

External Memory Interfaces (EMIF) IP User Guide

AgilexTM 5 FPGAs and SoCs

Updated for Quartus® Prime Design Suite: **25.1**

IP Version: **3.0.0**



Online Version

Send Feedback

817467

2025.03.31

Contents

1. About the External Memory Interfaces Agilex™ 5 FPGA IP.....	9
1.1. Release Information.....	9
2. Agilex™ 5 FPGA EMIF IP – Introduction.....	10
2.1. Agilex 5 EMIF IP Protocol and Feature Support.....	10
2.2. Agilex 5 EMIF IP Design Flow.....	10
2.2.1. Agilex 5 EMIF IP Design Checklist.....	11
3. Agilex 5 FPGA EMIF IP – Product Architecture.....	13
3.1. Agilex 5 EMIF Architecture: Protocol and Maximum Interface Width Support.....	13
3.2. Agilex 5 EMIF Architecture: Introduction.....	14
3.2.1. Agilex 5 EMIF Architecture: I/O Subsystem.....	15
3.2.2. Agilex 5 EMIF Architecture: I/O SSM.....	16
3.2.3. Agilex 5 EMIF Architecture: HSIO Bank.....	16
3.2.4. Agilex 5 EMIF Architecture: I/O Lane.....	35
3.2.5. Agilex 5 EMIF Architecture: Input DQS Clock Tree.....	37
3.2.6. Agilex 5 EMIF Architecture: PHY Clock Tree.....	38
3.2.7. Agilex 5 EMIF Architecture: PLL Reference Clock Networks.....	39
3.2.8. Agilex 5 EMIF Architecture: Clock Phase Alignment.....	40
3.2.9. User Clock in Different Core Access Modes.....	41
3.3. Agilex 5 EMIF Sequencer.....	42
3.4. Agilex 5 EMIF Controller.....	43
3.4.1. Hard Memory Controller.....	43
3.5. Agilex 5 EMIF IP for Hard Processor Subsystem (HPS).....	44
3.5.1. Restrictions on I/O Bank Usage for EMIF IP with HPS.....	49
3.5.2. Using the Legacy EMIF Debug Toolkit with Agilex 5 HPS Interfaces.....	50
4. Agilex 5 FPGA EMIF IP – End-User Signals.....	51
4.1. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component	51
4.1.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	52
4.1.2. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	52
4.1.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	52
4.1.4. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	52
4.1.5. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	53
4.1.6. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	54
4.1.7. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	54
4.1.8. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	54
4.1.9. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	55

4.1.10. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	56
4.1.11. mem_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	56
4.1.12. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	56
4.1.13. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component.....	56
4.2. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component	57
4.2.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	58
4.2.2. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	58
4.2.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	58
4.2.4. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	58
4.2.5. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	59
4.2.6. s1_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	60
4.2.7. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	61
4.2.8. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	61
4.2.9. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	62
4.2.10. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	62
4.2.11. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	63
4.2.12. mem_reset_n_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	63
4.2.13. mem_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	63
4.2.14. mem_ck_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	64
4.2.15. mem_reset_n_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	64
4.2.16. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	64
4.2.17. oct_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	64
4.2.18. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component.....	65
4.3. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4	65
4.3.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	66
4.3.2. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	67
4.3.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	67

4.3.4. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	67
4.3.5. s1_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	67
4.3.6. s1_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	68
4.3.7. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	68
4.3.8. s1_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	69
4.3.9. s2_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	70
4.3.10. s3_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	72
4.3.11. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	73
4.3.12. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	73
4.3.13. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	73
4.3.14. s1_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	74
4.3.15. s1_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	74
4.3.16. s1_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	75
4.3.17. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	75
4.3.18. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	76
4.3.19. mem_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	76
4.3.20. mem_ck_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	76
4.3.21. mem_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	77
4.3.22. mem_ck_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	77
4.3.23. mem_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	77
4.3.24. mem_ck_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	78
4.3.25. mem_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	78
4.3.26. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	78
4.3.27. oct_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	79
4.3.28. oct_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	79
4.3.29. oct_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	79
4.3.30. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4.....	79
4.4. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5	80
4.4.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	81
4.4.2. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	81

4.4.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	81
4.4.4. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	82
4.4.5. s1_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	82
4.4.6. s1_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	82
4.4.7. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.	83
4.4.8. s1_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.	84
4.4.9. s2_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.	85
4.4.10. s3_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	86
4.4.11. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	88
4.4.12. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	88
4.4.13. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	88
4.4.14. s1_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	89
4.4.15. s1_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	89
4.4.16. s1_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	89
4.4.17. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	90
4.4.18. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	90
4.4.19. mem_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	91
4.4.20. mem_ck_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	91
4.4.21. mem_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	91
4.4.22. mem_ck_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	92
4.4.23. mem_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	92
4.4.24. mem_ck_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	93
4.4.25. mem_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	93
4.4.26. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5...	93
4.4.27. oct_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5...	93
4.4.28. oct_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5...	94
4.4.29. oct_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5...	94
4.4.30. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5.....	94
5. Agilex 5 FPGA EMIF IP – Simulating Memory IP.....	95
5.1. Simulation Walkthrough.....	95
5.1.1. Calibration	96

5.1.2. Simulation Scripts.....	96
5.1.3. Functional Simulation with Verilog HDL.....	96
5.1.4. Simulating the Design Example.....	97
6. Agilex 5 FPGA EMIF IP - DDR4 Support.....	100
6.1. External Memory Interfaces (EMIF) IP - DDR4 Component Parameter Descriptions.....	100
6.2. External Memory Interfaces (EMIF) IP - DDR4 DIMM Parameter Descriptions.....	109
6.3. Agilex 5 FPGA EMIF IP Pin and Resource Planning.....	118
6.3.1. Agilex 5 FPGA EMIF IP Resources.....	119
6.3.2. Pin Guidelines for Agilex 5 FPGA EMIF IP.....	119
6.3.3. Pin Placements for Agilex 5 FPGA DDR4 EMIF IP.....	122
6.4. Agilex 5 EMIF Pin Swapping Guidelines.....	137
6.4.1. DDR4 Byte Lane Swapping.....	137
6.4.2. DDR4 Address and Command and CLK Lane.....	140
6.4.3. DDR4 Interface x8 Data Lane.....	140
6.4.4. DDR4 Interface x4 Data Lane.....	141
7. Agilex 5 FPGA EMIF IP - DDR5 Support.....	144
7.1. External Memory Interfaces (EMIF) IP - DDR5 Component Parameter Descriptions.....	144
7.2. External Memory Interfaces (EMIF) IP - DDR5 DIMM Parameter Descriptions.....	154
7.3. Agilex 5 FPGA EMIF IP Pin and Resource Planning.....	165
7.3.1. Agilex 5 FPGA EMIF IP Interface Pins.....	165
7.3.2. Agilex 5 FPGA EMIF IP Resources.....	168
7.3.3. Pin Guidelines for Agilex 5 FPGA EMIF IP.....	168
7.3.4. Pin Placements for Agilex 5 FPGA DDR5 EMIF IP.....	174
7.3.5. Agilex 5 EMIF Pin Swapping Guidelines.....	180
8. Agilex 5 FPGA EMIF IP - LPDDR4 Support.....	185
8.1. External Memory Interfaces (EMIF) IP - LPDDR4 Parameter Descriptions.....	185
8.2. Agilex 5 FPGA EMIF IP Pin and Resource Planning.....	193
8.2.1. Agilex 5 FPGA EMIF IP Interface Pins.....	193
8.2.2. Agilex 5 FPGA EMIF IP Resources.....	197
8.2.3. Pin Guidelines for Agilex 5 FPGA EMIF IP.....	198
9. Agilex 5 FPGA EMIF IP - LPDDR5 Support.....	206
9.1. External Memory Interfaces (EMIF) IP - LPDDR5 Parameter Descriptions.....	206
9.2. Agilex 5 FPGA EMIF IP Pin and Resource Planning.....	218
9.2.1. Agilex 5 FPGA EMIF IP Interface Pins.....	219
9.2.2. Agilex 5 FPGA EMIF IP Resources.....	222
9.2.3. Pin Guidelines for Agilex 5 FPGA EMIF IP.....	223
9.2.4. Pin Placements for Agilex 5 FPGA LPDDR5 EMIF IP.....	226
10. Agilex 5 FPGA EMIF IP – Timing Closure.....	231
10.1. Timing Closure	231
10.1.1. Timing Analysis.....	231
10.2. Optimizing Timing.....	232
11. Agilex 5 FPGA EMIF IP – Controller Optimization.....	234
11.1. Optimizing Efficiency for Secondary Controller.....	234
11.2. Interface Standard.....	236
11.3. Bank Management Efficiency.....	236
11.4. Data Transfer.....	237
11.5. Improving Controller Efficiency.....	237

11.5.1. Frequency of Operation.....	237
11.5.2. Series of Reads or Writes.....	237
11.5.3. AXI to Memory Mapping.....	238
12. Agilex 5 FPGA EMIF IP – Debugging.....	240
12.1. Interface Configuration Performance Issues.....	240
12.1.1. Interface Configuration Bottleneck and Efficiency Issues.....	240
12.2. Functional Issue Evaluation.....	241
12.2.1. Altera IP Memory Model.....	241
12.2.2. Vendor Memory Model.....	242
12.2.3. Transcript Window Messages.....	242
12.3. Timing Issue Characteristics.....	244
12.3.1. Evaluating FPGA Timing Issues.....	244
12.3.2. Evaluating External Memory Interface Timing Issues.....	245
12.4. Verifying Memory IP Using the Signal Tap Logic Analyzer.....	246
12.5. Debugging with the External Memory Interface Debug Toolkit.....	246
12.5.1. Prerequisites for Using the EMIF Debug Toolkit.....	247
12.5.2. Configuring a Design to Use the Debug Toolkit.....	247
12.5.3. Launching the EMIF Debug Toolkit.....	248
12.5.4. Using the EMIF Debug Toolkit.....	250
12.6. Generating Traffic with the Test Engine IP.....	251
12.7. Guidelines for Developing HDL for Traffic Generator.....	252
12.8. Guidelines for Traffic Generator Status Check.....	253
12.8.1. Status Check Using the Signal Tap Logic Analyzer.....	253
12.8.2. Exporting the Status Interface to the Top-Level Design.....	254
12.9. Hardware Debugging Guidelines.....	256
12.9.1. Create a Simplified Design that Demonstrates the Same Issue.....	257
12.9.2. Measure Power Distribution Network.....	257
12.9.3. Measure Signal Integrity and Setup and Hold Margin.....	257
12.9.4. Vary Voltage.....	257
12.9.5. Operate at a Lower Speed.....	257
12.9.6. Determine Whether the Issue Exists in Previous Versions of Software.....	258
12.9.7. Determine Whether the Issue Exists in the Current Version of Software.....	258
12.9.8. Try A Different PCB.....	258
12.9.9. Try Other Configurations.....	259
12.9.10. Debugging Checklist.....	259
12.10. Categorizing Hardware Issues.....	260
12.10.1. Signal Integrity Issues.....	260
12.10.2. Hardware and Calibration Issues.....	262
13. Agilex 5 FPGA EMIF IP - Mailbox Support.....	266
13.1. Agilex 5 Mailbox Structure and Register Definitions.....	266
13.1.1. Mailbox Supported Commands.....	268
13.1.2. Mailbox Command Definitions.....	274
13.1.3. ECC Syndrome Codes.....	283
13.1.4. ECC Error Handling.....	284
13.2. Accessing Read-Only Registers.....	287
13.2.1. Example 1: Reading IP_TYPE and IP_INSTANCE_ID of All the Interfaces in the IO96B Through Read-Only Registers	287
13.2.2. Example 2: Reading the Memory Clock Frequency for an Interface.....	288
13.3. Sending a Mailbox Command.....	288

13.3.1. Example 3: Sending Recalibration Request and Reading Calibration Status Using the Mailbox	289
13.4. IOSSM Mailbox Access Script.....	294
13.4.1. Mailbox Script Overview.....	294
13.4.2. Running the Mailbox Script.....	299
14. Document Revision History for External Memory Interfaces (EMIF) IP User Guide...	300

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 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.

