDR4, 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 LPDDR4 interfaces is permitted within an I/O bank.

Note: Intel Agilex 5 FPGAs do not support single-ended I/O PLL reference clocks for EMIF IP.

OCT

- For LPDDR4, 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 pins across multiple LPDDR4 interfaces is permitted within an I/O bank.

DQS/DQ/DM

For LPDDR4 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_p pin only.
- You must use pin index 5 for the DQS_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.
- The following table indicates the pin index within an I/O lane and the DQS, DQ and DM pin placement:

pin0	pin1	pin2	pin3	pin4	pin5	pin6	pin7	pin8	pin9	pin10	pin11
DQ	DQ	DQ	DQ	DQSp	DQSn	DM	unused	DQ	DQ	DQ	DQ

7.2.3.3. 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.

7.2.3.4. Clock Signals

LPDDR4 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.

7.2.3.5. Address and Command Pin Placement for LPDDR4

Agilex 5 FPGA IP for LPDDR4 supports fixed address and command pin placements as shown in the following table.

The IP supports up to two ranks.

Address/Command Lane	Index Within Byte Lane	LPDDR4
AC1	11	
	10	
	9	

continued...

Address/Command Lane	Index Within Byte Lane	LPDDR4
	8	
	7	MEM_CK_C
	6	MEM_CK_T
	5	
	4	
	3	MEM_RESET_N
	2	OCT_RZQIN
	1	
	0	
AC0	11	Differential "N-side" reference clock input site
	10	Differential "P-side" reference clock input site
	9	MEM_CS[1]
	8	MEM_CS[0]
	7	MEM_CKE[1]
	6	MEM_CKE[0]
	5	MEM_CA[5]
	4	MEM_CA[4]
	3	MEM_CA[3]
	2	MEM_CA[2]
	1	MEM_CA[1]
	0	MEM_CA[0]

7.2.3.6. Pin Placements for Agilex 5 FPGA EMIF IP for LPDDR4

Intel Agilex 5 FPGA LPDDR4 IP supports fixed address and command pin placement, and fixed data lanes placement.

You can only use fixed byte lanes within the I/O Bank as data lanes. Below is the summary of the location for address and command, and data lanes.

For two-channel x16 LPDDR4, the DQ group placement must follow the controller I/O sub-bank:

Controller	Data Width	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	LPDDR4 x32	DQ[3] P	DQ[2] P	GPIO	GPIO	AC1 P	AC0 P	DQ[1] P	DQ[0] P									
Primary	LPDDR4 1ch x16	GPIO	GPIO	GPIO	GPIO	AC1 P	AC0 P	wDQ[1]	wDQ[0]									
Secondary*	LPDDR4 1ch x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	GPIO	GPIO	GPIO	GPIO									
Primary + Secondary	LPDDR4 2ch x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC0- P	DQ[1] P	DQ[0] P									
Primary + Secondary	LPDDR4 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
- * Not supported on ES0 silicon. ES0 silicon supports LPDDR4x16 only on bottom sub-bank (BL0-BL3).

The diagrams below illustrates the pin connections for address and command and the data group.

Note: Refer to the LPDDR4 pin table in the *Product Architecture* chapter for detailed pin placement for both address and command and DQ pins.

Figure 31. Dual-Channel x16 LPDDR4, Single Rank

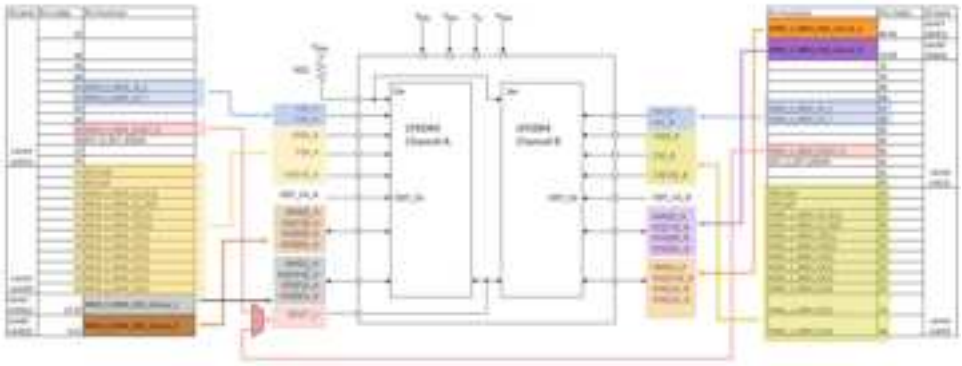
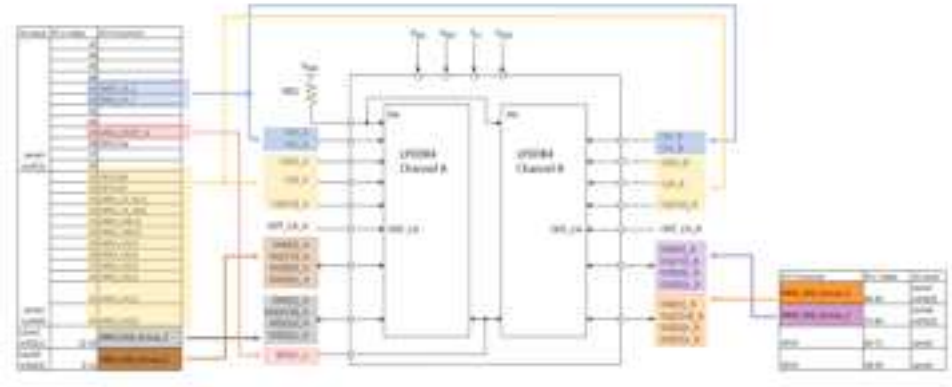


Figure 32. Single-Channel x32 LPDDR4, Single Rank



7.2.3.7. Pin Swapping Guidelines

In Agilex 5 devices, EMIF pin swapping is allowed under certain conditions.

Byte Lane Swapping

You can swap the data lane 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.

Table 108. LPDDR4 Byte Lane Swapping

Controller	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
Primary	LPDDR4 x32	wDQ[3]	wDQ[2]	GPIO	GPIO	wAC1	wAC0	wDQ[1]	wDQ[0]
Primary + Secondary	LPDDR4 x16 2ch	sDQ[1]	sDQ[0]	sAC1	sAC0	wAC1	wAC0	wDQ[1]	wDQ[0]

Example 1: LPDDR4 x32

BL4 and BL5 are not used by EMIF. The BL0, 1, 6 and 7 are used DQ lanes. Byte lane swapping between BL0 and BL1 is allowed; byte lane swapping between BL6 and BL7 is allowed.

Example 2: LPDDR4 x16, 2ch

Channel 0 uses BL0 and BL1 as the DQ lanes, and BL2 and BL3 as AC lanes. Byte lane swapping between BL0,1 is allowed.

Channel 1 uses BL6 and BL7 as the DQ lanes, and BL4 and BL5 as AC lanes. Byte lane swapping between BL6,7 is allowed.

Cross channel DQ lane swapping is not allowed.

Address and Command and CLK Lane

You cannot swap address and command and control signals in a bank. Pin mapping must adhere to the requirements defined in the table in the *Address and Command Pin Placement for LPDDR4* 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 *LPDDR4 Data Width Mapping* topic.

The T and C pins for the CLK_T and CLK_C cannot be swapped with each other, nor can the T and C pins for the DQS_T and DQS_N be swapped with each other.

7.3. LPDDR4 Layout Design Guidelines

This section provides PCB layout design recommendations and guidelines for Agilex 5 E-Series Group B FPGA devices with GPIO-B (Input/Output) silicon implementation.

A successful DDR design on PCB requires not only following the topology and routing guidelines provided here, but also must meet PDN design requirements. For power delivery network (PDN) design guideline information, refer to Agilex 5 Power Distribution Network Design Guidelines, available on the Intel website. For high-speed transceiver PCB layout guidelines, refer to Agilex 5 High Speed PCB Layout Design Guidelines, available on the Intel website.

7.3.1. LPDDR4 PCB Stackup and Design Considerations

The following figures show an example of a PCB stackup with 14 layers that has been used on PCB design for an Intel platform board. You may use other stackups (thin such as PCIE board or thick board) if you meet the recommendations in this guideline.

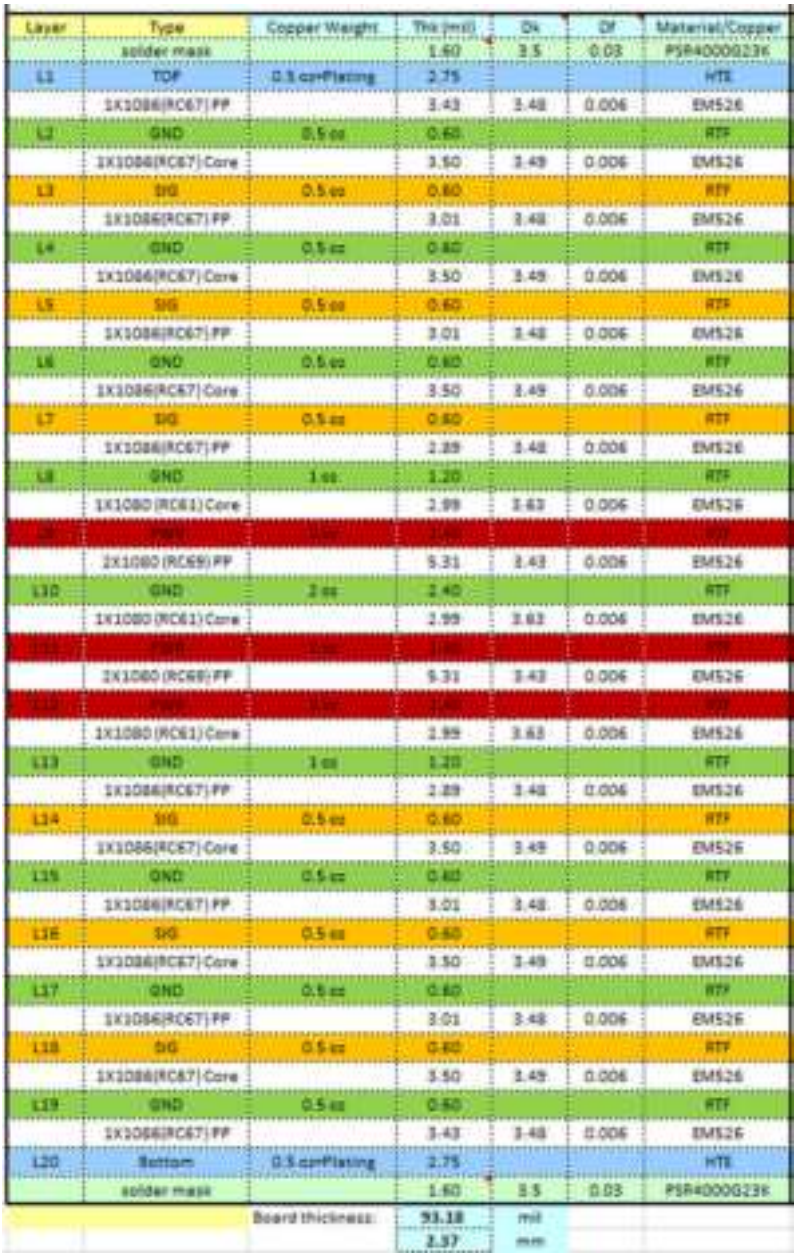
The figure below shows a 14L thin board, high performance Type-IV PCB with micro vias, stacked vias, buried vias and through vias.