IP Name	IP Version	Quartus Prime	Release Date
External Memory Interfaces (EMIF) IP - DDR4 Component	3.0.0	25.1	2025.03.31
External Memory Interfaces (EMIF) IP - DDR4 DIMM	3.0.0	25.1	2025.03.31
External Memory Interfaces (EMIF) IP - DDR5 Component	3.0.0	25.1	2025.03.31
External Memory Interfaces (EMIF) IP - DDR5 DIMM	3.0.0	25.1	2025.03.31
External Memory Interfaces (EMIF) IP - LPDDR4 Component	3.0.0	25.1	2025.03.31
External Memory Interfaces (EMIF) IP - LPDDR5 Component	3.0.0	25.1	2025.03.31



2. Agilex™ 5 FPGA EMIF IP – Introduction

Altera'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

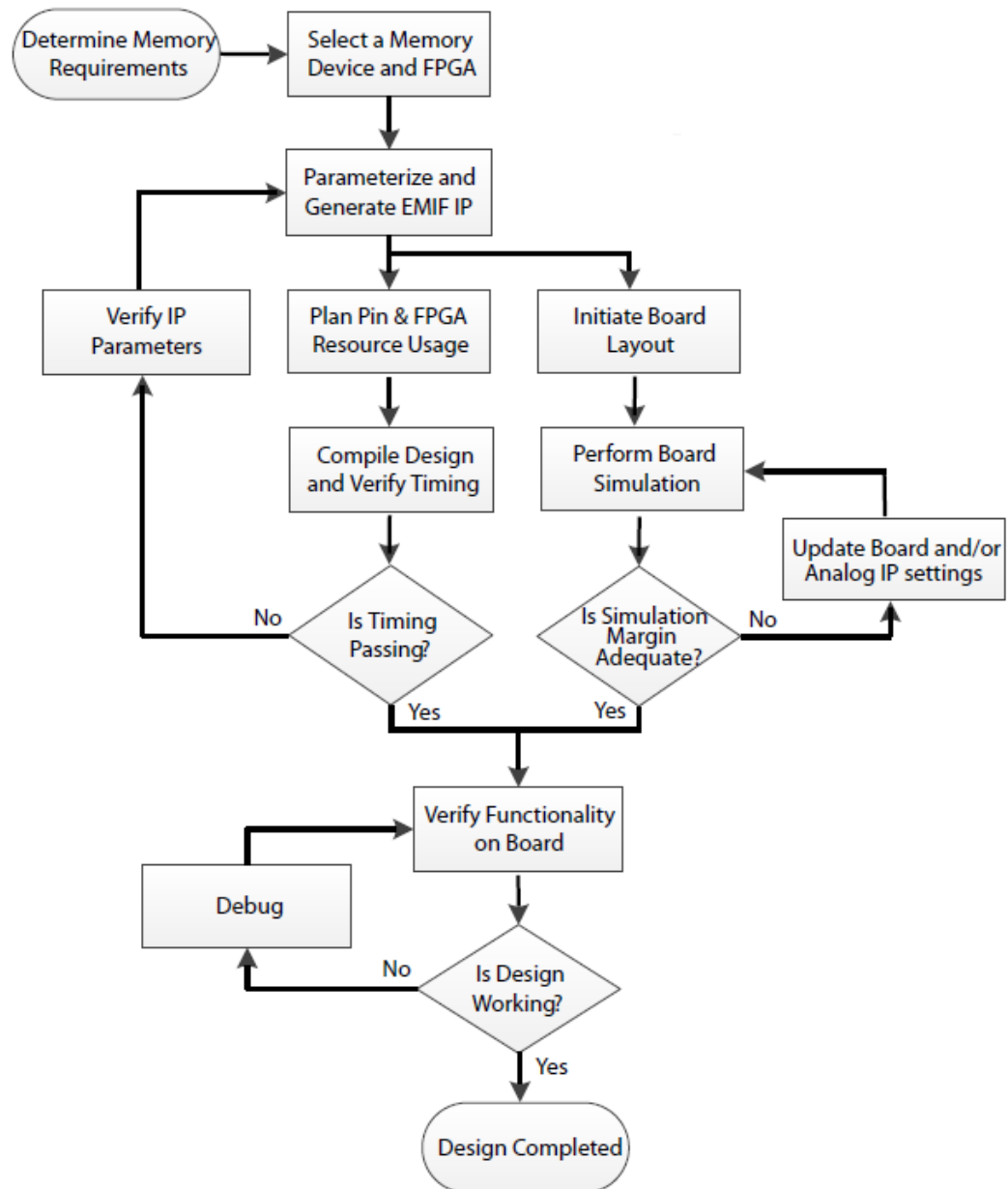
- The Agilex 5 FPGA EMIF IP supports DDR4 with hard memory controller and hard PHY.
- The Agilex 5 FPGA EMIF IP supports LPDDR4 with hard memory controller and hard PHY.
- The Agilex 5 FPGA EMIF IP supports LPDDR5 with hard memory controller and hard PHY.
- The Agilex 5 FPGA EMIF IP supports DDR5 with hard memory controller and hard PHY (for D-Series and E-Series Device Group A).

2.2. Agilex 5 EMIF IP Design Flow

Altera 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 1. EMIF Design Checklist

Design Step	Description	Resources
Select an FPGA	Not all Altera FPGAs support all memory types and configurations. To help with the FPGA selection process, refer to the resources listed in the right column.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> DDR4 Component Parameter Descriptions DDR4 DIMM Parameter Descriptions DDR5 Component Parameter Descriptions DDR5 DIMM 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> DDR4 Component Parameter Descriptions DDR4 DIMM Parameter Descriptions DDR5 Component Parameter Descriptions DDR5 DIMM 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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 2. 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 3. 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, DDR5, and LPDDR5 memory protocols.
 - LPDDR5 on Agilex 5 E-Series Device Group B can support only one frequency set point (FSP0).
 - You can only use asynchronous fabric clocking mode with the DDR5 protocol on -3 speed bin for E-Series Device Group A and D-Series devices.

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, DDR5, 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.

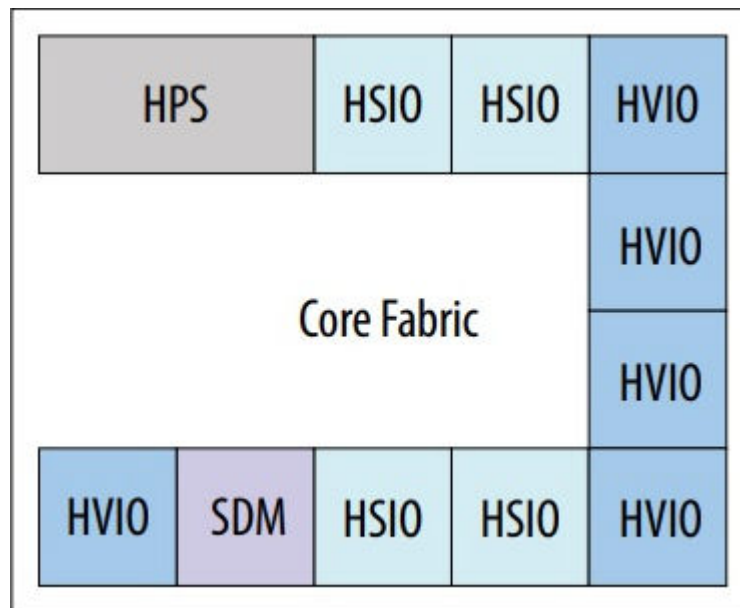
3.2.1. Agilex 5 EMIF Architecture: I/O Subsystem

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.

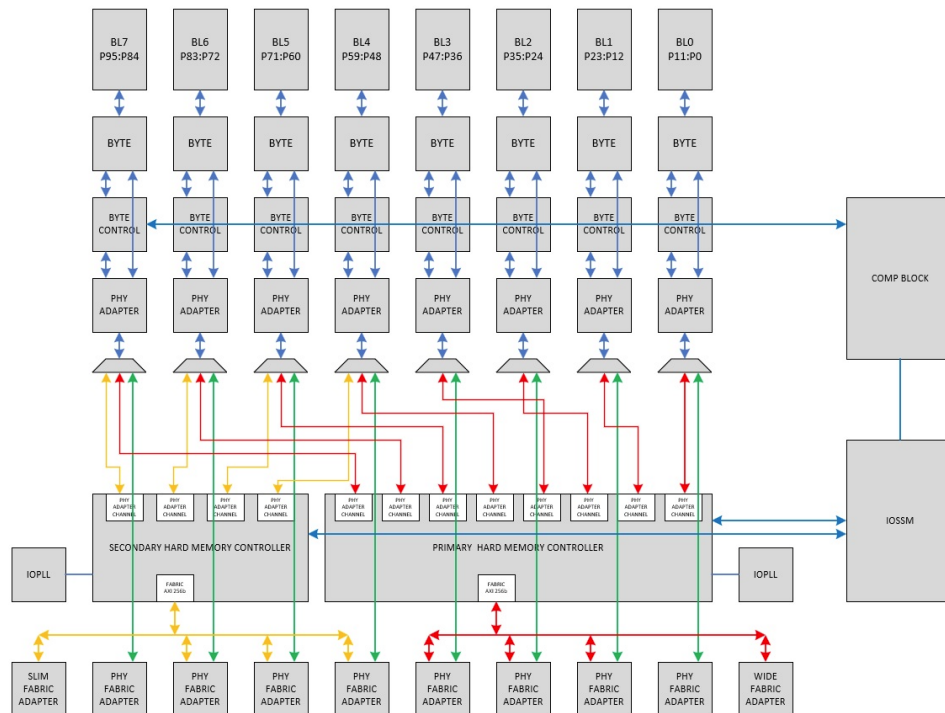
3.2.3. Agilex 5 EMIF Architecture: HSIO Bank

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. Lockstep Configuration

To support user data widths greater than 32 bits in DDR4 or DDR5 on Agilex 5 D-Series devices, the external memory interface (EMIF) IP instantiates multiple memory controllers driven in lockstep. In lockstep configurations, multiple controllers run the same set of operations simultaneously in parallel.

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

The following table summarizes the supported lockstep configurations:

Table 5. Supported Lockstep Controller Configurations

Memory Protocol	Configuration	DQ Width	AXI Interface
DDR4/DDR5	x40	40	256 b + 64 b USER DATA
DDR4	x64	64	512 b
DDR4	x64 with ECC	72	512 b
DDR4	x72	72	512 b + 64 b USER DATA
DDR5 RDIMM	2 ch x40	40	512 b + 64 b USER DATA

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

- Lockstep configurations support only synchronous fabric clocking mode.
- Some of the controller scheduling and optimization features are disabled.
- There are limitations on the types of AXI transactions supported. Refer to the following table for details.

Table 6. Limitations on AXI Transactions on Lockstep Configuration

Supported Protocol	Device	Configuration	Known Limitations
DDR4	Agilex 5 D-Series	x72	<ul style="list-style-type: none"> — AXI transfer size min = 3 (8 bytes + 1 byte WUSER/RUSER) — Supports 8-byte aligned transfers only
DDR4	Agilex 5 D-Series	x64	<ul style="list-style-type: none"> — AXI transfer size min = 1 (2 bytes) — Supports 2-byte aligned transfers only
DDR4/DDR5	Agilex 5 D-Series	x40	<ul style="list-style-type: none"> — AXI transfer size min = 2 (4 bytes + 1 byte RUSER/WUSER) — Supports 32-bit aligned transfers only

Narrow Read Transfer Support

Narrow read transfer happens when the AXI master generates a read data transfer that is narrower than its data bus width. For example, for a 256-bit AXI data bus, a narrow read transfer happens when ARSIZE is less than 5.

Table 7. Limitations on Narrow Read Transfer

Configuration	Controller	OPN	Narrow Read Transfer Support
Lockstep	Both primary and secondary controllers.	All Agilex 5 D-Series OPNs	<ul style="list-style-type: none"> Not supported. Must perform full-width read data transfer. Can support up to 6-bit of ARID/AWID for lockstep configuration that uses both Primary and Secondary Controller (x40 Lockstep, x64+ECC Lockstep and x72 Lockstep).
	Both Primary	All Agilex 5 D-Series OPNs	<ul style="list-style-type: none"> Not supported. Must perform full-width read data transfer. Can support up to 7-bit of ARID/AWID for lockstep configuration that uses Primary Controller only (x64 Lockstep).
Non-Lockstep	Both primary and secondary controllers.	<ul style="list-style-type: none"> A5EC065BBXXAEXSR0 A5ED065BBXXAEXSR0 	Not supported. Must perform full-width read data transfer.
	Primary controller.	All Agilex 5 D-Series and E-Series except the following: <ul style="list-style-type: none"> A5EC065BBXXAEXSR0 A5ED065BBXXAEXSR0 	Supported. Up to 7-bit of ARID/AWID.
	Secondary controller.		Supported. Up to 6-bit of ARID/AWID.

Lockstep configuration does not support narrow transfer. For non-lockstep configurations, the following OPN do not support narrow read transfer:

- A5EC065BBXXAEXSR0
- A5ED065BBXXAEXSR0

Unaligned Memory Access

Unaligned memory access is not supported on Agilex 5 EMIF interfaces. For Fabric EMIF, unaligned memory access is not supported in Fabric Direct Access Mode.

For HPS EMIF, unaligned memory access is not supported on the F2SDRAM Bridge. Unaligned memory access happens when the ARADDR/AWADDR is not a multiple of the data bus width (in bytes).

The following table shows the AWADDR/ARADDR requirement for aligned memory access in Fabric EMIF and HPS EMIF.

Table 8. AWADDR/ARADDR Requirement for Aligned Memory Access

Category	AXI Data Width (Bits)	AXI Data Width (Bytes)	AWADDR/ARADDR for Aligned Access
Fabric EMIF-Direct Access Mode	256	32	32n
HPS EMIF - F2SDRAM	64	8	No Requirement; any access allowed.
<i>continued...</i>			

Category	AXI Data Width (Bits)	AXI Data Width (Bytes)	AWADDR/ARADDR for Aligned Access
HPS EMIF - F2SDRAM	128	16	No Requirement; any access allowed.
HPS EMIF - F2SDRAM	256	32	32n

Note: n is an integer value equal to or greater than 0.

Performing unaligned memory access on the EMIF interfaces might result in a data error.

Two Controllers in Lockstep Within One IO96 Bank

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

Table 9. Mapping of WDATA/RDATA & WUSER/RUSER in x40 Configuration

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

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

Table 10. Supported Byte Lane Placement for x40 configuration

Scheme	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
DDR4_AC_ TOP	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]
DDR4_AC_ BOT	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	sDQ[4]

Two/Three Controllers in Lockstep Within One IO96 Bank

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

Table 11. Supported 2/3 Controller Configurations in Lockstep

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

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

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

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

Table 12. Example Mapping of WDATA/RDATA & WUSER/RUSER in x72 Configuration

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

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

Figure 4.