Figure 33. 14L Thin Board, High Performance Type-IV PCB Stackup

Layer	Structure	Material	Copper TYPE	Process thikcness (um)	Process thikcness (mil)
	Soldermask	PSR4000 Series		15	0.59
1	Copper	1/3OZ+Plating	HVLP	30	1.18
	Prepreg	1x1035 RC75%		64	2.52
2	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
3	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		74	2.91
4	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
5	Copper	1/3OZ+Plating	HVLP	25	0.98
	Prepreg	1x1078 RC66%		75	2.95
6	Copper	H OZ	HVLP	15	0.59
	Core	0.130mm H/2		130	5.12
7	Copper	2 OZ	RTF	62	2.44
	Prepreg	3x1078 RC66%		200	7.87
8	Copper	2 OZ	RTF	62	2.44
	Core	0.130mm H/2		130	5.12
9	Copper	H OZ	HVLP	15	0.59
	Prepreg	1x1078 RC66%		75	2.95
10	Copper	1/3OZ+Plating	HVLP	25	0.98
	Prepreg	1x1078 RC66%		63	2.48
11	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		74	2.91
12	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
13	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1035 RC75%		65	2.56
14	Copper	1/3OZ+Plating	HVLP	30	1.18
	Soldermask	PSR4000 Series		15	0.59

The figure below shows a 20L thick Type-III Board stack-up (high performance with PTH with/without backdrill example used at Intel platform boards and development kits.

Figure 34. 20L Thick Type III Board Stackup

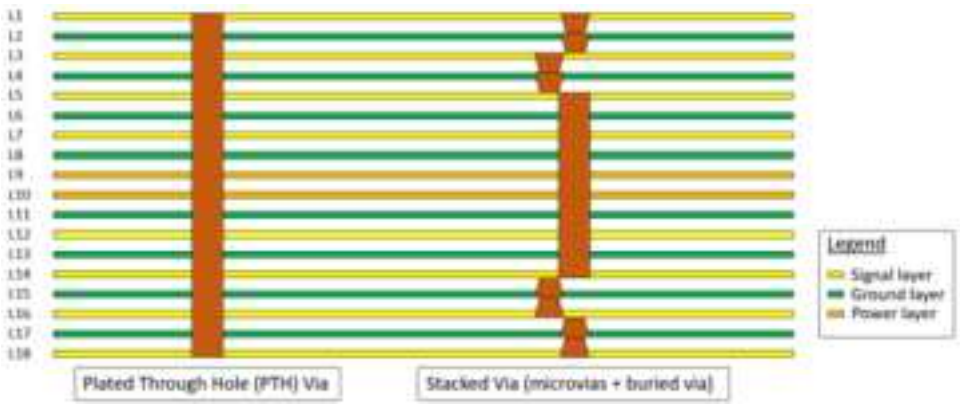


Layer	Type	Copper Weight	Thk (mil)	Dk	Df	Material/Copper
	solder mask		1.60	3.5	0.03	PSR4000G23K
L1	TOP	0.5 oz+Plating	2.75			HTE
	1X1086(RC67) PP		3.43	3.48	0.006	EM526
L2	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L3	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L4	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L5	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L6	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L7	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		2.89	3.48	0.006	EM526
L8	GND	1 oz	1.20			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
[illegible]	[illegible]	[illegible]	[illegible]			[illegible]
	2X1080 (RC69) PP		5.31	3.43	0.006	EM526
L10	GND	2 oz	2.40			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
[illegible]	[illegible]	[illegible]	[illegible]			[illegible]
	2X1080 (RC69) PP		5.31	3.43	0.006	EM526
[illegible]	[illegible]	[illegible]	[illegible]			[illegible]
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L13	GND	1 oz	1.20			RTF
	1X1086(RC67) PP		2.89	3.48	0.006	EM526
L14	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L15	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L16	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L17	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L18	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L19	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.43	3.48	0.006	EM526
L20	Bottom	0.5 oz+Plating	2.75			HTE
	solder mask		1.60	3.5	0.03	PSR4000G23K
		Board thickness:	93.18	mil		
			2.37	mm		

A type-IV PCB is a precise and high-quality PCB. This type-IV PCB utilizes not only PTH vias to connect from top to bottom layers, but also stacked vias, micro vias and buried vias to connect between layers. For example, a full-height stacked via of a 14-layer PCB is made up of a combination of dual-stacked micro vias and buried vias.

The following figure depicts a cross-sectional comparison of a PTH and a stacked via.

Figure 35. Cross-sectional Comparison Between PTH and Stacked Via



7.3.2. LPDDR4 General Design Considerations

General DDR Signal Routing Guideline on PCB

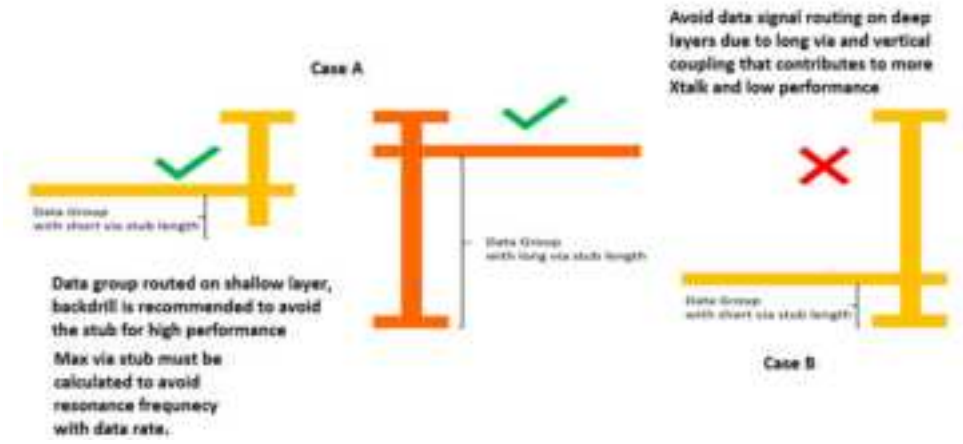
Intel recommends to route all data signals within a specific group on the same layer.

The figure below illustrates a routing example for a type-III PCB board for an LPDDR4 design. Data Group signals such as DQ, DM and DQS signals should be routed on shallow layers as stripline with the least Z-height via transition to avoid vertical crosstalk to achieve high performance.

For example, the recommended routing layers for data group on a 20-layers board and using PTH via will be on the top half of the PCB such as L3, L5 and L7. Other signals such as CA, CTRL and clock signals can be routed with longer Z-height via transition on the bottom half of the PCB such as L14, L16 and L18.

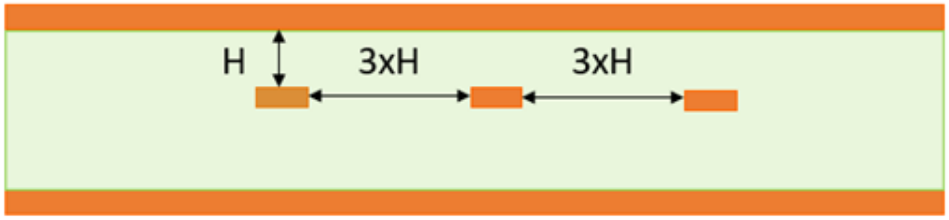
Minimal stub effect or back drill is recommended, but not required, to avoid high reflection for maximum data rate performance. Long via stubs will affect the ISI of channel but the impact of ISI is less than impact of crosstalk for the max data rate performance.

Figure 36. Case A Routing is Suggested for Data Group Signals Over Case B



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

Figure 37. Minimum Trace-to-Trace Separation Distance



DDR FPGA Break Out Routing

Agilex 5 devices come with various pitch sizes for different FPGA pins. The GPIO pin pitch is very small, the device BGA pad stack is also very small, therefore, it is highly recommended to use a dog bone configuration for inner GPIO pins fanout along with stripline routing; however, for the GPIO pins on the edge of device, it is recommended to use microstrip routing. Both microstrip and stripline routing guidelines are presented in this document for all supported EMIF interfaces and topologies.

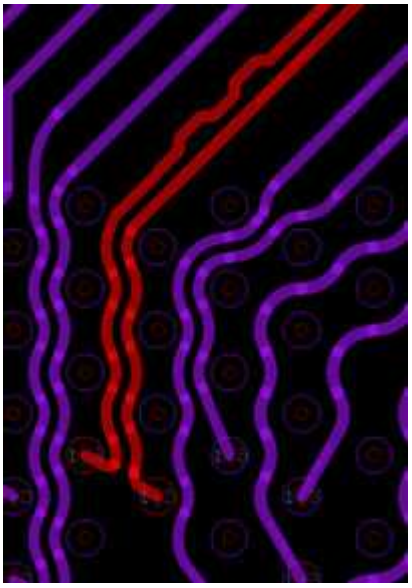
DDR Differential Signals Routing

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 area.

Intel recommends a symmetrical fan-out routing at the FPGA pin field. Non-symmetrical routing for differential signals will cause 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.

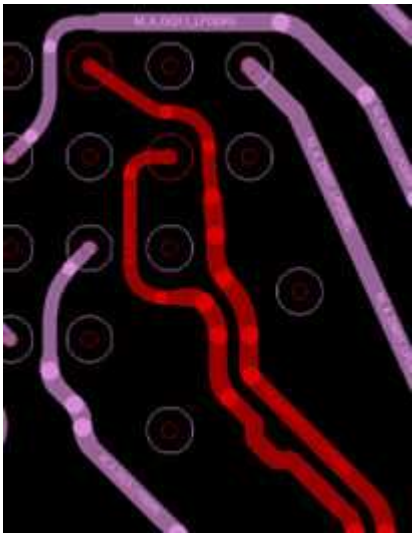
The figure below shows the symmetrical routing of differential signals (DQS/CLK) at FPGA pin field along with length/skew matching between P/N lanes right after FPGA device edge.