Configuration	x64		3AC -x72 or x64+ECC		4AC -x72 or x64+ECC	
mem_DQ0	31:0		39:0		31:0	
mem_DQ1		31:0		31:0		39:0
	↓	↓	↓	↓	↓	↓
mem_DQ	31:0	63:32	39:0	71:40	31:0	63:32

3.2.3.2. Pin Placement

Table 13. DDR4 Pin Placement

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

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
BL4	60	MEM_DQ[16]	MEM_DQ[16]	MEM_DQ[16] *	
	59	MEM_DQ[15]	MEM_DQ[15]	MEM_DQ[15]	MEM_DQ[15]
	58	MEM_DQ[14]	MEM_DQ[14]	MEM_DQ[14]	MEM_DQ[14]
	57	MEM_DQ[13]	MEM_DQ[13]	MEM_DQ[13]	MEM_DQ[13]
	56	MEM_DQ[12]	MEM_DQ[12]	MEM_DQ[12]	MEM_DQ[12]
	55				
	54	MEM_DM_N[1]	MEM_DM_N[1]	MEM_DM_N[1]	MEM_DM_N[1]
	53	MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_DQS_C[1]
	52	MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_DQS_T[1]
	51	MEM_DQ[11]	MEM_DQ[11]	MEM_DQ[11]	MEM_DQ[11]
	50	MEM_DQ[10]	MEM_DQ[10]	MEM_DQ[10]	MEM_DQ[10]
	49	MEM_DQ[9]	MEM_DQ[9]	MEM_DQ[9]	MEM_DQ[9]
	48	MEM_DQ[8]	MEM_DQ[8]	MEM_DQ[8]	MEM_DQ[8]
BL3	47	MEM_BG[0]	MEM_BG[0]	MEM_BG[0]	MEM_BG[0]
	46	MEM_BA[1]	MEM_BA[1]	MEM_BA[1]	MEM_BA[1]
	45	MEM_BA[0]	MEM_BA[0]	MEM_BA[0]	MEM_BA[0]
	44	MEM_ALERT_N[0]	MEM_ALERT_N[0]	MEM_ALERT_N[0]	MEM_ALERT_N[0]
	43	MEM_A[16]	MEM_A[16]	MEM_A[16]	MEM_A[16]
	42	MEM_A[15]	MEM_A[15]	MEM_A[15]	MEM_A[15]
	41	MEM_A[14]	MEM_A[14]	MEM_A[14]	MEM_A[14]
	40	MEM_A[13]	MEM_A[13]	MEM_A[13]	MEM_A[13]
	39	MEM_A[12]	MEM_A[12]	MEM_A[12]	MEM_A[12]
	38	RZQ Site	RZQ Site	RZQ Site	RZQ Site
	37	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site
	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]
continued...					

Lane Number	Pin Index	x32+ECC *	x 32	x16 + ECC *	x16
	26	MEM_A[2]	MEM_A[2]	MEM_A[2]	MEM_A[2]
	25	MEM_A[1]	MEM_A[1]	MEM_A[1]	MEM_A[1]
	24	MEM_A[0]	MEM_A[0]	MEM_A[0]	MEM_A[0]
BL1	23	MEM_PAR[0]	MEM_PAR[0]	MEM_PAR[0]	MEM_PAR[0]
	22	MEM_CS_N[1]	MEM_CS_N[1]	MEM_CS_N[1]	MEM_CS_N[1]
	21	MEM_CK_C[0]	MEM_CK_C[0]	MEM_CK_C[0]	MEM_CK_C[0]
	20	MEM_CK_T[0]	MEM_CK_T[0]	MEM_CK_T[0]	MEM_CK_T[0]
	19	MEM_CKE[1]	MEM_CKE[1]	MEM_CKE[1]	MEM_CKE[1]
	18	MEM_CKE[0]	MEM_CKE[0]	MEM_CKE[0]	MEM_CKE[0]
	17	MEM_ODT[1]	MEM_ODT[1]	MEM_ODT[1]	MEM_ODT[1]
	16	MEM_ODT[0]	MEM_ODT[0]	MEM_ODT[0]	MEM_ODT[0]
	15	MEM_ACT_N[0]	MEM_ACT_N[0]	MEM_ACT_N[0]	MEM_ACT_N[0]
	14	MEN_CS_N[0]	MEN_CS_N[0]	MEN_CS_N[0]	MEN_CS_N[0]
	13	MEM_RESET_N[0]	MEM_RESET_N[0]	MEM_RESET_N[0]	MEM_RESET_N[0]
	12	MEM_BG[1]	MEM_BG[1]	MEM_BG[1]	MEM_BG[1]
BL0	11	MEM_DQ[7]	MEM_DQ[7]	MEM_DQ[7]	MEM_DQ[7]
	10	MEM_DQ[6]	MEM_DQ[6]	MEM_DQ[6]	MEM_DQ[6]
	9	MEM_DQ[5]	MEM_DQ[5]	MEM_DQ[5]	MEM_DQ[5]
	8	MEM_DQ[4]	MEM_DQ[4]	MEM_DQ[4]	MEM_DQ[4]
	7				
	6	MEM_DM_N[0]	MEM_DM_N[0]	MEM_DM_N[0]	MEM_DM_N[0]
	5	MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_DQS_C[0]
	4	MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_DQS_T[0]
	3	MEM_DQ[3]	MEM_DQ[3]	MEM_DQ[3]	MEM_DQ[3]
	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 14. DDR5 Pin Placement

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
BL7	95			MEM_1_MEM_DQ[15]		
	94			MEM_1_MEM_DQ[14]		
	93			MEM_1_MEM_DQ[13]		
<i>continued...</i>						

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	92			MEM_1_MEM_DQ[12]		
	91					
	90			MEM_1_MEM_DM_N[1]		
	89			MEM_1_MEM_DQS_C[1]		
	88			MEM_1_MEM_DQS_T[1]		
	87			MEM_1_MEM_DQ[11]		
	86			MEM_1_MEM_DQ[10]		
	85			MEM_1_MEM_DQ[9]		
	84			MEM_1_MEM_DQ[8]		
BL6	83	MEM_DQ[39]*		MEM_1_MEM_DQ[7]		
	82	MEM_DQ[38]*		MEM_1_MEM_DQ[6]		
	81	MEM_DQ[37]*		MEM_1_MEM_DQ[5]		
	80	MEM_DQ[36]*		MEM_1_MEM_DQ[4]		
	79					
	78	MEM_DM_N[4]		MEM_1_MEM_DM_N[0]		
	77	MEM_DQS_C[4]		MEM_1_MEM_DQS_C[0]		
	76	MEM_DQS_T[4]		MEM_1_MEM_DQS_T[0]		
	75	MEM_DQ[35]*		MEM_1_MEM_DQ[3]		
	74	MEM_DQ[34]*		MEM_1_MEM_DQ[2]		
	73	MEM_DQ[33]*		MEM_1_MEM_DQ[1]		
	72	MEM_DQ[32]*		MEM_1_MEM_DQ[0]		
BL5	71	MEM_DQ[31]	MEM_DQ[31]	MEM_1_CK_C[1]		
	70	MEM_DQ[30]	MEM_DQ[30]	MEM_1_CK_T[1]		
	69	MEM_DQ[29]	MEM_DQ[29]	MEM_1_MEM_CS_N[0]		
continued...						

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	68	MEM_DQ[28]	MEM_DQ[28]	MEM_1_MEM_CS_N[1]		
	67			MEM_1_CK_C[0]		
	66	MEM_DM_N[3]	MEM_DM_N[3]	MEM_1_CK_T[0]		
	65	MEM_DQS_C[3]	MEM_DQS_C[3]	MEM_1_MEM_CA[12]		
	64	MEM_DQS_T[3]	MEM_DQS_T[3]	MEM_1_MEM_CA[11]		
	63	MEM_DQ[27]	MEM_DQ[27]	MEM_1_RESET_N		
	62	MEM_DQ[26]	MEM_DQ[26]	OCT_1_OCT_RZQIN		
	61	MEM_DQ[25]	MEM_DQ[25]	MEM_1_ALERT_N		
	60	MEM_DQ[24]	MEM_DQ[24]	MEM_1_MEM_CA[10]		
BL4	59	MEM_DQ[23]	MEM_DQ[23]	Differential "NSide" Reference Clock Input Site	MEM_DQ[23]*	
	58	MEM_DQ[22]	MEM_DQ[22]	Differential "PSide" Reference Clock Input Site	MEM_DQ[22]*	
	57	MEM_DQ[21]	MEM_DQ[21]	MEM_1_MEM_CA[9]	MEM_DQ[21]*	
	56	MEM_DQ[20]	MEM_DQ[20]	MEM_1_MEM_CA[8]	MEM_DQ[20]*	
	55			MEM_1_MEM_CA[7]		
	54	MEM_DM_N[2]	MEM_DM_N[2]	MEM_1_MEM_CA[6]	MEM_DM_N[2]	
	53	MEM_DQS_C[2]	MEM_DQS_C[2]	MEM_1_MEM_CA[5]	MEM_DQS_C[2]	
	52	MEM_DQS_T[2]	MEM_DQS_T[2]	MEM_1_MEM_CA[4]	MEM_DQS_T[2]	
	51	MEM_DQ[19]	MEM_DQ[19]	MEM_1_MEM_CA[3]	MEM_DQ[19]*	
	50	MEM_DQ[18]	MEM_DQ[18]	MEM_1_MEM_CA[2]	MEM_DQ[18]*	
	49	MEM_DQ[17]	MEM_DQ[17]	MEM_1_MEM_CA[1]	MEM_DQ[17]*	
	48	MEM_DQ[16]	MEM_DQ[16]	MEM_1_MEM_CA[0]	MEM_DQ[16]*	
continued...						

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
BL3	47	MEM_CK_C[1]	MEM_CK_C[1]	MEM_0_CK_C[1]	MEM_CK_C[1]	MEM_CK_C[1]
	46	MEM_CK_T[1]	MEM_CK_T[1]	MEM_0_CK_T[1]	MEM_CK_T[1]	MEM_CK_T[1]
	45	MEM_CS_N[0]	MEM_CS_N[0]	MEM_0_MEM_CS_N[0]	MEM_CS_N[0]	MEM_CS_N[0]
	44	MEM_CS_N[1]	MEM_CS_N[1]	MEM_0_MEM_CS_N[1]	MEM_CS_N[1]	MEM_CS_N[1]
	43	MEM_CK_C[0]	MEM_CK_C[0]	MEM_0_CK_C[0]	MEM_CK_C[0]	MEM_CK_C[0]
	42	MEM_CK_T[0]	MEM_CK_T[0]	MEM_0_CK_T[0]	MEM_CK_T[0]	MEM_CK_T[0]
	41	MEM_CA[12]	MEM_CA[12]	MEM_0_MEM_CA[12]	MEM_CA[12]	MEM_CA[12]
	40	MEM_CA[11]	MEM_CA[11]	MEM_0_MEM_CA[11]	MEM_CA[11]	MEM_CA[11]
	39	MEM_RESET_N[0]	MEM_RESET_N[0]	MEM_0_RESET_N	MEM_RESET_N[0]	MEM_RESET_N[0]
	38	RZQ Site	RZQ Site	OCT_0_OCT_RZQIN	RZQ Site	RZQ Site
	37	MEM_ALERT_N[0]	MEM_ALERT_N[0]	MEM_0_ALERT_N	MEM_ALERT_N[0]	MEM_ALERT_N[0]
	36	MEM_CA[10]	MEM_CA[10]	MEM_0_MEM_CA[10]	MEM_CA[10]	MEM_CA[10]
BL2	35	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site	Differential "N-Side" Reference Clock Input Site
	34	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	Differential "P-Side" Reference Clock Input Site
	33	MEM_CA[9]	MEM_CA[9]	MEM_0_MEM_CA[9]	MEM_CA[9]	MEM_CA[9]
	32	MEM_CA[8]	MEM_CA[8]	MEM_0_MEM_CA[8]	MEM_CA[8]	MEM_CA[8]
	31	MEM_CA[7]	MEM_CA[7]	MEM_0_MEM_CA[7]	MEM_CA[7]	MEM_CA[7]
	30	MEM_CA[6]	MEM_CA[6]	MEM_0_MEM_CA[6]	MEM_CA[6]	MEM_CA[6]
	29	MEM_CA[5]	MEM_CA[5]	MEM_0_MEM_CA[5]	MEM_CA[5]	MEM_CA[5]
	28	MEM_CA[4]	MEM_CA[4]	MEM_0_MEM_CA[4]	MEM_CA[4]	MEM_CA[4]
	27	MEM_CA[3]	MEM_CA[3]	MEM_0_MEM_CA[3]	MEM_CA[3]	MEM_CA[3]
continued...						

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	26	MEM_CA[2]	MEM_CA[2]	MEM_0_MEM_CA[2]	MEM_CA[2]	MEM_CA[2]
	25	MEM_CA[1]	MEM_CA[1]	MEM_0_MEM_CA[1]	MEM_CA[1]	MEM_CA[1]
	24	MEM_CA[0]	MEM_CA[0]	MEM_0_MEM_CA[0]	MEM_CA[0]	MEM_CA[0]
BL1	23	MEM_DQ[7]	MEM_DQ[7]	MEM_0_MEM_DQ[7]	MEM_DQ[7]	MEM_DQ[7]
	22	MEM_DQ[6]	MEM_DQ[6]	MEM_0_MEM_DQ[6]	MEM_DQ[6]	MEM_DQ[6]
	21	MEM_DQ[5]	MEM_DQ[5]	MEM_0_MEM_DQ[5]	MEM_DQ[5]	MEM_DQ[5]
	20	MEM_DQ[4]	MEM_DQ[4]	MEM_0_MEM_DQ[4]	MEM_DQ[4]	MEM_DQ[4]
	19					
	18	MEM_DM_N[0]	MEM_DM_N[0]	MEM_0_MEM_DM_N[0]	MEM_DM_N[0]	MEM_DM_N[0]
	17	MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_0_MEM_DQS_C[0]	MEM_DQS_C[0]	MEM_DQS_C[0]
	16	MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_0_MEM_DQS_T[0]	MEM_DQS_T[0]	MEM_DQS_T[0]
	15	MEM_DQ[3]	MEM_DQ[3]	MEM_0_MEM_DQ[3]	MEM_DQ[3]	MEM_DQ[3]
	14	MEM_DQ[2]	MEM_DQ[2]	MEM_0_MEM_DQ[2]	MEM_DQ[2]	MEM_DQ[2]
	13	MEM_DQ[1]	MEM_DQ[1]	MEM_0_MEM_DQ[1]	MEM_DQ[1]	MEM_DQ[1]
	12	MEM_DQ[0]	MEM_DQ[0]	MEM_0_MEM_DQ[0]	MEM_DQ[0]	MEM_DQ[0]
BLO	11	MEM_DQ[15]	MEM_DQ[15]	MEM_0_MEM_DQ[15]	MEM_DQ[15]	MEM_DQ[15]
	10	MEM_DQ[14]	MEM_DQ[14]	MEM_0_MEM_DQ[14]	MEM_DQ[14]	MEM_DQ[14]
	9	MEM_DQ[13]	MEM_DQ[13]	MEM_0_MEM_DQ[13]	MEM_DQ[13]	MEM_DQ[13]
	8	MEM_DQ[12]	MEM_DQ[12]	MEM_0_MEM_DQ[12]	MEM_DQ[12]	MEM_DQ[12]
	7					
	6	MEM_DM_N[1]	MEM_DM_N[1]	MEM_0_MEM_DM_N[1]	MEM_DM_N[1]	MEM_DM_N[1]
	5	MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_0_MEM_DQS_C[1]	MEM_DQS_C[1]	MEM_DQS_C[1]
	4	MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_0_MEM_DQS_T[1]	MEM_DQS_T[1]	MEM_DQS_T[1]
	3	MEM_DQ[11]	MEM_DQ[11]	MEM_0_MEM_DQ[11]	MEM_DQ[11]	MEM_DQ[11]
continued...						