Figure 38. Symmetrical Routing of Differential Signals (DQS/CLK) at FPGA Pin Field



The figure below shows the single-ended routing for differential signals (DQS/CLK) at DRAM pin field if the pitch is very small along with skew matching right at edge of DRAM pin field.

Figure 39. Single-ended Routing for Differential Signals (DQS/CLK) at DRAM



Intel recommends to implement length/skew matching for differential signals (if there is) right after FPGA device to avoid additional shifting on differential signals common mode voltage.

In case of having limitation for implementing symmetrical routing at DRAM pin field for differential signals due to very small pitch, Intel recommends to route the differential signals as single-ended signals within the DRAM pin field, ensuring to keep 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/N lanes must be applied before reaching the DRAM pin field.

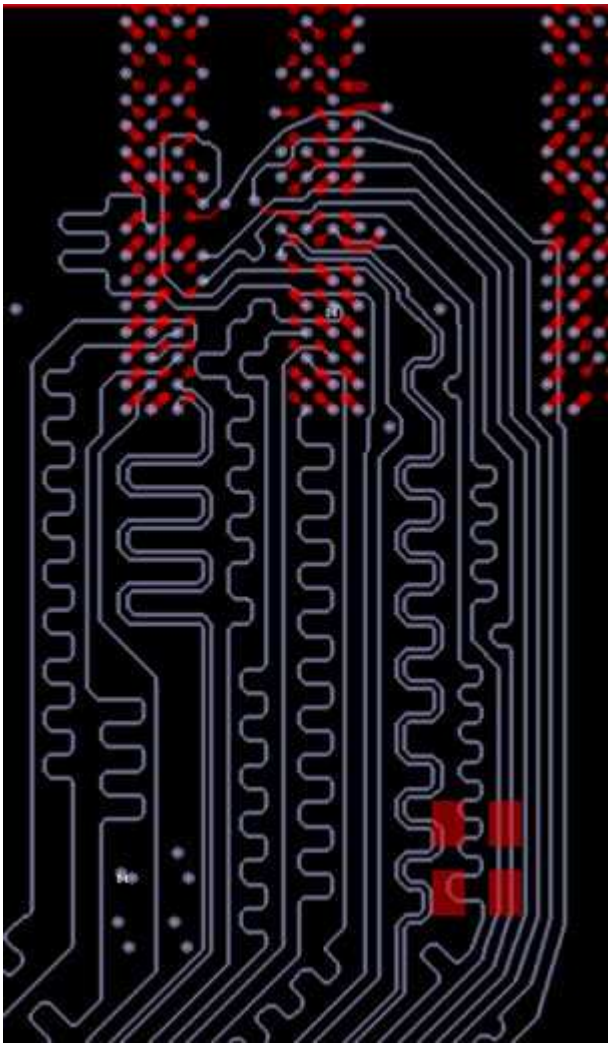
Ground Plane and Return Path

A continuous and solid ground reference plane is crucial for data lines to ensure good signal integrity performance. Low impedance ground return path from the FPGA to DRAM devices should be provisioned. In addition, it is desirable to keep ground stitching vias within 80 mils from signal transition for better return path on signal via and better signal integrity performance.

DRAM Break Out in Layout Guideline

For discrete DRAM components on PCB, you can either use the dog-bone or via in pad at DRAM for the signal transition from inner layer to DRAM. If dog-bone via transition is used, it is recommended to separate them with larger pitch to avoid crosstalk between signal vias.

Figure 40. Data Signal Group Routing on PCB for Memory Down Configuration



7.3.3. LPDDR4 Interface Design Guidelines

This section describes PCB layout guidelines for LPDDR4 interfaces. LPDDR4 is supported by Agilex 5 E-Series group B devices only for memory down configurations. It supports both thin and thick PCB stackups. The maximum supported data rates vary depending on the selected topology.

7.3.3.1. LPDDR4 Discrete Component/Memory Down Topology (up to 32 bits interface)

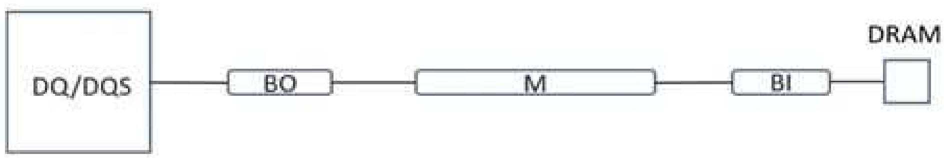
LPDDR4 memory down supports single-rank configuration up to 32 bits. There are four DRAM interface signal groupings, namely: data group, command-address group, control group, and clock group. The connection between the FPGA and DRAM device uses point-to-point topology as depicted in the following two figures.

The following figure illustrates the stripline routing for inner pins.

Figure 41. Stripline Routing for Data, CA, CTRL and Clock Signals Point-to-Point Topology

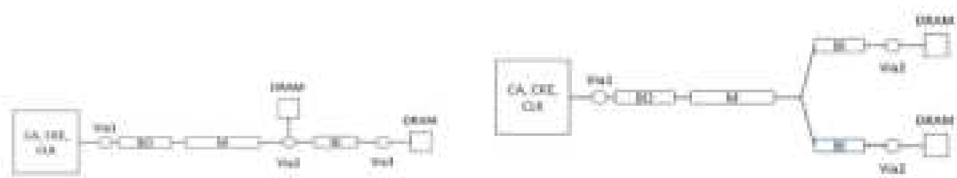
The following figure illustrates the microstrip routing for the edge pins per byte.

Figure 42. Microstrip Routing for Data Signals Point-to-Point Topology



The following figure shows the connection topology for CA, CLK, CTRL signals for LPDDR4 topology in daisy-chain and T-Line connections.

Figure 43. Stripline Routing for CA, CLK, CTRL Daisy-chain and T-Line Topology



The following tables provide a comprehensive routing guideline for each of the LPDDR4 signals based on memory down topology. For example, the trace impedance, the total trace length, the maximum length of the main trace routing can be derived by subtracting break-out and break-in trace segment length routed from total trace length.

Table 109. Stripline Routing Guide for LPDDR4 Memory Down Topology

Signal Group	Segment	Routing Layer	Max Length (mil)		Target Zse (ohms)	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								
CA (Direct Connect)	BO1/BO2	SL	450	4000		3	5, 17	17		4		17
	M	SL	2500		40	3.2		9 (3H)		4		9 (3H)
	BI	SL	50			3		9 (3H)		4		9 (3H)
CA/CS/CKE (Daisy or T)	BO	SL	300	4000		3	5, 17	17				
	M	SL	2500		40	3.2	8 (2H)	9 (3H)				
	BI	SL	1000			3	8 (2H)	9 (3H)				
DQ	BO1	US	50	4000		4	5, 17		17			
	BO2	SL	1000			4	5, 17		17			
	M	SL			40	3.2	8 (2H)		9 (3H)			
	BI	US	50			4	8 (2H)		9 (2H)			
DQS	BO1	US	50	4000		4				4	17	
	BO2	SL	1000			4				4	17	
	M	SL			40	3.2				4	9 (3H)	
	BI	US	50			4				4	9 (3H)	

The above table shows the stripline routing guideline for LPDDR4 memory down topology. The *h* value in the table represents the minimum substrate height from signal layer to reference layer. The signal trace width, and minimum spacing/gaps (in mils) between edge to edge of signal traces are based on the default stackup; however, PCB designers can use the target impedance for any other stackups.

The table below shows microstrip routing for LPDDR4 memory down topology.

Table 110. Microstrip Routing Guide for LPDDR4 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
DQ	BO	L6	350	3000	40	5.5	9 (3h)		9 (3h)			
	M	L6										
	BI	L6	50			3	4					
DQS	BO	L6	300	3000	40	5.5				4	4	
	M	L6								4	12 (4h)	
	BI	L6	50			3				4	12 (4h)	
Memory Down Topology Guidelines												
Reference plane				Continuous Ground Only								
Use 3x of Dielectric height for serpentine routing spacing												
For DQ/DQS signals routing for 8 or 4 layer board, please utilize internal routing layer first, if needed top routing layer can be used without board via transition. Avoid using bottom routing layer with via transition.												

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

Skew matching for LPDDR4 interfaces 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 requirement for DQ and DQS within a byte group, DQS and CLK. The table below provides a detailed skew matching guideline to facilitate PCB trace routing efforts. In this table, the length matching criteria represents a default PCB on an Intel platform board design; you must always follow the skew matching criteria in any other stackup.

Table 111. Skew Matching Requirement for LPDDR4 Memory Down Topology

Length Matching Rules	Length	Time (assuming 170ps/in delay)
Length matching between CLK and DQS per x16	-2000mil < CLK - DQS < 1000mil	-340ps < CLK - DQS < 170ps
Length matching between DQ and DQS per x8	-400mil < DQ - DQS < 250mil	-68ps < DQ - DQS < 43ps
Length matching between DQ signals per x8	< 250mil	< 43ps
Length matching between CLK and CA bits per x16	-200mil < CLK - CA < 200mil	-34ps < CLK - CA < 34ps
Length matching between CA bits per x16	< 200mil	< 34ps
Length matching between CLK and CS/CKE per x16	-500mil < CLK - CS/CKE < 500mil	-85ps < CKL - CS/CKE < 85ps
Length matching between CS/CKE bits per x16	< 200mil	< 34ps
Length matching between DQS_N and DQS_P	< 5mil	< 1ps
Length matching between CLK_N and CLK_P	< 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		

LPDDR4 eye margin is sensitive to crosstalk, especially when the signals are routed on deep layers in stackup. The deep-layer vertical transition induces more vertical coupling between signals and hence more crosstalk.

The maximum data rate of LPDDR4 depends on the DDR memory down configuration and the type of PCB stackup. Reduced data transfer rate is seen whenever 2 x rank LPDDR4 memory down configuration is used.

7.3.3.2. LPDDR4 Simulation Strategy

The simulation strategy is divided into 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 max 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 max 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 performance on the layout, e.g the worst crosstalk (coupling between deep vertical vias), long trace routing and maximum reflection on routing path due to long via stubs if backdrilling is not applied

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

You must perform the channel analysis in the time domain, using a pseudorandom binary sequence (PRBS) pattern for I/O signal generation, while the channel is built by using actual per-pin package models at both ends, including PCB model in the format of scattering parameter along with I/O buffer model at both ends. An I/O buffer IBIS model is used for DDR4 interface signal integrity simulation. Evaluate the eye diagram after the simulation, to ensure the eye specification is met at both ends.

8. Intel Agilex 5 FPGA EMIF IP - LPDDR5 Support

This chapter contains IP parameter descriptions and pin planning information for Intel Agilex 5 FPGA external memory interface IP for LPDDR5.