Lane Number	Pin Index	x32+ECC *	x 32	2ch x16	x16 + ECC *	x16
	2	MEM_DQ[10]	MEM_DQ[10]	MEM_0_MEM_DQ[10]	MEM_DQ[10]	MEM_DQ[10]
	1	MEM_DQ[9]	MEM_DQ[9]	MEM_0_MEM_DQ[9]	MEM_DQ[9]	MEM_DQ[9]
	0	MEM_DQ[8]	MEM_DQ[8]	MEM_0_MEM_DQ[8]	MEM_DQ[8]	MEM_DQ[8]

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

Table 15. 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]
	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		
continued...			

Lane Number	Pin Index	x32	2 Channel x16
	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]
	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	
continued...			

Lane Number	Pin Index	x32	2 Channel x16
	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]
	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 16. 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]
	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		
continued...			

Lane Number	Pin Index	x32	2 Channel x16
BL4	60		
	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
	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]
continued...			

Lane Number	Pin Index	x32	2 Channel x16
	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]
	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.

3.2.3.3. 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 17. 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 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 x16 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 18. Lanes Used Per DQS Group

Group Size	Number of Lanes Used	Maximum Number of Data Pins per Group
x8	1	10
x16	2	22
pair of x4	1	4 per group, 8 per lane

Figure 5. x4 Group

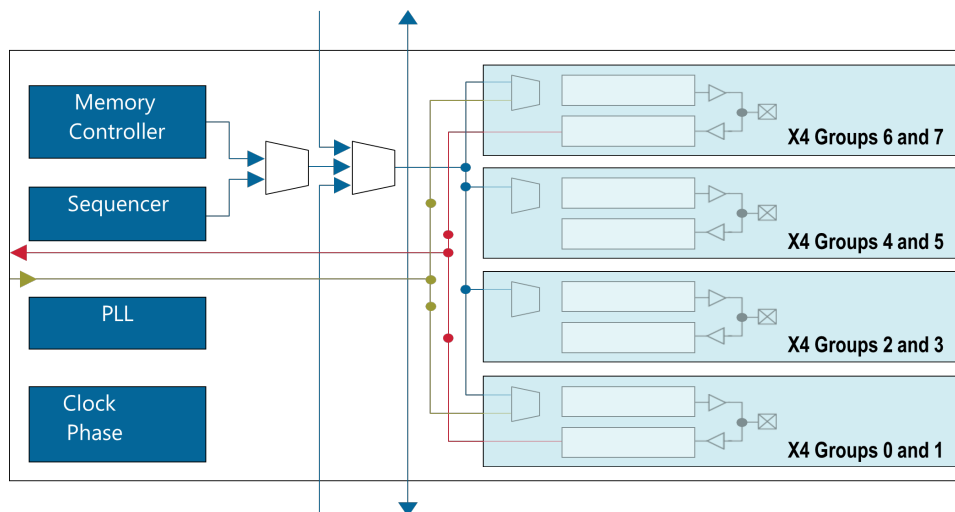


Figure 6. x8 Group

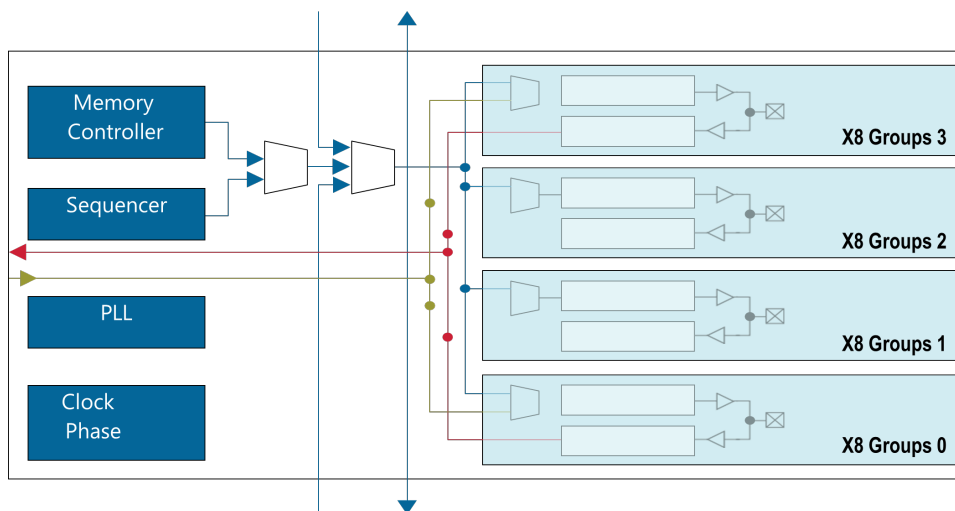
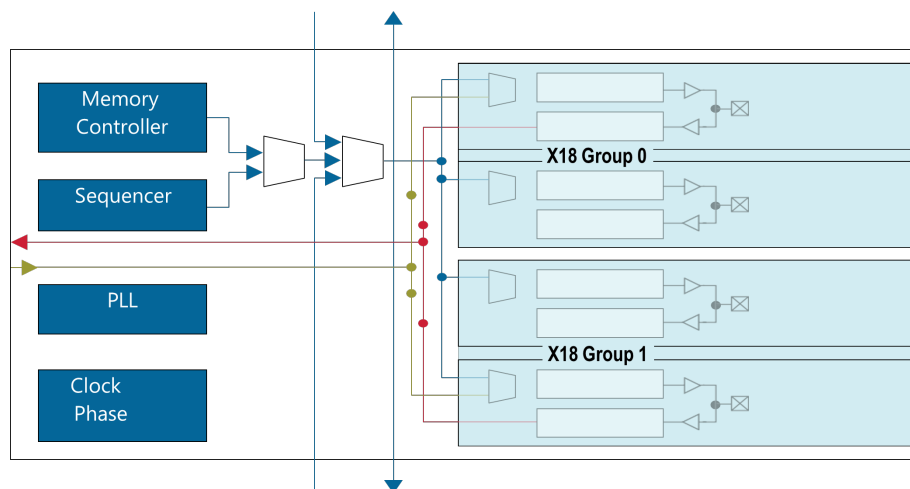


Figure 7. x16 Group



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 mode, or x16 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 19. 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	0		4	5
x8	1		16	17
x8	2		28	29
x8	3		40	41
x16	0, 1		4	5
x16	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	0		52	53
x8	1		64	65
x8	2		76	77
x8	3		88	89
x16	0,1		52	53
x16	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.

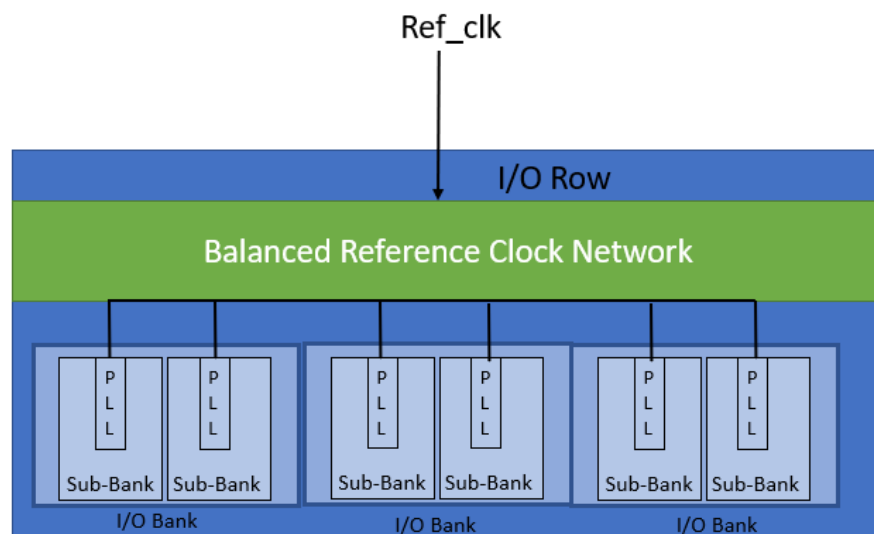
The PLL reference clock must be constrained to the address and command sub-bank only.

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 8. 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 9. Clock Phase Alignment Illustration

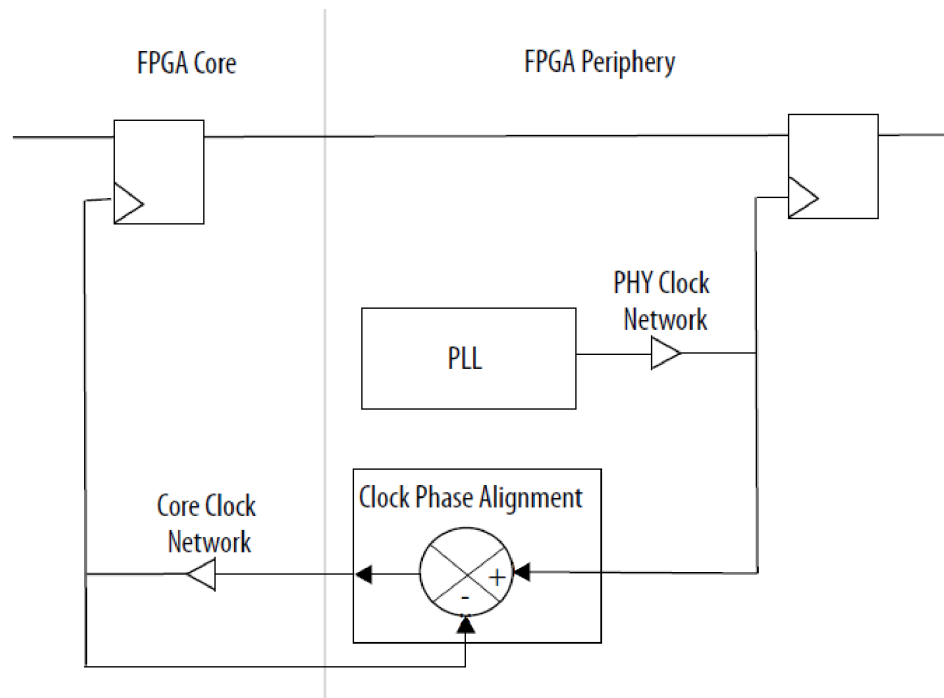
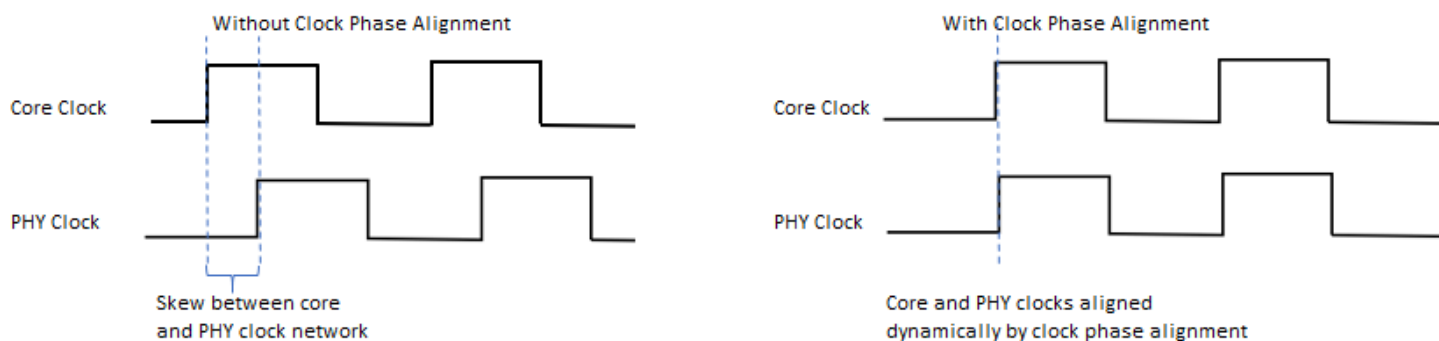


Figure 10. Effect of Clock Phase Alignment



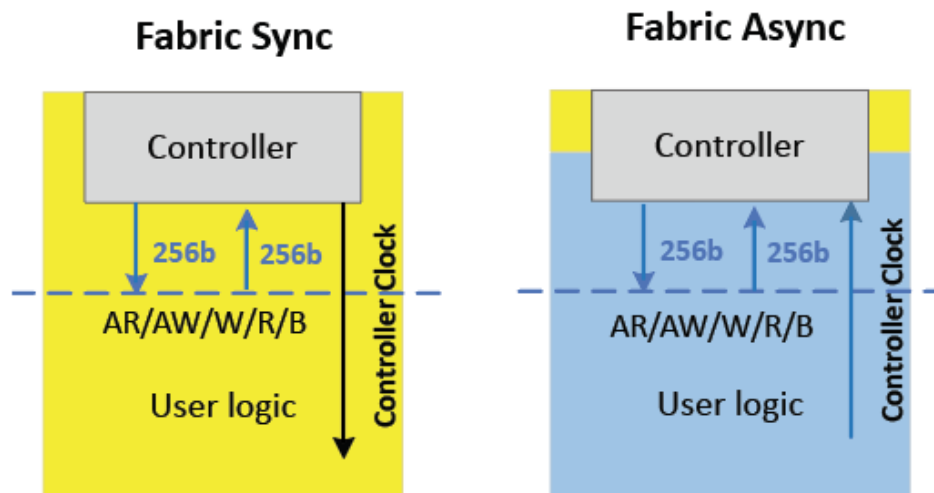
3.2.9. User Clock in Different Core Access Modes

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 $((\text{mem_CK})/4)$.
 - In DDR5, LPDDR5, and LPDDR4, the user clock frequency will be one-eighth of the memory clock frequency $((\text{mem CK})/8)$.
- 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.
 - It is recommended to set the user clock frequency to one-quarter of the memory clock.