8.1. Intel Agilex 5 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 5 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 112. 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)
Hydra Remote Access	Specifies whether the Hydra control and status registers are accessible via JTAG, exported to the fabric, or just disabled (Identifier: EX_DESIGN_HYDRA_REMOTE)

Table 113. 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)
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)

continued...

Display Name	Description
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 114. 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) (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 115. Group: General IP Parameters / Advanced Parameters / PHY / Topology

Display Name	Description
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 116. 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 117. 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 118. 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 119. 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.

continued...

Display Name	Description
	(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 120. 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 121. 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 122. 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

continued...

Display Name	Description
	(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 123. 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 124. 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 (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 125. 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 126. 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 127. 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 128. 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 129. 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)

8.1.2. Intel Agilex 5 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 130. Group: Configuration Save

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

Table 131. Group: High-Level Parameters

Display Name	Description
Memory Format	Specifies the packaging format of the memory device (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 in Memory Package	Number of Channels in Memory Package. This value must be consistent with the same MEM_NUM_CHANNELS parameter in the top-level EMIF IP GUI. (Identifier: LPDDR5_MEM_DEVICE_NUM_CHANNELS)

continued...

Display Name	Description
Number of Ranks Per Channel	Number of Ranks Per Channel in Memory Package. This value must be consistent with the same MEM_NUM_RANKS parameter in the top-level EMIF IP GUI. (Identifier: LPDDR5_MEM_DEVICE_NUM_RANKS_PER_CHANNEL)
Number of Individual DRAM Components Per Rank	Number of Individual DRAM Components Per Rank in Memory Package. This value must be consistent with the same MEM_COMPS_PER_RANK parameter in the top-level EMIF IP GUI. (Identifier: LPDDR5_MEM_DEVICE_NUM_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 132. 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 133. 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. (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 134. 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 135. 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 136. 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)
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)

continued...

Display Name	Description
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. (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.

continued...

Display Name	Description
	(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)
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)

continued...

Display Name	Description
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)

8.2. Agilex 5 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 5 FPGA EMIF IP Interface Pins

Any I/O banks that do not support transceiver operations in Agilex 5 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. LPDDR5 Component 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 137. 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]_A	DM[1:0]_B

continued...

Pins	1CH x32	2CH x16	
	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_c_B	CK_t_A CK_c_A	CK_t_B CK_c_B
Reset	RESET_n	RESET_n (Resistor jumper to select from mem_0 or mem_1.)	

Figure 44. Pin Options for LPDDR5 2ch x16

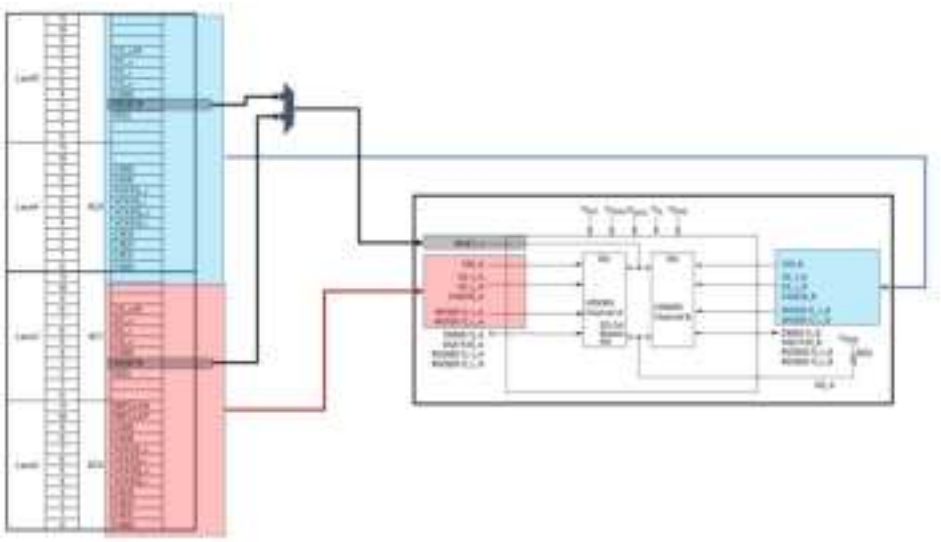
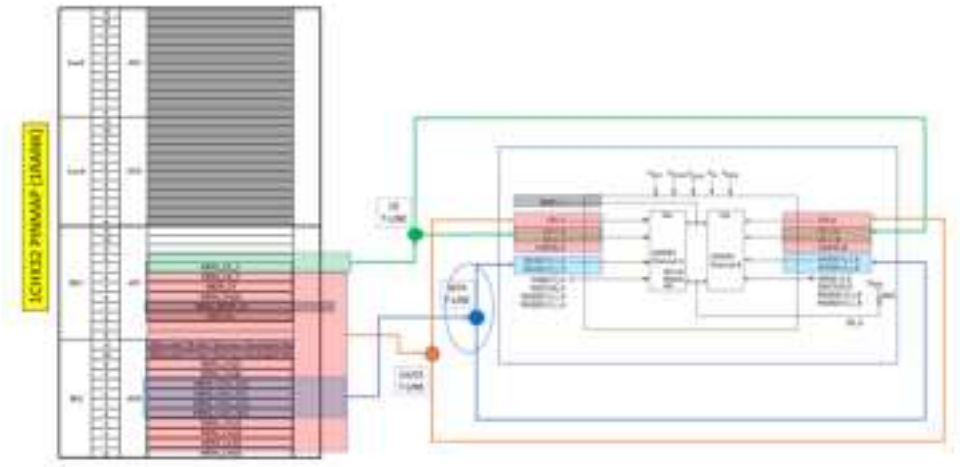


Figure 45. Pin Options for LPDDR5 1ch x32



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 5 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 5 FPGA EMIF IP Resources

The Agilex 5 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 5 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 5 FPGA EMIF IP

The Agilex 5 FPGA contains I/O banks on the top and bottom edges of the device, which can be used by external memory interfaces.

Agilex 5 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 5 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 5 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. Pins in any lane in the same IO96 bank that are 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 5 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 Intel Agilex 5 FPGA LPDDR5 EMIF IP

8.2.4.1. Address and Command Pin Placement for LPDDR5

Agilex 5 FPGA LPDDR5 IP supports fixed address and command pin placement as shown in the following table. The IP supports up to 2 ranks.

Table 138. 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]

continued...

Address/Command Lane	Index Within Byte Lane	LPDDR5
	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 5 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 139. 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									
Secondary	LPDDR5 x16[1]	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:
- [1] Not supported on ES0 silicon. ES0 silicon supports LPDDR5x16 only on bottom sub-bank (BL0-BL3).
- 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 140. 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 Layout Design Guidelines

This section provides PCB layout design recommendations and guidelines for Agilex 5 E-Series Group B FPGA devices with GPIO-B (Input/Output) silicon implementation.

A successful DDR design on PCB requires not only following the topology and routing guidelines provided here, but also must meet PDN design requirements. For power delivery network (PDN) design guideline information, refer to Agilex 5 Power

Distribution Network Design Guidelines, available on the Intel website. For high-speed transceiver PCB layout guidelines, refer to Agilex 5 High Speed PCB Layout Design Guidelines, availabel on the Intel website.

8.3.1. LPDDR5 PCB Stackup and Design Considerations

The following figures show an example of a PCB stackup with 14 layers that has been used on PCB design for an Intel platform board. You may use other stackups (thin such as PCIE board or thick board) if you meet the recommendations in this guideline.

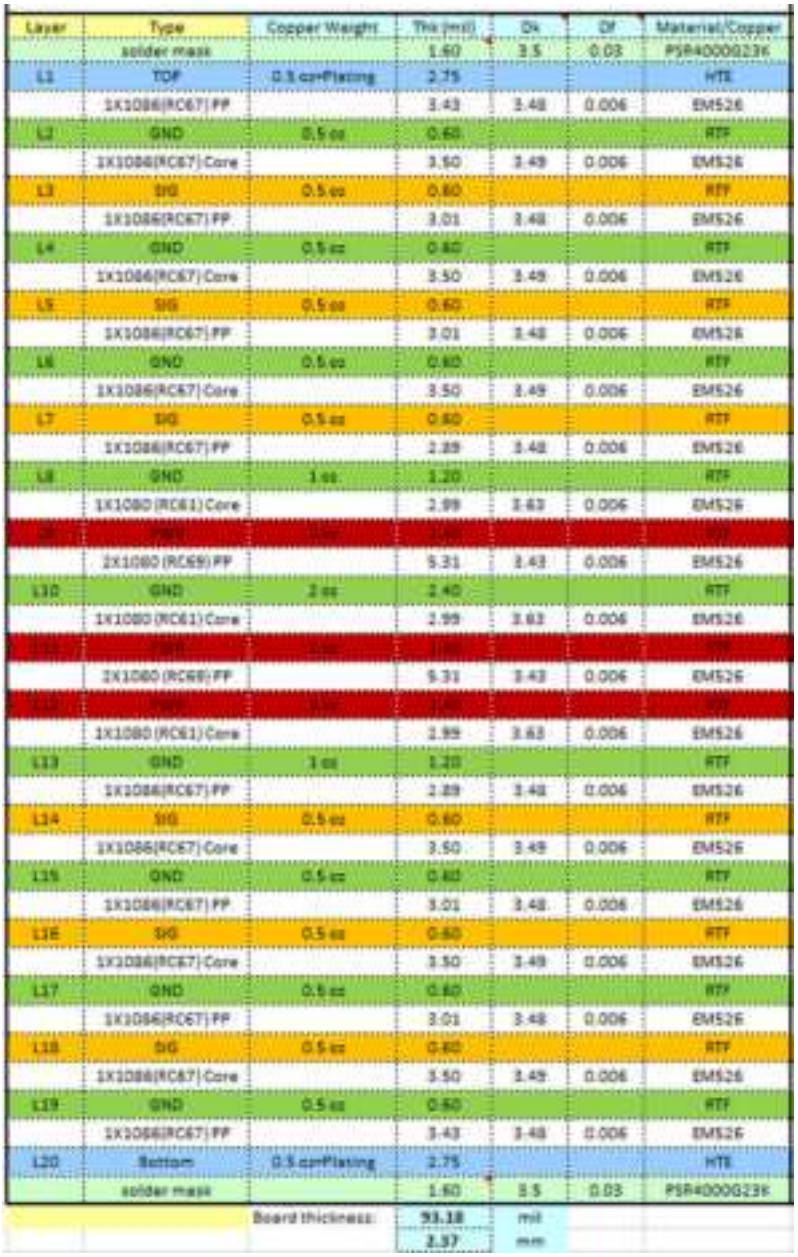
The figure below shows a 14L thin board, high performance Type-IV PCB with micro vias, stacked vias, buried vias and through vias.

Figure 46. 14L Thin Board, High Performance Type-IV PCB Stackup

Layer	Structure	Material	Copper TYPE	Process thickness (um)	Process thickness (mil)
	Soldermask	PSR4000 Series		15	0.59
1	Copper	1/3OZ+Plating	HVLP	30	1.18
	Prepreg	1x1035 RC75%		64	2.52
2	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
3	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		74	2.91
4	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
5	Copper	1/3OZ+Plating	HVLP	25	0.98
	Prepreg	1x1078 RC66%		75	2.95
6	Copper	H OZ	HVLP	15	0.59
	Core	0.130mm H/2		130	5.12
7	Copper	2 OZ	RTF	62	2.44
	Prepreg	3x1078 RC66%		200	7.87
8	Copper	2 OZ	RTF	62	2.44
	Core	0.130mm H/2		130	5.12
9	Copper	H OZ	HVLP	15	0.59
	Prepreg	1x1078 RC66%		75	2.95
10	Copper	1/3OZ+Plating	HVLP	25	0.98
	Prepreg	1x1078 RC66%		63	2.48
11	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		74	2.91
12	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1078 RC66%		62	2.44
13	Copper	1/3OZ+Plating	HVLP	20	0.79
	Prepreg	1x1035 RC75%		65	2.56
14	Copper	1/3OZ+Plating	HVLP	30	1.18
	Soldermask	PSR4000 Series		15	0.59

The figure below shows a 20L thick Type-III Board stack-up (high performance with PTH with/without backdrill example used at Intel platform boards and development kits.

Figure 47. 20L Thick Type III Board Stackup

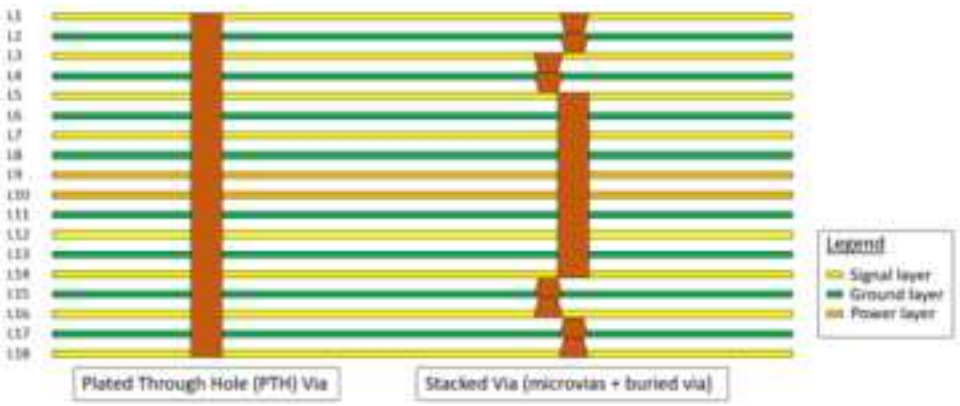


Layer	Type	Copper Weight	Thk (mil)	Dk	Df	Material/Copper
	solder mask		1.60	3.5	0.03	PSR4000G23K
L1	TOP	0.5 oz+Plating	2.75			HTE
	1X1086(RC67) PP		3.43	3.48	0.006	EM526
L2	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L3	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L4	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L5	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L6	GND	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L7	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) PP		2.89	3.48	0.006	EM526
L8	GND	1 oz	1.20			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L9	PWR	1 oz	1.20			RTF
	2X1080 (RC69) PP		5.31	3.43	0.006	EM526
L10	GND	2 oz	2.40			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L11	PWR	1 oz	1.20			RTF
	2X1080 (RC69) PP		5.31	3.43	0.006	EM526
L12	PWR	1 oz	1.20			RTF
	1X1080 (RC61) Core		2.99	3.63	0.006	EM526
L13	GND	1 oz	1.20			RTF
	1X1086(RC67) PP		2.89	3.48	0.006	EM526
L14	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L15	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L16	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L17	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.01	3.48	0.006	EM526
L18	SIG	0.5 oz	0.60			RTF
	1X1086(RC67) Core		3.50	3.49	0.006	EM526
L19	GND	0.5 oz	0.60			RTF
	1X1086(RC67) PP		3.43	3.48	0.006	EM526
L20	Bottom	0.5 oz+Plating	2.75			HTE
	solder mask		1.60	3.5	0.03	PSR4000G23K
		Board thickness:	93.18	mil		
			2.37	mm		

A type-IV PCB is a precise and high-quality PCB. This type-IV PCB utilizes not only PTH vias to connect from top to bottom layers, but also stacked vias, micro vias and buried vias to connect between layers. For example, a full-height stacked via of a 14-layer PCB is made up of a combination of dual-stacked micro vias and buried vias. depicts a cross-sectional comparison of a PTH and a stacked via.

A type-III PCB board with PTH vias, which is used to implement DDR4 designs and can also be used for LPDDR5 designs.

Figure 48. Cross-sectional Comparison Between PTH and Stacked Via



8.3.2. LPDDR5 General Design Considerations

General DDR Signal Routing Guideline on PCB

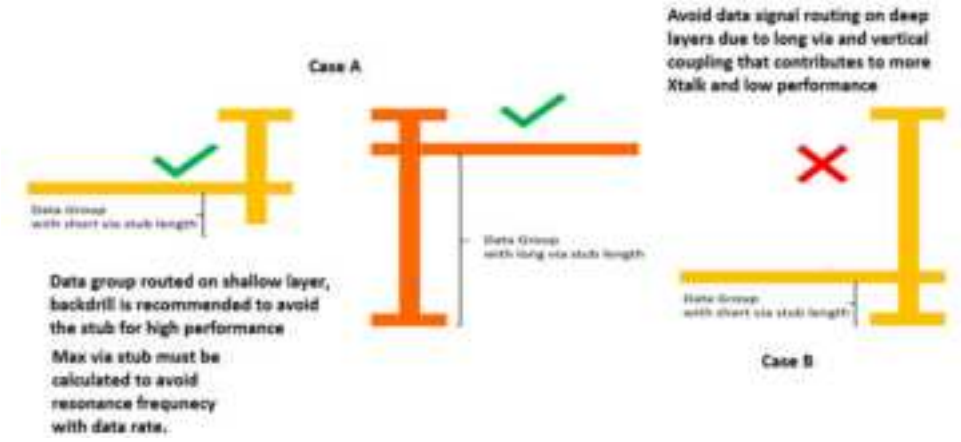
Intel recommends to route all data signals within a specific group on the same layer.

The figure below illustrates a routing example for a type-III PCB board for a DDR4, LPDDR4, or LPDDR5 design. Data Group signals such as DQ, DM and DQS signals should be routed on shallow layers as stripline with the least Z-height via transition to avoid vertical crosstalk to achieve high performance.

For example, the recommended routing layers for data group on a 20-layers board and using PTH via will be on the top half of the PCB such as L3, L5 and L7. Other signals such as CA, CTRL and clock signals can be routed with longer Z-height via transition on the bottom half of the PCB such as L14, L16 and L18.

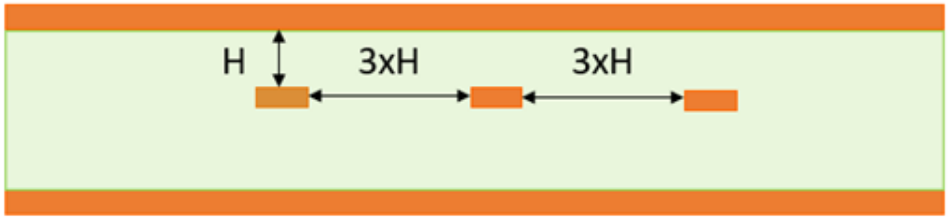
Minimal stub effect or back drill is recommended, but not required, to avoid high reflection for maximum data rate performance. Long via stubs will affect the ISI of channel but the impact of ISI is less than impact of crosstalk for the max data rate performance.

Figure 49. Case A Routing is Suggested for Data Group Signals Over Case B



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

Figure 50. Minimum Trace-to-Trace Separation Distance



DDR FPGA Break Out Routing

Agilex 5 devices come with various pitch sizes for different FPGA pins. The GPIO pin pitch is very small, the device BGA pad stack is also very small, therefore, it is highly recommended to use a dog bone configuration for inner GPIO pins fanout along with stripline routing; however, for the GPIO pins on the edge of device, it is recommended to use microstrip routing. Both microstrip and stripline routing guidelines are presented in this document for all supported EMIF interfaces and topologies.

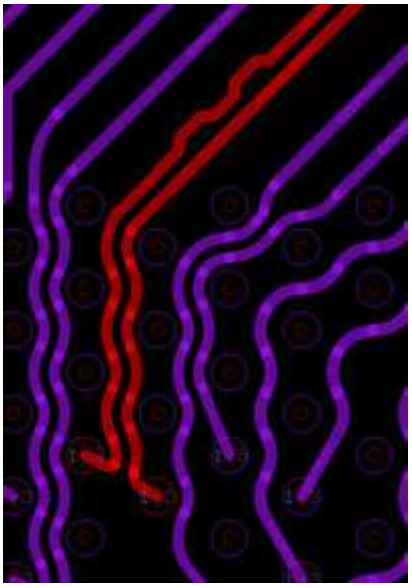
DDR Differential Signals Routing

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 area.

Intel recommends a symmetrical fan-out routing at the FPGA pin field. Non-symmetrical routing for differential signals will cause 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.

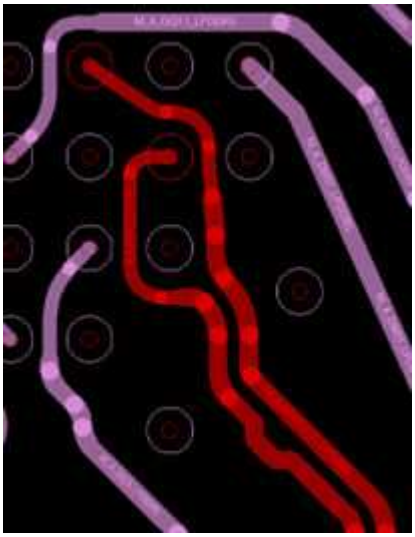
The figure below shows the symmetrical routing of differential signals (DQS/CLK) at FPGA pin field along with length/skew matching between P/N lanes right after FPGA device edge.

Figure 51. Symmetrical Routing of Differential Signals (DQS/CLK) at FPGA Pin Field



The figure below shows the single-ended routing for differential signals (DQS/CLK) at DRAM pin field if the pitch is very small along with skew matching right at edge of DRAM pin field.

Figure 52. Single-ended Routing for Differential Signals (DQS/CLK) at DRAM



Intel recommends to implement length/skew matching for differential signals (if there is) right after FPGA device to avoid additional shifting on differential signals common mode voltage.