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

Figure 11. Access Modes



Benefits of Each Access Mode

- Synchronous fabric clocking is required for DDR4 DIMM.
- Asynchronous fabric access mode has the lowest latency.
- Asynchronous fabric access mode can achieve higher memory clock frequency in some speed grade / protocol combinations.
- Asynchronous fabric access mode can achieve higher efficiency on secondary memory controller in the following configurations:
 - 2ch x16 LPDDR4 / LPDDR5 / DDR5
 - 4ch x16 LPDDR4 / LPDDR5
 - 1ch x16 of LPDDR4 / LPDDR5 / DDR5 on the top sub-bank

Altera recommends using asynchronous clocking mode when you use the configuration listed above. Refer to guidelines in *Optimizing Efficiency for Secondary Controller* when using asynchronous clocking mode, for achieving optimal efficiency with a secondary controller.

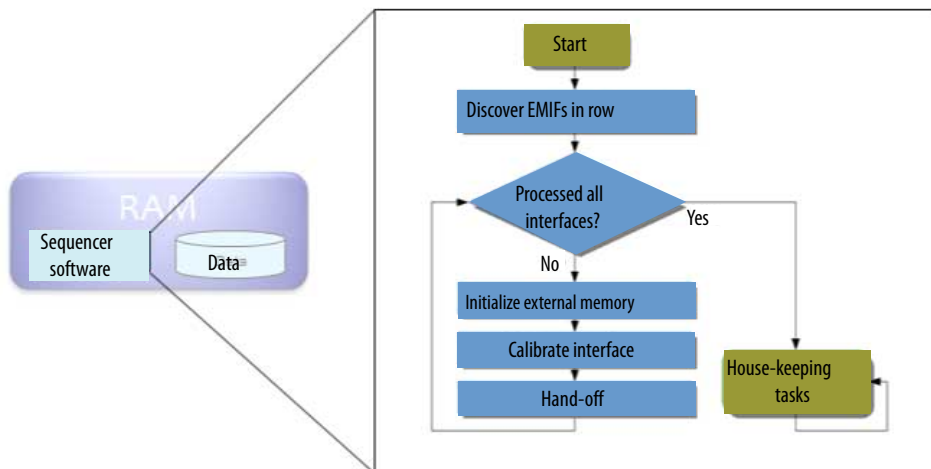
3.3. Agilex 5 EMIF Sequencer

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 12. Agilex 5 EMIF Sequencer Operation



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	<ul style="list-style-type: none">• Supports the AXI4 interface.
Burst devices support	Supports the following memory devices: <ul style="list-style-type: none">• Discrete (DDR4,LPDDR4,DDR5,LPDDR5)• UDIMM (DDR4, DDR5)• SODIMM (DDR4,DDR5)• RDIMM (DDR4,DDR5)
Burst length support	<ul style="list-style-type: none">• DDR4: BL8• DDR5: BL16• LPDDR4: BL16• LPDDR5: BL16
continued...	

Feature	Description
Efficiency optimization features	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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.

HPS EMIF Mapping (Both bridges are used)

When using both the F2H bridge and the F2SDRAM bridge, no I/O sharing is allowed. That is, the HPS can access DDR and fabric can access DDR via the F2H and/or F2SDRAM bridge, but no GPIO, etc is allowed.

Table 22. HPS EMIF Mapping (Both bridges are used)

Protocol	Banks	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR4	1	1x16	X	X	X	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16+E CC	X	X	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32	X	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32+E CC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16	DQ[0]	AC0	AC2	AC1	DQ[1]	X	X	X
	1	1x16+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[ECC]	X	X
	1 or 2	1x32	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	X
	1 or 2	1x32+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	DQ[ECC]
	N/A	1x64	Not supported							
	N/A	1x64+E CC	Not supported							
DDR5	1	1x16	X	X	X	X	AC1	AC0	DQ[0]	DQ[1]
	1	1x16+E CC	X	X	X	DQ[ECC]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32	X	X	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32+E CC	X	DQ[ECC]	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1	1x16	DQ[1]	DQ[0]	AC1	AC0	X	X	X	X
	1	1x16+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[ECC]	X	X	X
	1 or 2	1x32	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	X	X
	1 or 2	1x32+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	DQ[ECC]	X
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[0]	DQ[1]
LPDDR4 / LPDDR5	1	1x16	X	X	X	X	AC1	AC0	DQ[1]	DQ[0]
	1 or 2	1x32	DQ[3]	DQ[2]	X	X	AC1	AC0	DQ[1]	DQ[0]
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	2	4x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]

- GM - Available for GPIO, MIPI, LVDS SERDES, or PHYLite..
- GL - Available for GPIO or LVDS SERDES (no RZQ free). Can support MIPI through RZQ sharing.
- GO - Available for GPIO only (no refclk lane free). Can support MIPI through RZQ and refclk sharing.
- RZ - RZQ and refclk lane when using PHYLite/MIPI.
- X - Not Available.

HPS EMIF Mapping (Using only the F2SDRAM bridge)

The following table shows the I/O sharing that is allowed when using the F2SDRAM bridge and not using the F2H bridge. The HPS can access DDR and fabric can access DDR via the F2SDRAM bridge only.

Table 23. HPS EMIF Mapping (Using only the F2SDRAM bridge)

Protocol	Banks	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR4	1	1x16	GM	GM	RZ	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16+E CC	GO	GO	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32	GO	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32+E CC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16	DQ[0]	AC0	AC2	AC1	DQ[1]	X	X	X
	1	1x16+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[ECC]	X	X
	1 or 2	1x32	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	X
	1 or 2	1x32+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	DQ[ECC]
	N/A	1x64	Not supported							
	N/A	1x64+E CC	Not supported							
DDR5	1	1x16	GM	GM	RZ	GM	AC1	AC0	DQ[0]	DQ[1]
	1	1x16+E CC	GM	GM	RZ	DQ[ECC]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32	GO	GO	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32+E CC	GO	DQ[ECC]	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1	1x16	DQ[1]	DQ[0]	AC1	AC0	X	X	X	X
	1	1x16+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[ECC]	X	X	X
	1 or 2	1x32	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	X	X
	1 or 2	1x32+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	DQ[ECC]	X
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[0]	DQ[1]
LPDDR4 / LPDDR5	1	1x16	GM	GM	RZ	GM	AC1	AC0	DQ[1]	DQ[0]
	1 or 2	1x32	DQ[3]	DQ[2]	RZ	GM	AC1	AC0	DQ[1]	DQ[0]
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	2	4x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]

- GM - Available for GPIO, MIPI, LVDS SERDES, or PHYLite..
- GL - Available for GPIO or LVDS SERDES (no RZQ free). Can support MIPI through RZQ sharing.
- GO - Available for GPIO only (no refclk lane free). Can support MIPI through RZQ and refclk sharing.
- RZ - RZQ and refclk lane when using PHYLite/MIPI.
- X - Not Available.

HPS EMIF Mapping (Using only the F2H bridge)

The following table shows the I/O sharing permitted when using the F2H bridge and not using the F2SDRAM bridge. The HPS can access DDR and fabric can access DDR via the F2H bridge only.

Table 24. HPS EMIF Mapping (Using only the F2H bridge)

Protocol	Banks	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR4	1	1x16	X	X	X	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16+E CC	X	X	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32	X	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32+E CC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16	DQ[0]	AC0	AC2	AC1	DQ[1]	GL	GL	GL
	1	1x16+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[ECC]	GO	GO
	1 or 2	1x32	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	GO
	1 or 2	1x32+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	DQ[ECC]
	N/A	1x64	Not supported							
	N/A	1x64+E CC	Not supported							
DDR5	1	1x16	X	X	X	X	AC1	AC0	DQ[0]	DQ[1]
	1	1x16+E CC	X	X	X	DQ[ECC]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32	X	X	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32+E CC	X	DQ[ECC]	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1	1x16	DQ[1]	DQ[0]	AC1	AC0	RZ	GM	GM	GM
	1	1x16+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[ECC]	GL	GL	GL
	1 or 2	1x32	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	GO	GO
	1 or 2	1x32+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	DQ[ECC]	GO
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[0]	DQ[1]
LPDDR4 / LPDDR5	1	1x16	X	X	X	X	AC1	AC0	DQ[1]	DQ[0]
	1 or 2	1x32	DQ[3]	DQ[2]	X	X	AC1	AC0	DQ[1]	DQ[0]
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	2	4x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]

- GM - Available for GPIO, MIPI, LVDS SERDES, or PHYLite..
- GL - Available for GPIO or LVDS SERDES (no RZQ free). Can support MIPI through RZQ sharing.
- GO - Available for GPIO only (no refclk lane free). Can support MIPI through RZQ and refclk sharing.
- RZ - RZQ and refclk lane when using PHYLite/MIPI.
- X - Not Available.

HPS EMIF Mapping (No bridges are used)

The following table shows the I/O sharing permitted when using neither the F2H bridge nor the FS2DRAM bridge. The HPS can access DDR, but the fabric cannot.

Table 25. HPS EMIF Mapping (No bridges are used)

Protocol	Banks	Data Width	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR4	1	1x16	GM	GM	RZ	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16+E CC	GO	GO	DQ[ECC]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32	GO	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1 or 2	1x32+E CC	DQ[ECC]	DQ[3]	DQ[2]	DQ[1]	AC2	AC1	AC0	DQ[0]
	1	1x16	DQ[0]	AC0	AC2	AC1	DQ[1]	GL	GL	GL
	1	1x16+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[ECC]	GO	GO
	1 or 2	1x32	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	GO
	1 or 2	1x32+E CC	DQ[0]	AC0	AC2	AC1	DQ[1]	DQ[2]	DQ[3]	DQ[ECC]
	N/A	1x64	Not supported							
	N/A	1x64+E CC	Not supported							
DDR5	1	1x16	GM	GM	RZ	GM	AC1	AC0	DQ[0]	DQ[1]
	1	1x16+E CC	GM	GM	RZ	DQ[ECC]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32	GO	GO	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1 or 2	1x32+E CC	GO	DQ[ECC]	DQ[3]	DQ[2]	AC1	AC0	DQ[0]	DQ[1]
	1	1x16	DQ[1]	DQ[0]	AC1	AC0	RZ	GM	GM	GM
	1	1x16+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[ECC]	GL	GL	GL
	1 or 2	1x32	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	GO	GO
	1 or 2	1x32+E CC	DQ[1]	DQ[0]	AC1	AC0	DQ[2]	DQ[3]	DQ[ECC]	GO
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[0]	DQ[1]
LPDDR4 / LPDDR5	1	1x16	GM	GM	RZ	GM	AC1	AC0	DQ[1]	DQ[0]
	1 or 2	1x32	DQ[3]	DQ[2]	RZ	GM	AC1	AC0	DQ[1]	DQ[0]
	1	2x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]
	2	4x16	DQ[1]	DQ[0]	AC1	AC0	AC1	AC0	DQ[1]	DQ[0]

- GM - Available for GPIO, MIPI, LVDS SERDES, or PHYLite..
- GL - Available for GPIO or LVDS SERDES (no RZQ free). Can support MIPI through RZQ sharing.
- GO - Available for GPIO only (no refclk lane free). Can support MIPI through RZQ and refclk sharing.
- RZ - RZQ and refclk lane when using PHYLite/MIPI.
- X - Not Available.

I/O Sharing versus HPS EMIF DDR Protocol versus Bridge Allowed

The following table summarizes the information from the previous tables, showing which bridge can be used with which I/O sharing and DDR protocol combinations.

Table 26. I/O Sharing versus HPS EMIF DDR Protocol versus Bridge Allowed

I/O Sharing	Protocol	Banks	Data Width	Bridge Allowed
MIPI or PHYLite	DDR4	1	1x16	F2SDRAM
		1	1x16	F2SDRAM or F2H
	LPDDR4 / LPDDR5	1	1x16+ECC	F2SDRAM
		1	1x16	F2SDRAM
		1 or 2	1x32	F2SDRAM
MIPI through RZQ sharing	DDR4	1	1x16	F2H
	DDR5	1	1x16+ECC	F2H
MIPI through RZQ and refclk sharing	DDR4	1	1x16+ECC	F2SDRAM or F2H
		1 or 2	1x32	F2SDRAM or F2H
	DDR5	1 or 2	1x32	F2SDRAM or F2H
		1 or 2	1x32+ECC	F2SDRAM or F2H
LVDS	DDR4	1	1x16	F2SDRAM or F2H
	DDR5	1	1x16	F2SDRAM or F2H
		1	1x16+ECC	F2SDRAM or F2H
	LPDDR4 / LPDDR5	1	1x16	F2SDRAM
		1 or 2	1x32	F2SDRAM
GPIO	DDR4	1	1x16	F2SDRAM or F2H
		1	1x16+ECC	F2SDRAM or F2H
		1 or 2	1x32	F2SDRAM or F2H
	DDR5	1	1x16	F2SDRAM or F2H
		1	1x16+ECC	F2SDRAM or F2H
		1 or 2	1x32	F2SDRAM or F2H
		1 or 2	1x32+ECC	F2SDRAM or F2H
	LPDDR4 / LPDDR5	1	1x16	F2SDRAM
		1 or 2	1x32	F2SDRAM

3.5.1. Restrictions on I/O Bank Usage for 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.
- Pins that are not used by the HPS-EMIF directly are available for I/O sharing with other protocols, such as GPIO, MIPI, LVDS SERDES, or PHYLite, with certain HPS bridge restrictions which are described in the following tables.
- HPS-EMIF and AVSTx16 configuration mode cannot be used simultaneously, because both use bank 3A.
- Reference clock sharing is allowed between HPS-EMIF IP and other IPs in certain cases.
- For multi-channel EMIFs or when multiple EMIFs are used inside HPS-EMIF IP, they must have identical IP parameters.

3.5.2. Using the Legacy EMIF Debug Toolkit with Agilex 5 HPS Interfaces

The Legacy External Memory Interface Debug Toolkit is not directly compatible with HPS interfaces.