In case of having limitation for implementing symmetrical routing at DRAM pin field for differential signals due to very small pitch, Intel recommends to route the differential signals as single-ended signals within the DRAM pin field, ensuring to keep 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/N lanes must be applied before reaching the DRAM pin field.

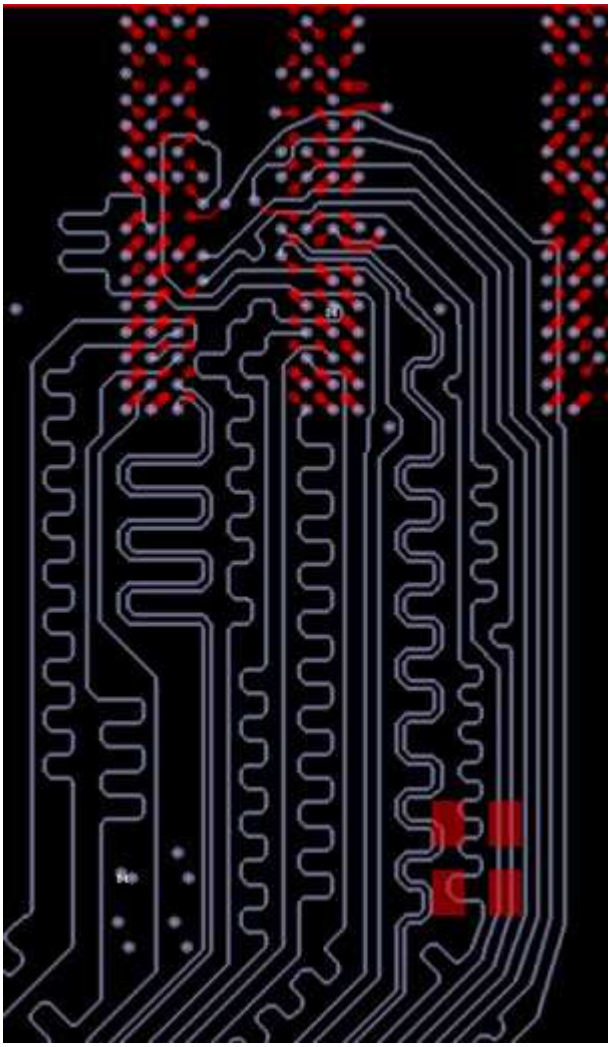
Ground Plane and Return Path

A continuous and solid ground reference plane is crucial for data lines to ensure good signal integrity performance. Low impedance ground return path from the FPGA to DRAM devices should be provisioned. In addition, it is desirable to keep ground stitching vias within 80 mils from signal transition for better return path on signal via and better signal integrity performance.

DRAM Break Out in Layout Guideline

For discrete DRAM components on PCB, you can either use the dog-bone or via in pad at DRAM for the signal transition from inner layer to DRAM. If dog-bone via transition is used, it is recommended to separate them with larger pitch to avoid crosstalk between signal vias.

Figure 53. Data Signal Group Routing on PCB for Memory Down Configuration



8.3.3. LPDDR5 Interface Design Guidelines

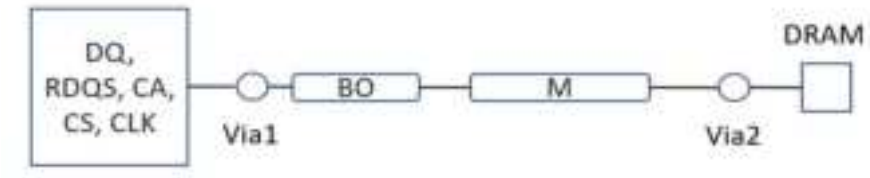
This section describes PCB layout guidelines for LPDDR5 interfaces. Agilex 5 E-Series devices Group B support LPDDR5 interfaces for memory down configuration only. Both thin and thick PCB stackups are supported. The maximum supported data rates depend on the selected topology.

8.3.3.1. LPDDR5 Discrete Component/Memory Down Topology (Single Rank or Dual-Rank)

LPDDR5 memory down support is available in two configurations: single rank or dual–rank.

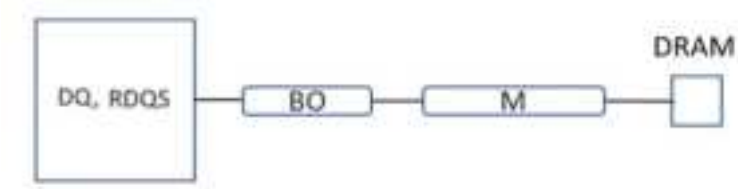
There are four DRAM interface signal groupings: data group, command-address group, control group, and clock group. The FPGA to DRAM connection uses point-to-point topology for data group, command-address group, control group, and clock group. The figure below shows the stripline routing topology for inner pins.

Figure 54. Stripline Routing Topology for Data, CA, CTRL and Clock Signals Point-to-Point Topology



The figure below shows the microstrip routing for the edge pins per byte.

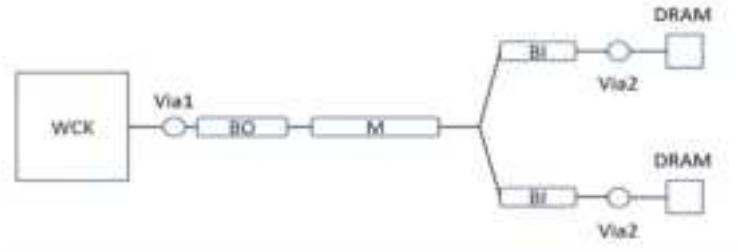
Figure 55. Microstrip Routing Topology for Data Signals Point-to-Point Topology



LPDDR5 interfaces do not support 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 added. The following two figures show the T-Line connection topology for WCK signal and CA, CLK, CTRL signals.

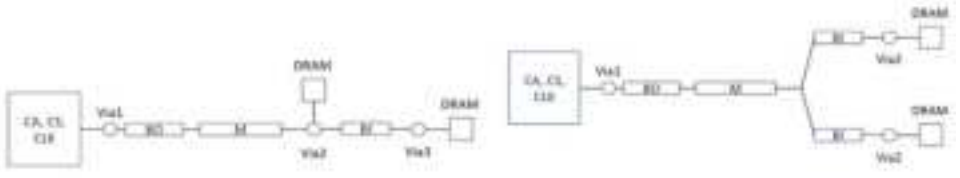
The figure below shows the stripline routing topology for inner pins.

Figure 56. Stripline Routing for WCK Signals T-Line Topology



The figure below show the daisy or T-Line connection topology for CA, CLK, and CTRL signals.

Figure 57. Stripline Routing for CA, CLK, CTRL Signals Daisy or T-Line Topology



The following tables provide a comprehensive routing guideline for each of the LPDDR5 signals based on memory down topology. For example, the trace impedance, the total trace length, the maximum length of the main trace routing can be derived by subtracting break-out and break-in trace segment length routed from total trace length. The signal trace width and minimum spacing and gaps (in mils) from edge to edge of signal traces are based on the default stackup.

The table below shows the stripline routing guidelinefor LPDDR5 memory down topology. The *h* value in the table represents the minimum substrate height from signal layer to reference layer.

Table 141. Stripline Routing Guideline for LPDDR5 Memory Down Topology

Signal Group	Signal Type	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
DQ, RDQS	(R)	SL	1000	4000		4	5, 17	17	17	4		17
	M	SL			40	5.2	6 (2h)	9 (5h)	9 (5h)	4	9 (5h)	9 (5h)
CA (Direct Connect)	R(D),R(C)	SL	400	4000		3	5, 17	17		4		17
	M	SL	2400		40	5.2		9 (5h)		4		9 (5h)
	(R)	SL	50			3		9 (5h)		4		9 (5h)
CA/CS/CKE (Daisy or T)	(M)	SL	500	4000		3	5, 17	17				
	M	SL	2000		40	5.2	6 (2h)	9 (5h)				
	(R)	SL	1000			3	6 (2h)	9 (5h)				
WCK	(R)	SL	1000	4000		4	5, 17	17		4		
	M	SL			40	5.2	6 (2h)	9 (5h)		4	9 (5h)	
	(R)	SL	(50 ~ 500)			3	6 (2h)	9 (5h)		4		

Memory Down Topology Guidelines	
Reference plane	Continuous Ground Only
Use 3x of Dielectric Height for serpentine routing spacing	
Typical PCB stackup assumed, and DQ/RDQS/WCK signal layer transition with vias only.	

The table below shows the microstrip routing for LPDDR5 memory down topology.

Table 142. Microstrip Routing Guideline for LPDDR5 Memory Down Topology

Signal Group	Signal Type	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
DQ, RDQS	(R)	US	350	3000		3	4	4	4	4		
	M	US			40	5.2	9 (5h)	9 (5h)	9 (5h)	4	12 (8h)	

Memory Down Topology Guidelines	
Reference plane	Continuous Ground Only
Use 3x of Dielectric Height for serpentine routing spacing	
For DQ/DQS signals routing for 6 or 8 layer board, please utilize internal routing layer first, if needed top routing layer can be used without board via transition. Avoid using bottom routing layer with via transition.	

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

Skew matching for LPDDR5 interfaces consists of both package routing skew and PCB physical routing skew. Skew matching of CA and CTRL with respect to the clock signals must be maintained 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 table below 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 stackups.

Table 143. Skew Matching Requirements 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	< 150mil	< 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		

The LPDDR5 eye margin is sensitive to crosstalk, especially when signals are routed on deep layers in stackup; the deep-layer vertical transition induces greater vertical coupling between signals and hence more crosstalk.

The maximum data rate of LPDDR5 is highly dependent on the DDR memory down configuration and on the type of PCB. For example, with a Type-III PCB, the LPDDR5 interface can run at the maximum data rate for a single-rank x 32 bit LPDDR5 memory down configuration. A reduced data transfer rate is seen whenever 2 x rank LPDDR5 memory down configuration is used.