To debug your HPS interface using the Legacy EMIF Debug Toolkit, you should create an identically parameterized, non-HPS version of your interface, and apply the toolkit to that interface. When you finish debugging this non-HPS interface, you can then apply any needed changes to your HPS interface, and continue your design development.

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. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

The interfaces in the Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 27. Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Interface Name	Interface Type	Description
s0_axi4_clock_in	clock	Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.
s0_axi4_clock_out	clock	Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.
s0_axi4_ctrl_ready	reset	Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.
core_init_n	reset	An input to indicate that core configuration is complete.
s0_axi4	axi4	Mainband AXI4 from fabric to controller, channel 0.
s0_axi4lite_clock	clock	Clock for sideband interface (primary I/O bank).
s0_axi4lite_reset_n	reset	Reset for sideband interface (primary I/O bank).
s0_axi4lite	axi4lite	Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.
mem_0	conduit	Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.
mem_ck_0	conduit	Clock pin to the memory (channel 0).
mem_reset_n	conduit	Reset pin to the memory. Must always be placed along with channel 0, but shared for entire interface (all channels within one EMIF).
oct_0	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).
ref_clk	clock	Reference clock used by the EMIF PLL.

4.1.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.

Table 28. Interface: s0_axi4_clock_in

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_in	Input	User clock for mainband axi. Input clock to the EMIF IP, no relationship to PHY clock.

4.1.2. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.

Table 29. Interface: s0_axi4_clock_out

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_out	Output	User clock for maiband axi (primary I/O bank). Output clock from the EMIF IP (output from CPA block, synchronous to PHY clock).

4.1.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.

Table 30. Interface: s0_axi4_ctrl_ready

Interface type: reset

Port Name	Direction	Description
s0_axi4_reset_n	Output	Output signal from EMIF IP (primary I/O bank), indicating that Calibration of the channels in this I/O bank is complete, and controllers in this I/O bank are ready for use.

4.1.4. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

An input to indicate that core configuration is complete.

Table 31. 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. When high, indicates core initialization is complete.

4.1.5. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Mainband AXI4 from fabric to controller, channel 0.

Table 32. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
s0_axi4_awaddr	Input	Write Address , channel 0.
s0_axi4_awburst	Input	Write Burst Type, channel 0.
s0_axi4_awid	Input	Write Address ID, channel 0.
s0_axi4_awlen	Input	Write Burst Length, channel 0.
s0_axi4_awlock	Input	Write Lock Type, channel 0.
s0_axi4_awqos	Input	Write Quality of Service, channel 0.
s0_axi4_awsz	Input	Write Burst Size, channel 0.
s0_axi4_awvalid	Input	Write Address Valid, channel 0.
s0_axi4_awuser	Input	Write Address User Signal, channel 0.
s0_axi4_awprot	Input	Write Protection Type, channel 0.
s0_axi4_awready	Output	Write Address Ready, channel 0.
s0_axi4_araddr	Input	Read Address , channel 0.
s0_axi4_arburst	Input	Read Burst Type, channel 0.
s0_axi4_arid	Input	Read Address ID, channel 0.
s0_axi4_arlen	Input	Read Burst Length, channel 0.
s0_axi4_arlock	Input	Read Lock Type, channel 0.
s0_axi4_arqos	Input	Read Quality of Service, channel 0.
s0_axi4_arsz	Input	Read Burst Size, channel 0.
s0_axi4_arvalid	Input	Read Address Valid, channel 0.
s0_axi4_aruser	Input	Read Address User Signal, channel 0.
s0_axi4_arprot	Input	Read Protection Type, channel 0.
s0_axi4_arready	Output	Read Address Ready, channel 0.
s0_axi4_wdata	Input	Write Data , channel 0.
s0_axi4_wstrb	Input	Write Strobes, channel 0.
s0_axi4_wlast	Input	Write Last, channel 0.
continued...		

Port Name	Direction	Description
s0_axi4_wvalid	Input	Write Valid, channel 0.
s0_axi4_wready	Output	Write Ready, channel 0.
s0_axi4_bready	Input	Write Response Ready, channel 0.
s0_axi4_bid	Output	Write Response ID, channel 0.
s0_axi4_bresp	Output	Write Response , channel 0.
s0_axi4_bvalid	Output	Write Response Valid, channel 0.
s0_axi4_rready	Input	Read Ready, channel 0.
s0_axi4_rdata	Output	Read Data, channel 0.
s0_axi4_rid	Output	Read ID , channel 0.
s0_axi4_rlast	Output	Read Last, channel 0.
s0_axi4_rresp	Output	Read Response, channel 0.
s0_axi4_rvalid	Output	Read Valid, channel 0.

4.1.6. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Clock for sideband interface (primary I/O bank).

Table 33. Interface: s0_axi4lite_clock

Interface type: clock

Port Name	Direction	Description
s0_axi4lite_clock	Input	Axi-Lite clock, to primary IOSSM.

4.1.7. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Reset for sideband interface (primary I/O bank).

Table 34. Interface: s0_axi4lite_reset_n

Interface type: reset

Port Name	Direction	Description
s0_axi4lite_reset_n	Input	Axi-Lite reset_n, to primary IOSSM.

4.1.8. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.

Table 35. Interface: s0_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s0_axi4lite_awaddr	Input	Axi-Lite Write Address, to primary IOSSM.
s0_axi4lite_awprot	Input	Axi-Lite Write Address Protection Signal, to primary IOSSM.
s0_axi4lite_awvalid	Input	Axi-Lite Write Address Valid, to primary IOSSM.
s0_axi4lite_awready	Output	Axi-Lite Write Address Ready, to primary IOSSM.
s0_axi4lite_araddr	Input	Axi-Lite Read Address, to primary IOSSM.
s0_axi4lite_arprot	Input	Axi-Lite Read Address Protection Signal, to primary IOSSM.
s0_axi4lite_arvalid	Input	Axi-Lite Read Address Valid, to primary IOSSM.
s0_axi4lite_arready	Output	Axi-Lite Read Address Ready, to primary IOSSM.
s0_axi4lite_wdata	Input	Axi-Lite Write Data, to primary IOSSM.
s0_axi4lite_wstrb	Input	Axi-Lite Write Strobe, to primary IOSSM.
s0_axi4lite_wvalid	Input	Axi-Lite Write Valid, to primary IOSSM.
s0_axi4lite_wready	Output	Axi-Lite Write Ready, to primary IOSSM.
s0_axi4lite_bready	Input	Axi-Lite Write Response Ready, to primary IOSSM.
s0_axi4lite_bresp	Output	Axi-Lite Write Response, to primary IOSSM.
s0_axi4lite_bvalid	Output	Axi-Lite Write Response Valid, to primary IOSSM.
s0_axi4lite_rready	Input	Axi-Lite Read Ready, to primary IOSSM.
s0_axi4lite_rdata	Output	Axi-Lite Read Data, to primary IOSSM.
s0_axi4lite_rresp	Output	Axi-Lite Read Response, to primary IOSSM.
s0_axi4lite_rvalid	Output	Axi-Lite Read Valid, to primary IOSSM.

4.1.9. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.

Table 36. Interface: mem_0

Interface type: conduit

Port Name	Direction	Description
mem_0_cke	Output	Clock Enable channel 0.
mem_0_odt	Output	On-Die Termination channel 0.
mem_0_cs_n	Output	Chip Select channel 0.
mem_0_a	Output	Address channel 0.
mem_0_ba	Output	Bank Address channel 0.
mem_0_bg	Output	Bank Group channel 0.
mem_0_act_n	Output	Activation Command channel 0.
mem_0_par	Output	Command/Address Parity channel 0.
<i>continued...</i>		

Port Name	Direction	Description
mem_0_dq	Bidir	Data (read/write) channel 0.
mem_0_dqs_t	Bidir	Data Strobe (true) channel 0.
mem_0_dqs_c	Bidir	Data Strobe (complement) channel 0.
mem_0_alert_n	Input	Indicates Write CRC Error channel 0.

4.1.10. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Clock pin to the memory (channel 0).

Table 37. Interface: mem_ck_0

Interface type: conduit

Port Name	Direction	Description
mem_0_ck_t	Output	CK Clock (true) channel 0.
mem_0_ck_c	Output	CK Clock (complement) channel 0.

4.1.11. mem_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Reset pin to the memory. Must always be placed along with channel 0, but shared for entire interface (all channels within one EMIF).

Table 38. Interface: mem_reset_n

Interface type: conduit

Port Name	Direction	Description
mem_0_reset_n	Output	Asynchronous Reset channel 0.

4.1.12. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).

Table 39. Interface: oct_0

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_0	Input	Calibrated On-Chip Termination (OCT) input pin channel 0.

4.1.13. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR4 Component

Reference clock used by the EMIF PLL.

Table 40. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	Input	PLL reference clock input.

4.2. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

The interfaces in the Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 41. Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Interface Name	Interface Type	Description
s0_axi4_clock_in	clock	Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.
s0_axi4_clock_out	clock	Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.
s0_axi4_ctrl_ready	reset	Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.
core_init_n	reset	An input to indicate that core configuration is complete.
s0_axi4	axi4	Mainband AXI4 from fabric to controller, channel 0.
s1_axi4	axi4	Mainband AXI4 from fabric to controller, channel 1.
s0_axi4lite_clock	clock	Clock for sideband interface (primary I/O bank).
s0_axi4lite_reset_n	reset	Reset for sideband interface (primary I/O bank).
s0_axi4lite	axi4lite	Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.
mem_0	conduit	Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.
mem_ck_0	conduit	Clock pin to the memory (channel 0).
mem_reset_n_0	conduit	Reset pin to the memory (channel 0).
mem_1	conduit	Interface to the memory (channel 1), including all CA pins, DQ pins, and DQS pins.
mem_ck_1	conduit	Clock pin to the memory (channel 1).
mem_reset_n_1	conduit	Reset pin to the memory (channel 1).
oct_0	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).
oct_1	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 1).
ref_clk	clock	Reference clock used by the EMIF PLL.

4.2.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.

Table 42. Interface: s0_axi4_clock_in

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_in	Input	User clock for mainband axi. Input clock to the EMIF IP, no relationship to PHY clock.

4.2.2. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.

Table 43. Interface: s0_axi4_clock_out

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_out	Output	User clock for maiband axi (primary I/O bank). Output clock from the EMIF IP (output from CPA block, synchronous to PHY clock).

4.2.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.

Table 44. Interface: s0_axi4_ctrl_ready

Interface type: reset

Port Name	Direction	Description
s0_axi4_reset_n	Output	Output signal from EMIF IP (primary I/O bank), indicating that Calibration of the channels in this I/O bank is complete, and controllers in this I/O bank are ready for use.

4.2.4. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

An input to indicate that core configuration is complete.

Table 45. 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. When high, indicates core initialization is complete.

4.2.5. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Mainband AXI4 from fabric to controller, channel 0.

Table 46. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
s0_axi4_awaddr	Input	Write Address , channel 0.
s0_axi4_awburst	Input	Write Burst Type, channel 0.
s0_axi4_awid	Input	Write Address ID, channel 0.
s0_axi4_awlen	Input	Write Burst Length, channel 0.
s0_axi4_awlock	Input	Write Lock Type, channel 0.
s0_axi4_awqos	Input	Write Quality of Service, channel 0.
s0_axi4_awsz	Input	Write Burst Size, channel 0.
s0_axi4_awvalid	Input	Write Address Valid, channel 0.
s0_axi4_awuser	Input	Write Address User Signal, channel 0.
s0_axi4_awprot	Input	Write Protection Type, channel 0.
s0_axi4_awready	Output	Write Address Ready, channel 0.
s0_axi4_araddr	Input	Read Address , channel 0.
s0_axi4_arburst	Input	Read Burst Type, channel 0.
s0_axi4_arid	Input	Read Address ID, channel 0.
s0_axi4_arlen	Input	Read Burst Length, channel 0.
s0_axi4_arlock	Input	Read Lock Type, channel 0.
s0_axi4_arqos	Input	Read Quality of Service, channel 0.
s0_axi4_arsz	Input	Read Burst Size, channel 0.
s0_axi4_arvalid	Input	Read Address Valid, channel 0.
s0_axi4_aruser	Input	Read Address User Signal, channel 0.
s0_axi4_arprot	Input	Read Protection Type, channel 0.
s0_axi4_arready	Output	Read Address Ready, channel 0.
s0_axi4_wdata	Input	Write Data , channel 0.
s0_axi4_wstrb	Input	Write Strobes, channel 0.
s0_axi4_wlast	Input	Write Last, channel 0.
continued...		

Port Name	Direction	Description
s0_axi4_wvalid	Input	Write Valid, channel 0.
s0_axi4_wready	Output	Write Ready, channel 0.
s0_axi4_bready	Input	Write Response Ready, channel 0.
s0_axi4_bid	Output	Write Response ID, channel 0.
s0_axi4_bresp	Output	Write Response , channel 0.
s0_axi4_bvalid	Output	Write Response Valid, channel 0.
s0_axi4_rready	Input	Read Ready, channel 0.
s0_axi4_rdata	Output	Read Data, channel 0.
s0_axi4_rid	Output	Read ID , channel 0.
s0_axi4_rlast	Output	Read Last, channel 0.
s0_axi4_rresp	Output	Read Response, channel 0.
s0_axi4_rvalid	Output	Read Valid, channel 0.

4.2.6. s1_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Mainband AXI4 from fabric to controller, channel 1.

Table 47. Interface: s1_axi4

Interface type: axi4

Port Name	Direction	Description
s1_axi4_awaddr	Input	Write Address , channel 1.
s1_axi4_awburst	Input	Write Burst Type, channel 1.
s1_axi4_awid	Input	Write Address ID, channel 1.
s1_axi4_awlen	Input	Write Burst Length, channel 1.
s1_axi4_awlock	Input	Write Lock Type, channel 1.
s1_axi4_awqos	Input	Write Quality of Service, channel 1.
s1_axi4_awsz	Input	Write Burst Size, channel 1.
s1_axi4_awvalid	Input	Write Address Valid, channel 1.
s1_axi4_awuser	Input	Write Address User Signal, channel 1.
s1_axi4_awprot	Input	Write Protection Type, channel 1.
s1_axi4_awready	Output	Write Address Ready, channel 1.
s1_axi4_araddr	Input	Read Address , channel 1.
s1_axi4_arburst	Input	Read Burst Type, channel 1.
s1_axi4_arid	Input	Read Address ID, channel 1.
s1_axi4_arlen	Input	Read Burst Length, channel 1.
s1_axi4_arlock	Input	Read Lock Type, channel 1.

continued...