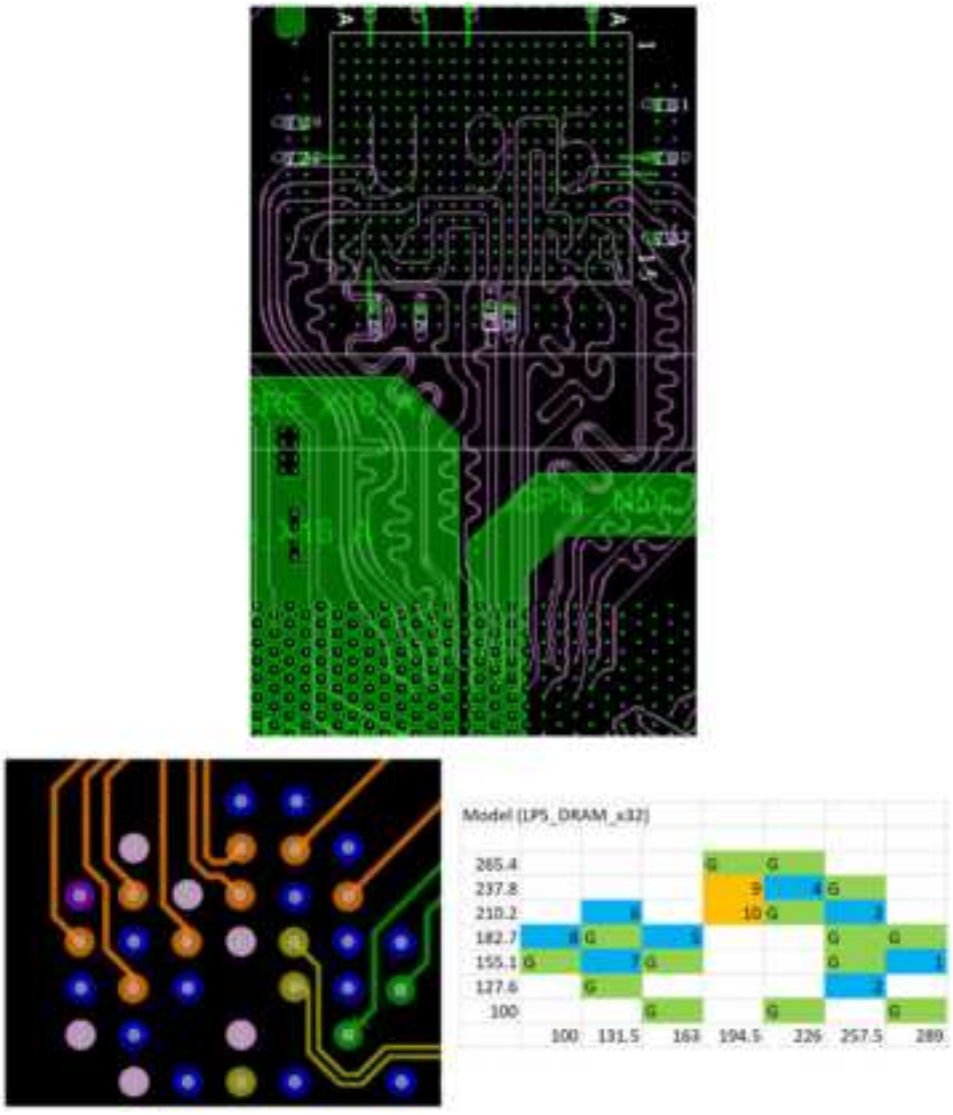
Table 144. LPDDR5 Maximum Data Rate w.r.t Memory and PCB configuration

Memory	LPDDR5	
Signal group	DQ	
Board thickness (mil)	65mil or 120mil (Routing must be on upper layers)	65mil or 120mil (Routing must be on upper layers)
PCB trace impedance (ohm)	Microstrip line: 45	Microstrip line: 45
	Stripline: 40	Stripline: 40
Memory configuration	Memory Down	Memory Down
Number of rank	1 (x16 bits or x32 bits-Double Die)	2 (x 16 or x 32 bits)
Maximum data rate (MT/s)	Refer to Agilex 5 data sheet	Refer to Agilex 5 data sheet
Maximum length total (inch)	Microstrip line: 3.0	Microstrip line: 3.0
	Stripline: 4.0	Stripline: 4.0

8.3.3.2. Example of an LPDDR5 Layout on an Intel FPGA Platform Board

The following figure shows the layout example of a single rank LPDDR5 x 32 bit device with pitch size of 0.7×0.8 mm in an Intel FPGA platform design. It uses a thick PCB (120mil stackup) with 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 66. Board Layout and Via Pattern for Single Rank LPDDR5 x32 Device on an Intel FPGA Platform Design



8.3.3.3. LPDDR5 Simulation Strategy

The simulation strategy is divided into 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 max 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 max vertical crosstalk between signals).

Intel recommends that a 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 performance on the layout, e.g the worst crosstalk (coupling between deep vertical vias), long trace routing and maximum reflection on routing path due to long via stubs if backdrilling is not applied.

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

You must perform the channel analysis in the time domain, using a pseudorandom binary sequence (PRBS) pattern for I/O signal generation, while the channel is built by using actual per-pin package models at both ends, including PCB model in the format of scattering parameter along with I/O buffer model at both ends. An I/O buffer IBIS model is used for DDR4 interface signal integrity simulation. Evaluate the eye diagram after the simulation, to ensure the eye specification is met at both ends.

9. Agilex 5 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 5 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 5 EMIF IP is somewhat simpler than that of some earlier device families, because Agilex 5 devices have more hardened blocks and fewer soft logic registers to be analyzed, because most are user logic registers.

Your Agilex 5 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.

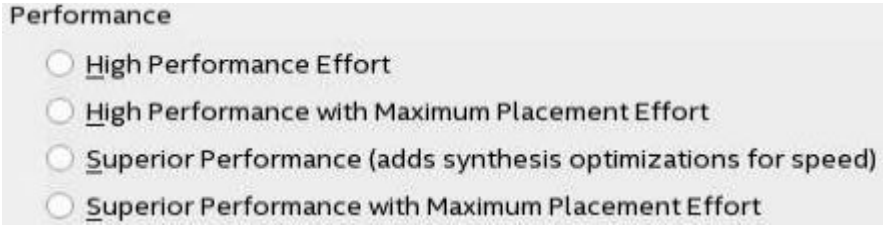
Core timing analysis excludes user logic timing to or from EMIF blocks. The EMIF IP provides a constrained clock (for example: ddr4_usr_clk) with which to clock customer logic; pll_afi_clk serves this purpose.

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`.

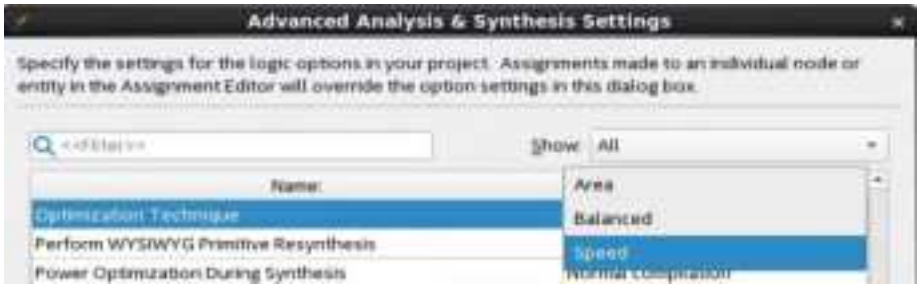
9.2. Optimizing Timing

The Quartus Prime software offers several advanced features that you can use to assist in meeting core timing requirements.

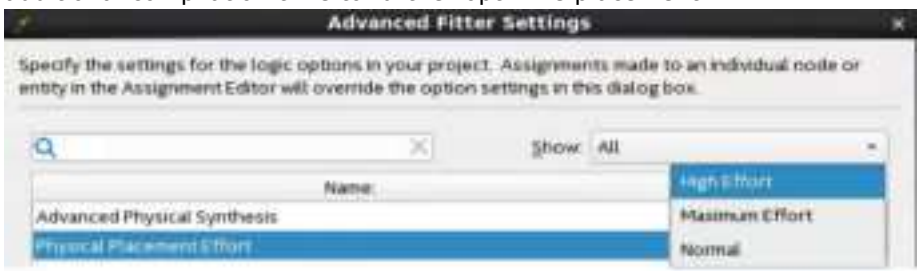
1. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings**. Under **Optimization mode**, select one of the **Performance** options.



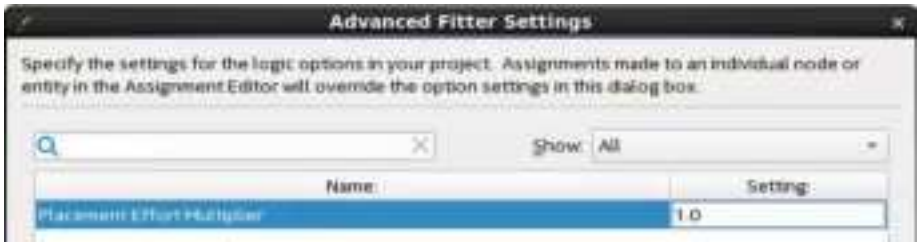
2. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings ➤ Advanced Settings (Synthesis).** For **Optimization Technique**, select **Speed**.



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 5 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 5 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. Generating Traffic with the Test Engine IP

Every Agilex 5 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 LPDDR4: 3
   - For LPDDR5: 1
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 LPDDR4: 3
   - For LPDDR5: 1
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.

11.6. 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.

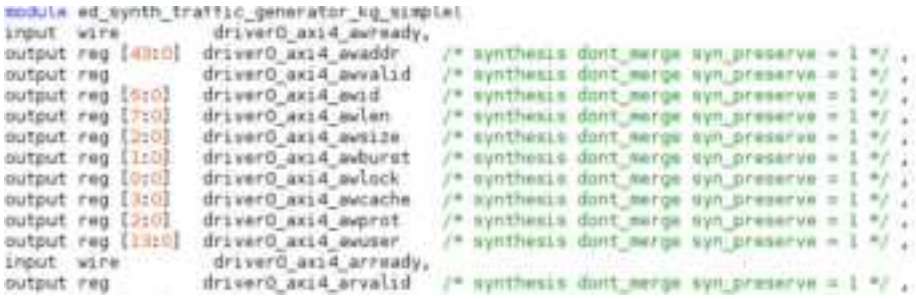
1. The traffic generator issues transactions only after calibration has completed successfully. You can check the calibration status by using the AXI-Lite interface. In the EMIF example design, the `cal_done_rst_n port` on the `ed_synth_axil_driver_0` corresponds to the calibration status. A value of `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 70. `cal_done_rst_n` in `ed_synth_axil_driver_0`

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 71. Specifying *dont_merge* and *preserve* Attributes to all Registers Driving AXI Port



11.7. Hardware Debugging Guidelines

Before debugging your design, confirm that it follows the recommended design flow. Refer to the *Agilex 5 EMIF IP Design Flow* section in chapter 1 of this user guide.

Always keep a record of tests, to avoid repeating the same tests later. To start debugging the design, perform the following initial steps.

11.8. Create a Simplified Design that Demonstrates the Same Issue

To help debugging, create a simple design that replicates the problem.

A simple design should compile quickly and be easy to understand. The EMIF IP generates an example top-level file that is ideal for debugging. The example top-level file uses all the same parameters, pin-outs, and so on.

11.9. Measure Power Distribution Network

Measure voltages of the various power supplies on their hardware development platform over a suitable time base and with a suitable trigger.

Ensure that you use an appropriate probe and grounding scheme. In addition, take the measurements directly on the pins or vias of the devices in question, and with the hardware operational.

Confirm that reference voltages (V_{REF_CA}) and termination voltages are active and within specification.

11.10. Measure Signal Integrity and Setup and Hold Margin

Measure the signals on the PCB. When measuring any signal, consider the edge rate of the signal, not just its frequency. Modern FPGA devices have very fast edge rates, therefore you must use a suitable oscilloscope, probe, and grounding scheme when you measure the signals.

You can take measurements to capture the setup and hold time of key signal classes with respect to their clock or strobe. Ensure that the measured setup and hold margin is at least better than that reported in the Quartus Prime software. A worse margin indicates a timing discrepancy somewhere in the project; however, this issue may not be the cause of your problem.

11.11. Vary Voltage

Vary the voltage of your system, if you suspect a marginality issue.

Increasing the voltage usually causes devices to operate faster and also usually provides increased noise margin.

11.12. Operate at a Lower Speed

Test the interface at a lower speed. If the interface works at a lower speed, the interface is correctly pinned out and functional.

If the interface fails at a lower speed, determine if the test is valid. Many high-speed memory components have a minimal operating frequency or require subtly different configurations when operating at a lower speeds.

For example, DDR4 SDRAM typically requires modification to the following parameters if you want to operate the interface at lower speeds:

- t_{MRD}
- t_{WTR}
- CAS latency and CAS write latency

11.13. Determine Whether the Issue Exists in Previous Versions of Software

Hardware that works before an update to either the Quartus Prime software or the memory IP indicates that the development platform is not the issue.

However, the previous generation IP may be less susceptible to a PCB issue, masking the issue.

11.14. Determine Whether the Issue Exists in the Current Version of Software

Designs are often tested using previous generations of Intel software or IP.

Projects may not be upgraded for various reasons:

- Multiple engineers are on the same project. To ensure compatibility, a common release of Intel software is used by all engineers for the duration of the product development. The design may be several releases behind the current Quartus Prime software version.
- Many companies delay before adopting a new release of software so that they can first monitor Internet forums to get a feel for how successful other users say the software is.
- Many companies never use the latest version of any software, preferring to wait until the first service pack is released that fixes the primary issues.
- Some users may only have a license for the older version of the software and can only use that version until their company makes the financial decision to upgrade.
- The local interface specification from Intel FPGA IP to the customer's logic sometimes changes from software release to software release. If you have already spent resources designing interface logic, you may be reluctant to repeat this exercise. If a block of code is already signed off, you may be reluctant to modify it to upgrade to newer IP from Intel.

In all of the above scenarios, you must determine if the issue still exists in the latest version of the Intel software. Bug fixes and enhancements are added to the Intel FPGA IP every release. Depending on the nature of the bug or enhancement, it may not always be clearly documented in the release notes.

Finally, if the latest version of the software resolves the issue, it may be easier to debug the version of software that you are using.

11.15. Try A Different PCB

If you are using the same Intel FPGA IP on several different hardware platforms, determine whether the problem occurs on all platforms or just on one.

Multiple instances of the same PCB, or multiple instances of the same interface, on physically different hardware platforms may exhibit different behavior. You can determine if the configuration is fundamentally not working, or if some form of marginality is involved in the issue.

Issues are often reported on the alpha build of a development platform. These are produced in very limited numbers and often have received limited bare-board testing, or functional testing. These early boards are often more unreliable than production quality PCBs.

Additionally, if the IP is from a previous project to help save development resources, determine whether the specific IP configuration works on a previous platform.

11.16. Try Other Configurations

Designs are often quite large, using multiple blocks of IP in many different combinations. Determine whether any other configurations work correctly on the development platform.

The full project may have multiple external memory controllers in the same device, or may have configurations where only half the memory width or frequency is required. Find out what does and does not work to help the debugging of the issue.

11.17. Debugging Checklist

The following checklist is a good starting point when debugging an external memory interface.

Table 145. Checklist

Check	Item
☐	Try a different fit.
☐	Check IP parameters at the operating frequency (t_{MRD}, t_{WTR} for example).
☐	Ensure you have constrained your design with proper timing deration and have closed timing.
☐	Simulate the design. If it fails in simulation, it will fail in hardware.
☐	Analyze timing.
☐	Place and assign R_{ZQ} (OCT).
☐	Measure the power distribution network (PDN).
☐	Measure signal integrity.
☐	Measure setup and hold timing.
☐	Measure FPGA voltages.
☐	Vary voltages.
☐	Heat and cool the PCB.
☐	Operate at a lower or higher frequency.
☐	Check board timing and trace Information.
☐	Check LVDS and clock sources, I/O voltages and termination.
☐	Check PLL clock source, specification, and jitter.
☐	Retarget to a smaller interface width or a single bank.

11.18. Categorizing Hardware Issues

The following topics divide issues into categories. By determining the category (or categories) in which an issue belongs, you may be able to better focus on the cause of the issue.

Hardware issues fall into three categories:

- Signal integrity issues
- Hardware and calibration issues
- Intermittent issues

11.19. Signal Integrity Issues

Many design issues, including some at the protocol layer, can be traced back to signal integrity problems. You should check circuit board construction, power systems, command, and data signaling to determine if they meet specifications.

If infrequent, random errors exist in the memory subsystem, product reliability suffers. Check the bare circuit board or PCB design file. Circuit board errors can cause poor signal integrity, signal loss, signal timing skew, and trace impedance mismatches. Differential traces with unbalanced lengths or signals that are routed too closely together can cause crosstalk.

11.20. Characteristics of Signal Integrity Issues

Signal integrity problems often appear when the performance of the hardware design is marginal.

The design may not always initialize and calibrate correctly, or may exhibit occasional bit errors in user mode. Severe signal integrity issues can result in total failure of an interface at certain data rates, and sporadic component failure because of electrical stress. PCB component variance and signal integrity issues often show up as failures on one PCB, but not on another identical board. Timing issues can have a similar characteristic. Multiple calibration windows or significant differences in the calibration results from one calibration to another can also indicate signal integrity issues.

11.21. Evaluating Signal Integrity Issues

Signal integrity problems can only really be evaluated in two ways:

- direct measurement using suitable test equipment like an oscilloscope and probe
- simulation using a tool like HyperLynx or Allegro PCB SI

Compare signals to the respective electrical specification. You should look for overshoot and undershoot, non-monotonicity, eye height and width, and crosstalk.

11.22. Skew

Ensure that all clocked signals, commands, addresses, and control signals arrive at the memory inputs at the same time.

Trace length variations cause data valid window variations between the signals, reducing margin. For example, DDR4-3200 at 1600 MHz has a data valid window that is smaller than 313 ps. Trace length skew or crosstalk can reduce this data valid window further, making it difficult to design a reliably operating memory interface. Ensure that the skew figure previously entered into the Intel FPGA IP matches that actually achieved on the PCB, otherwise Quartus Prime timing analysis of the interface is accurate.

11.23. Crosstalk

Crosstalk is best evaluated early in the memory design phase.

Check the clock-to-data strobes, because they are bidirectional. Measure the crosstalk at both ends of the line. Check the data strobes to clock, because the clocks are unidirectional, these only need checking at the memory end of the line.

11.24. Power System

Some memory interfaces draw current in spikes from their power delivery system as SDRAMs are based on capacitive memory cells.

Rows are read and refreshed one at a time, which causes dynamic currents that can stress any power distribution network (PDN). The various power rails should be checked either at or as close as possible to the SDRAM power pins. Ideally, you should use a real-time oscilloscope set to fast glitch triggering to check the power rails.

11.25. Clock Signals

The clock signal quality is important for any external memory system.

Measurements include frequency, digital core design (DCD), high width, low width, amplitude, jitter, rise, and fall times.

11.26. Address and Command Signals

Confirm that address and command signals are reaching the memory devices correctly.

For example, if you are targeting DDR4, you can probe the ALERT_N pin after the memory interface has been successfully calibrated, to determine if any memory component has encountered an address and command parity error.

11.27. Read Data Valid Window and Eye Diagram

The memory generates the read signals. Take measurements at the FPGA end of the line.

To ease read diagram capture, modify the example driver to mask writes or modify the PHY to include a signal that you can trigger on when performing reads.

11.28. Write Data Valid Window and Eye Diagram

The FPGA generates the write signals. Take measurements at the memory device end of the line.

To ease write diagram capture, modify the example driver to mask reads or modify the PHY export a signal that is asserted when performing writes.

For the FPGA, ensure that you perform the following:

- Connect the R_{ZQ} pin to the correct resistors and pull-down to ground in the schematic or PCB.
- Contain the R_{ZQ} pins within a bank of the device that is operating at the same VCCIO voltage as the interface that is terminated.
- Review the Fitter Pin-Out file for R_{ZQ} pins to ensure that they are on the correct pins, and that only the correct number of calibration blocks exists in your design.
- Check in the fitter report that the input, output, and bidirectional signals with calibrated OCT all have the termination control block applicable to the associated R_{ZQ} pins.

For the memory components, ensure that you perform the following:

- Connect the required resistor to the correct pin on each and every component, and ensure that it is pulled to the correct voltage.
- Place the required resistor close to the memory component.
- Correctly configure the IP to enable the desired termination at initialization time.
- Check that the speed grade of memory component supports the selected ODT setting.
- Check that the second source part that may have been fitted to the PCB, supports the same ODT settings as the original.

11.29. Hardware and Calibration Issues

Hardware and calibration issues have the following definitions:

- Calibration issues result in calibration failure.
- Hardware issues result in read and write failures.

Note: Ensure that functional, timing, and signal integrity issues are not the direct cause of your hardware issue, as functional, timing or signal integrity issues are usually the cause of any hardware issue.

11.30. Memory Timing Parameter Evaluation

Review and update the memory timing parameters, CAS, and Write CAS latency based on the speed bin of the targeted memory component and the operating frequency of your interface.

Incorrect memory timing parameters, CAS, or Write CAS latency can cause data corruption in the memory component.

11.31. Verify that the Board Has the Correct Memory Component or DIMM Installed

Verify that the correct memory component or DIMM is installed on the circuit board.

If an incorrect memory part is used, the memory part may not support the memory timing parameters, CAS, or Write CAS latency used in parameterizing the IP; this situation can result in data corruption. If a memory device with smaller memory capacity is installed incorrectly, data written to the higher address space overwrites the data at the lower address space.

12. Document Revision History for External Memory Interfaces (EMIF) IP User Guide

Document Version	Quartus Prime Version	IP Version	Changes
2024.04.01	24.1	6.1.0	Initial release.