Port Name	Direction	Description
s1_axi4_arqos	Input	Read Quality of Service, channel 1.
s1_axi4_arsize	Input	Read Burst Size, channel 1.
s1_axi4_arvalid	Input	Read Address Valid, channel 1.
s1_axi4_aruser	Input	Read Address User Signal, channel 1.
s1_axi4_arprot	Input	Read Protection Type, channel 1.
s1_axi4_arready	Output	Read Address Ready, channel 1.
s1_axi4_wdata	Input	Write Data , channel 1.
s1_axi4_wstrb	Input	Write Strobes, channel 1.
s1_axi4_wlast	Input	Write Last, channel 1.
s1_axi4_wvalid	Input	Write Valid, channel 1.
s1_axi4_wready	Output	Write Ready, channel 1.
s1_axi4_bready	Input	Write Response Ready, channel 1.
s1_axi4_bid	Output	Write Response ID, channel 1.
s1_axi4_bresp	Output	Write Response , channel 1.
s1_axi4_bvalid	Output	Write Response Valid, channel 1.
s1_axi4_rready	Input	Read Ready, channel 1.
s1_axi4_rdata	Output	Read Data, channel 1.
s1_axi4_rid	Output	Read ID , channel 1.
s1_axi4_rlast	Output	Read Last, channel 1.
s1_axi4_rresp	Output	Read Response, channel 1.
s1_axi4_rvalid	Output	Read Valid, channel 1.

4.2.7. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Clock for sideband interface (primary I/O bank).

Table 48. Interface: s0_axi4lite_clock

Interface type: clock

Port Name	Direction	Description
s0_axi4lite_clock	Input	Axi-Lite clock, to primary IOSSM.

4.2.8. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Reset for sideband interface (primary I/O bank).

Table 49. Interface: s0_axi4lite_reset_n

Interface type: reset

Port Name	Direction	Description
s0_axi4lite_reset_n	Input	Axi-Lite reset_n, to primary IOSSM.

4.2.9. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.

Table 50. Interface: s0_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s0_axi4lite_awaddr	Input	Axi-Lite Write Address, to primary IOSSM.
s0_axi4lite_awprot	Input	Axi-Lite Write Address Protection Signal, to primary IOSSM.
s0_axi4lite_awvalid	Input	Axi-Lite Write Address Valid, to primary IOSSM.
s0_axi4lite_awready	Output	Axi-Lite Write Address Ready, to primary IOSSM.
s0_axi4lite_araddr	Input	Axi-Lite Read Address, to primary IOSSM.
s0_axi4lite_arprot	Input	Axi-Lite Read Address Protection Signal, to primary IOSSM.
s0_axi4lite_arvalid	Input	Axi-Lite Read Address Valid, to primary IOSSM.
s0_axi4lite_arready	Output	Axi-Lite Read Address Ready, to primary IOSSM.
s0_axi4lite_wdata	Input	Axi-Lite Write Data, to primary IOSSM.
s0_axi4lite_wstrb	Input	Axi-Lite Write Strobe, to primary IOSSM.
s0_axi4lite_wvalid	Input	Axi-Lite Write Valid, to primary IOSSM.
s0_axi4lite_wready	Output	Axi-Lite Write Ready, to primary IOSSM.
s0_axi4lite_bready	Input	Axi-Lite Write Response Ready, to primary IOSSM.
s0_axi4lite_bresp	Output	Axi-Lite Write Response, to primary IOSSM.
s0_axi4lite_bvalid	Output	Axi-Lite Write Response Valid, to primary IOSSM.
s0_axi4lite_rready	Input	Axi-Lite Read Ready, to primary IOSSM.
s0_axi4lite_rdata	Output	Axi-Lite Read Data, to primary IOSSM.
s0_axi4lite_rresp	Output	Axi-Lite Read Response, to primary IOSSM.
s0_axi4lite_rvalid	Output	Axi-Lite Read Valid, to primary IOSSM.

4.2.10. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.

Table 51. Interface: mem_0

Interface type: conduit

Port Name	Direction	Description
mem_0_cs_n	Output	Chip Select channel 0.
mem_0_ca	Output	Command/Address Bus channel 0.
mem_0_dq	Bidir	Data (read/write) channel 0.
mem_0_dqs_t	Bidir	Data Strobe (true) channel 0.
mem_0_dqs_c	Bidir	Data Strobe (complement) channel 0.
mem_0_alert_n	Input	Indicates Write CRC Error channel 0.

4.2.11. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Clock pin to the memory (channel 0).

Table 52. Interface: mem_ck_0

Interface type: conduit

Port Name	Direction	Description
mem_0_ck_t	Output	CK Clock (true) channel 0.
mem_0_ck_c	Output	CK Clock (complement) channel 0.

4.2.12. mem_reset_n_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Reset pin to the memory (channel 0).

Table 53. Interface: mem_reset_n_0

Interface type: conduit

Port Name	Direction	Description
mem_0_reset_n	Output	Asynchronous Reset channel 0.

4.2.13. mem_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Interface to the memory (channel 1), including all CA pins, DQ pins, and DQS pins.

Table 54. Interface: mem_1

Interface type: conduit

Port Name	Direction	Description
mem_1_cs_n	Output	Chip Select channel 1.
mem_1_ca	Output	Command/Address Bus channel 1.
<i>continued...</i>		

Port Name	Direction	Description
mem_1_dq	Bidir	Data (read/write) channel 1.
mem_1_dqs_t	Bidir	Data Strobe (true) channel 1.
mem_1_dqs_c	Bidir	Data Strobe (complement) channel 1.
mem_1_alert_n	Input	Indicates Write CRC Error channel 1.

4.2.14. mem_ck_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Clock pin to the memory (channel 1).

Table 55. Interface: mem_ck_1

Interface type: conduit

Port Name	Direction	Description
mem_1_ck_t	Output	CK Clock (true) channel 1.
mem_1_ck_c	Output	CK Clock (complement) channel 1.

4.2.15. mem_reset_n_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Reset pin to the memory (channel 1).

Table 56. Interface: mem_reset_n_1

Interface type: conduit

Port Name	Direction	Description
mem_1_reset_n	Output	Asynchronous Reset channel 1.

4.2.16. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).

Table 57. Interface: oct_0

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_0	Input	Calibrated On-Chip Termination (OCT) input pin channel 0.

4.2.17. oct_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

On-Chip Termination (OCT) interface, representing RZQ pin (channel 1).

Table 58. Interface: oct_1

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_1	Input	Calibrated On-Chip Termination (OCT) input pin channel 1.

4.2.18. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - DDR5 Component

Reference clock used by the EMIF PLL.

Table 59. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	Input	PLL reference clock input.

4.3. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

The interfaces in the Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4 each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 60. Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Interface Name	Interface Type	Description
s0_axi4_clock_in	clock	Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.
core_init_n	reset	An input to indicate that core configuration is complete.
s0_axi4_ctrl_ready	reset	Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.
s0_axi4_clock_out	clock	Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.
s1_axi4_ctrl_ready	reset	Reset for mainband, from secondary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.
s1_axi4_clock_out	clock	Output user clock for mainband (from CPA of secondary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.
s0_axi4	axi4	Mainband AXI4 from fabric to controller, channel 0.
s1_axi4	axi4	Mainband AXI4 from fabric to controller, channel 1.
s2_axi4	axi4	Mainband AXI4 from fabric to controller, channel 2.
s3_axi4	axi4	Mainband AXI4 from fabric to controller, channel 3.
s0_axi4lite_clock	clock	Clock for sideband interface (primary I/O bank).
s0_axi4lite_reset_n	reset	Reset for sideband interface (primary I/O bank).
continued...		

Interface Name	Interface Type	Description
s0_axi4lite	axi4lite	Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.
s1_axi4lite_clock	clock	Clock for sideband interface (secondary I/O bank).
s1_axi4lite_reset_n	reset	Reset for sideband interface (secondary I/O bank).
s1_axi4lite	axi4lite	Sideband interface (secondary I/O bank) that will connect to the IOSSM, through a gearbox in the core.
mem_0	conduit	Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.
mem_ck_0	conduit	Clock pin to the memory (channel 0).
mem_1	conduit	Interface to the memory (channel 1), including all CA pins, DQ pins, and DQS pins.
mem_ck_1	conduit	Clock pin to the memory (channel 1).
mem_2	conduit	Interface to the memory (channel 2), including all CA pins, DQ pins, and DQS pins.
mem_ck_2	conduit	Clock pin to the memory (channel 2).
mem_3	conduit	Interface to the memory (channel 3), including all CA pins, DQ pins, and DQS pins.
mem_ck_3	conduit	Clock pin to the memory (channel 3).
mem_reset_n	conduit	Reset pin to the memory. Must always be placed along with channel 0, but shared for entire interface (all channels within one EMIF).
oct_0	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).
oct_1	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 1).
oct_2	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 2).
oct_3	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 3).
ref_clk	clock	Reference clock used by the EMIF PLL.

4.3.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.

Table 61. Interface: s0_axi4_clock_in

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_in	Input	User clock for mainband axi. Input clock to the EMIF IP, no relationship to PHY clock.

4.3.2. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

An input to indicate that core configuration is complete.

Table 62. 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. When high, indicates core initialization is complete.

4.3.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.

Table 63. Interface: s0_axi4_ctrl_ready

Interface type: reset

Port Name	Direction	Description
s0_axi4_reset_n	Output	Output signal from EMIF IP (primary I/O bank), indicating that Calibration of the channels in this I/O bank is complete, and controllers in this I/O bank are ready for use.

4.3.4. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.

Table 64. Interface: s0_axi4_clock_out

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_out	Output	User clock for maiband axi (primary I/O bank). Output clock from the EMIF IP (output from CPA block, synchronous to PHY clock).

4.3.5. s1_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Reset for mainband, from secondary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.

Table 65. Interface: s1_axi4_ctrl_ready

Interface type: reset

Port Name	Direction	Description
s1_axi4_reset_n	Output	Output signal from EMIF IP (secondary I/O bank), indicating that Calibration of the channels in this I/O bank is complete, and controllers in this I/O bank are ready for use.

4.3.6. s1_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Output user clock for mainband (from CPA of secondary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.

Table 66. Interface: s1_axi4_clock_out

Interface type: clock

Port Name	Direction	Description
s1_axi4_clock_out	Output	User clock for maiband axi (secondary I/O bank). Output clock from the EMIF IP (output from CPA block, synchronous to PHY clock).

4.3.7. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Mainband AXI4 from fabric to controller, channel 0.

Table 67. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
s0_axi4_awaddr	Input	Write Address , channel 0.
s0_axi4_awburst	Input	Write Burst Type, channel 0.
s0_axi4_awid	Input	Write Address ID, channel 0.
s0_axi4_awlen	Input	Write Burst Length, channel 0.
s0_axi4_awlock	Input	Write Lock Type, channel 0.
s0_axi4_awqos	Input	Write Quality of Service, channel 0.
s0_axi4_awsz	Input	Write Burst Size, channel 0.
s0_axi4_awvalid	Input	Write Address Valid, channel 0.
s0_axi4_awuser	Input	Write Address User Signal, channel 0.
s0_axi4_awprot	Input	Write Protection Type, channel 0.
s0_axi4_awready	Output	Write Address Ready, channel 0.
s0_axi4_araddr	Input	Read Address , channel 0.
s0_axi4_arburst	Input	Read Burst Type, channel 0.
s0_axi4_arid	Input	Read Address ID, channel 0.

continued...

Port Name	Direction	Description
s0_axi4_arlen	Input	Read Burst Length, channel 0.
s0_axi4_arlock	Input	Read Lock Type, channel 0.
s0_axi4_arqos	Input	Read Quality of Service, channel 0.
s0_axi4_arsize	Input	Read Burst Size, channel 0.
s0_axi4_arvalid	Input	Read Address Valid, channel 0.
s0_axi4_aruser	Input	Read Address User Signal, channel 0.
s0_axi4_arprot	Input	Read Protection Type, channel 0.
s0_axi4_arready	Output	Read Address Ready, channel 0.
s0_axi4_wdata	Input	Write Data , channel 0.
s0_axi4_wstrb	Input	Write Strobes, channel 0.
s0_axi4_wlast	Input	Write Last, channel 0.
s0_axi4_wvalid	Input	Write Valid, channel 0.
s0_axi4_wready	Output	Write Ready, channel 0.
s0_axi4_bready	Input	Write Response Ready, channel 0.
s0_axi4_bid	Output	Write Response ID, channel 0.
s0_axi4_bresp	Output	Write Response , channel 0.
s0_axi4_bvalid	Output	Write Response Valid, channel 0.
s0_axi4_rready	Input	Read Ready, channel 0.
s0_axi4_rdata	Output	Read Data, channel 0.
s0_axi4_rid	Output	Read ID , channel 0.
s0_axi4_rlast	Output	Read Last, channel 0.
s0_axi4_rresp	Output	Read Response, channel 0.
s0_axi4_rvalid	Output	Read Valid, channel 0.

4.3.8. s1_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Mainband AXI4 from fabric to controller, channel 1.

Table 68. Interface: s1_axi4

Interface type: axi4

Port Name	Direction	Description
s1_axi4_awaddr	Input	Write Address , channel 1.
s1_axi4_awburst	Input	Write Burst Type, channel 1.
s1_axi4_awid	Input	Write Address ID, channel 1.
s1_axi4_awlen	Input	Write Burst Length, channel 1.
s1_axi4_awlock	Input	Write Lock Type, channel 1.
<i>continued...</i>		

Port Name	Direction	Description
s1_axi4_awqos	Input	Write Quality of Service, channel 1.
s1_axi4_awsz	Input	Write Burst Size, channel 1.
s1_axi4_awvalid	Input	Write Address Valid, channel 1.
s1_axi4_awuser	Input	Write Address User Signal, channel 1.
s1_axi4_awprot	Input	Write Protection Type, channel 1.
s1_axi4_awready	Output	Write Address Ready, channel 1.
s1_axi4_araddr	Input	Read Address , channel 1.
s1_axi4_arburst	Input	Read Burst Type, channel 1.
s1_axi4_arid	Input	Read Address ID, channel 1.
s1_axi4_arlen	Input	Read Burst Length, channel 1.
s1_axi4_arlock	Input	Read Lock Type, channel 1.
s1_axi4_arqos	Input	Read Quality of Service, channel 1.
s1_axi4_arsz	Input	Read Burst Size, channel 1.
s1_axi4_arvalid	Input	Read Address Valid, channel 1.
s1_axi4_aruser	Input	Read Address User Signal, channel 1.
s1_axi4_arprot	Input	Read Protection Type, channel 1.
s1_axi4_arready	Output	Read Address Ready, channel 1.
s1_axi4_wdata	Input	Write Data , channel 1.
s1_axi4_wstrb	Input	Write Strobcs, channel 1.
s1_axi4_wlast	Input	Write Last, channel 1.
s1_axi4_wvalid	Input	Write Valid, channel 1.
s1_axi4_wready	Output	Write Ready, channel 1.
s1_axi4_bready	Input	Write Response Ready, channel 1.
s1_axi4_bid	Output	Write Response ID, channel 1.
s1_axi4_bresp	Output	Write Response , channel 1.
s1_axi4_bvalid	Output	Write Response Valid, channel 1.
s1_axi4_rready	Input	Read Ready, channel 1.
s1_axi4_rdata	Output	Read Data, channel 1.
s1_axi4_rid	Output	Read ID , channel 1.
s1_axi4_rlast	Output	Read Last, channel 1.
s1_axi4_rresp	Output	Read Response, channel 1.
s1_axi4_rvalid	Output	Read Valid, channel 1.

4.3.9. s2_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Mainband AXI4 from fabric to controller, channel 2.

Table 69. Interface: s2_axi4

Interface type: axi4

Port Name	Direction	Description
s2_axi4_awaddr	Input	Write Address , channel 2.
s2_axi4_awburst	Input	Write Burst Type, channel 2.
s2_axi4_awid	Input	Write Address ID, channel 2.
s2_axi4_awlen	Input	Write Burst Length, channel 2.
s2_axi4_awlock	Input	Write Lock Type, channel 2.
s2_axi4_awqos	Input	Write Quality of Service, channel 2.
s2_axi4_awsz	Input	Write Burst Size, channel 2.
s2_axi4_awvalid	Input	Write Address Valid, channel 2.
s2_axi4_awuser	Input	Write Address User Signal, channel 2.
s2_axi4_awprot	Input	Write Protection Type, channel 2.
s2_axi4_awready	Output	Write Address Ready, channel 2.
s2_axi4_araddr	Input	Read Address , channel 2.
s2_axi4_arburst	Input	Read Burst Type, channel 2.
s2_axi4_arid	Input	Read Address ID, channel 2.
s2_axi4_arlen	Input	Read Burst Length, channel 2.
s2_axi4_arlock	Input	Read Lock Type, channel 2.
s2_axi4_arqos	Input	Read Quality of Service, channel 2.
s2_axi4_arsz	Input	Read Burst Size, channel 2.
s2_axi4_arvalid	Input	Read Address Valid, channel 2.
s2_axi4_aruser	Input	Read Address User Signal, channel 2.
s2_axi4_arprot	Input	Read Protection Type, channel 2.
s2_axi4_arready	Output	Read Address Ready, channel 2.
s2_axi4_wdata	Input	Write Data , channel 2.
s2_axi4_wstrb	Input	Write Strobes, channel 2.
s2_axi4_wlast	Input	Write Last, channel 2.
s2_axi4_wvalid	Input	Write Valid, channel 2.
s2_axi4_wready	Output	Write Ready, channel 2.
s2_axi4_bready	Input	Write Response Ready, channel 2.
s2_axi4_bid	Output	Write Response ID, channel 2.
s2_axi4_bresp	Output	Write Response , channel 2.
s2_axi4_bvalid	Output	Write Response Valid, channel 2.
s2_axi4_rready	Input	Read Ready, channel 2.
s2_axi4_rdata	Output	Read Data, channel 2.
s2_axi4_rid	Output	Read ID , channel 2.

continued...

Port Name	Direction	Description
s2_axi4_rlast	Output	Read Last, channel 2.
s2_axi4_rresp	Output	Read Response, channel 2.
s2_axi4_rvalid	Output	Read Valid, channel 2.

4.3.10. s3_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Mainband AXI4 from fabric to controller, channel 3.

Table 70. Interface: s3_axi4

Interface type: axi4

Port Name	Direction	Description
s3_axi4_awaddr	Input	Write Address , channel 3.
s3_axi4_awburst	Input	Write Burst Type, channel 3.
s3_axi4_awid	Input	Write Address ID, channel 3.
s3_axi4_awlen	Input	Write Burst Length, channel 3.
s3_axi4_awlock	Input	Write Lock Type, channel 3.
s3_axi4_awqos	Input	Write Quality of Service, channel 3.
s3_axi4_awsiz	Input	Write Burst Size, channel 3.
s3_axi4_awvalid	Input	Write Address Valid, channel 3.
s3_axi4_awuser	Input	Write Address User Signal, channel 3.
s3_axi4_awprot	Input	Write Protection Type, channel 3.
s3_axi4_awready	Output	Write Address Ready, channel 3.
s3_axi4_araddr	Input	Read Address , channel 3.
s3_axi4_arburst	Input	Read Burst Type, channel 3.
s3_axi4_arid	Input	Read Address ID, channel 3.
s3_axi4_arlen	Input	Read Burst Length, channel 3.
s3_axi4_arlock	Input	Read Lock Type, channel 3.
s3_axi4_arqos	Input	Read Quality of Service, channel 3.
s3_axi4_arsiz	Input	Read Burst Size, channel 3.
s3_axi4_arvalid	Input	Read Address Valid, channel 3.
s3_axi4_aruser	Input	Read Address User Signal, channel 3.
s3_axi4_arprot	Input	Read Protection Type, channel 3.
s3_axi4_arready	Output	Read Address Ready, channel 3.
s3_axi4_wdata	Input	Write Data , channel 3.
s3_axi4_wstrb	Input	Write Stobes, channel 3.
s3_axi4_wlast	Input	Write Last, channel 3.

continued...

Port Name	Direction	Description
s3_axi4_wvalid	Input	Write Valid, channel 3.
s3_axi4_wready	Output	Write Ready, channel 3.
s3_axi4_bready	Input	Write Response Ready, channel 3.
s3_axi4_bid	Output	Write Response ID, channel 3.
s3_axi4_bresp	Output	Write Response , channel 3.
s3_axi4_bvalid	Output	Write Response Valid, channel 3.
s3_axi4_rready	Input	Read Ready, channel 3.
s3_axi4_rdata	Output	Read Data, channel 3.
s3_axi4_rid	Output	Read ID , channel 3.
s3_axi4_rlast	Output	Read Last, channel 3.
s3_axi4_rresp	Output	Read Response, channel 3.
s3_axi4_rvalid	Output	Read Valid, channel 3.

4.3.11. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Clock for sideband interface (primary I/O bank).

Table 71. Interface: s0_axi4lite_clock

Interface type: clock

Port Name	Direction	Description
s0_axi4lite_clock	Input	Axi-Lite clock, to primary IOSSM.

4.3.12. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Reset for sideband interface (primary I/O bank).

Table 72. Interface: s0_axi4lite_reset_n

Interface type: reset

Port Name	Direction	Description
s0_axi4lite_reset_n	Input	Axi-Lite reset_n, to primary IOSSM.

4.3.13. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.

Table 73. Interface: s0_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s0_axi4lite_awaddr	Input	Axi-Lite Write Address, to primary IOSSM.
s0_axi4lite_awprot	Input	Axi-Lite Write Address Protection Signal, to primary IOSSM.
s0_axi4lite_awvalid	Input	Axi-Lite Write Address Valid, to primary IOSSM.
s0_axi4lite_awready	Output	Axi-Lite Write Address Ready, to primary IOSSM.
s0_axi4lite_araddr	Input	Axi-Lite Read Address, to primary IOSSM.
s0_axi4lite_arprot	Input	Axi-Lite Read Address Protection Signal, to primary IOSSM.
s0_axi4lite_arvalid	Input	Axi-Lite Read Address Valid, to primary IOSSM.
s0_axi4lite_arready	Output	Axi-Lite Read Address Ready, to primary IOSSM.
s0_axi4lite_wdata	Input	Axi-Lite Write Data, to primary IOSSM.
s0_axi4lite_wstrb	Input	Axi-Lite Write Strobe, to primary IOSSM.
s0_axi4lite_wvalid	Input	Axi-Lite Write Valid, to primary IOSSM.
s0_axi4lite_wready	Output	Axi-Lite Write Ready, to primary IOSSM.
s0_axi4lite_bready	Input	Axi-Lite Write Response Ready, to primary IOSSM.
s0_axi4lite_bresp	Output	Axi-Lite Write Response, to primary IOSSM.
s0_axi4lite_bvalid	Output	Axi-Lite Write Response Valid, to primary IOSSM.
s0_axi4lite_rready	Input	Axi-Lite Read Ready, to primary IOSSM.
s0_axi4lite_rdata	Output	Axi-Lite Read Data, to primary IOSSM.
s0_axi4lite_rresp	Output	Axi-Lite Read Response, to primary IOSSM.
s0_axi4lite_rvalid	Output	Axi-Lite Read Valid, to primary IOSSM.

4.3.14. s1_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Clock for sideband interface (secondary I/O bank).

Table 74. Interface: s1_axi4lite_clock

Interface type: clock

Port Name	Direction	Description
s1_axi4lite_clock	Input	Axi-Lite clock, to secondary IOSSM.

4.3.15. s1_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Reset for sideband interface (secondary I/O bank).

Table 75. Interface: s1_axi4lite_reset_n

Interface type: reset

Port Name	Direction	Description
s1_axi4lite_reset_n	Input	Axi-Lite reset_n, to secondary IOSSM.

4.3.16. s1_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Sideband interface (secondary I/O bank) that will connect to the IOSSM, through a gearbox in the core.

Table 76. Interface: s1_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s1_axi4lite_awaddr	Input	Axi-Lite Write Address, to secondary IOSSM.
s1_axi4lite_awprot	Input	Axi-Lite Write Address Protection Signal, to secondary IOSSM.
s1_axi4lite_awvalid	Input	Axi-Lite Write Address Valid, to secondary IOSSM.
s1_axi4lite_awready	Output	Axi-Lite Write Address Ready, to secondary IOSSM.
s1_axi4lite_araddr	Input	Axi-Lite Read Address, to secondary IOSSM.
s1_axi4lite_arprot	Input	Axi-Lite Read Address Protection Signal, to secondary IOSSM.
s1_axi4lite_arvalid	Input	Axi-Lite Read Address Valid, to secondary IOSSM.
s1_axi4lite_arready	Output	Axi-Lite Read Address Ready, to secondary IOSSM.
s1_axi4lite_wdata	Input	Axi-Lite Write Data, to secondary IOSSM.
s1_axi4lite_wstrb	Input	Axi-Lite Write Strobe, to secondary IOSSM.
s1_axi4lite_wvalid	Input	Axi-Lite Write Valid, to secondary IOSSM.
s1_axi4lite_wready	Output	Axi-Lite Write Ready, to secondary IOSSM.
s1_axi4lite_bready	Input	Axi-Lite Write Response Ready, to secondary IOSSM.
s1_axi4lite_bresp	Output	Axi-Lite Write Response, to secondary IOSSM.
s1_axi4lite_bvalid	Output	Axi-Lite Write Response Valid, to secondary IOSSM.
s1_axi4lite_rready	Input	Axi-Lite Read Ready, to secondary IOSSM.
s1_axi4lite_rdata	Output	Axi-Lite Read Data, to secondary IOSSM.
s1_axi4lite_rresp	Output	Axi-Lite Read Response, to secondary IOSSM.
s1_axi4lite_rvalid	Output	Axi-Lite Read Valid, to secondary IOSSM.

4.3.17. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.

Table 77. Interface: mem_0

Interface type: conduit

Port Name	Direction	Description
mem_0_cs	Output	Chip Select channel 0.
mem_0_ca	Output	Command/Address Bus channel 0.
mem_0_cke	Output	Clock Enable channel 0.
mem_0_dq	Bidir	Data (read/write) channel 0.
mem_0_dqs_t	Bidir	Data Strobe (true) channel 0.
mem_0_dqs_c	Bidir	Data Strobe (complement) channel 0.
mem_0_dmi	Bidir	Data Mask/Data Inversion channel 0.

4.3.18. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Clock pin to the memory (channel 0).

Table 78. Interface: mem_ck_0

Interface type: conduit

Port Name	Direction	Description
mem_0_ck_t	Output	CK Clock (true) channel 0.
mem_0_ck_c	Output	CK Clock (complement) channel 0.

4.3.19. mem_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Interface to the memory (channel 1), including all CA pins, DQ pins, and DQS pins.

Table 79. Interface: mem_1

Interface type: conduit

Port Name	Direction	Description
mem_1_cs	Output	Chip Select channel 1.
mem_1_ca	Output	Command/Address Bus channel 1.
mem_1_cke	Output	Clock Enable channel 1.
mem_1_dq	Bidir	Data (read/write) channel 1.
mem_1_dqs_t	Bidir	Data Strobe (true) channel 1.
mem_1_dqs_c	Bidir	Data Strobe (complement) channel 1.
mem_1_dmi	Bidir	Data Mask/Data Inversion channel 1.

4.3.20. mem_ck_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Clock pin to the memory (channel 1).

Table 80. Interface: mem_ck_1

Interface type: conduit

Port Name	Direction	Description
mem_1_ck_t	Output	CK Clock (true) channel 1.
mem_1_ck_c	Output	CK Clock (complement) channel 1.

4.3.21. mem_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Interface to the memory (channel 2), including all CA pins, DQ pins, and DQS pins.

Table 81. Interface: mem_2

Interface type: conduit

Port Name	Direction	Description
mem_2_cs	Output	Chip Select channel 2.
mem_2_ca	Output	Command/Address Bus channel 2.
mem_2_cke	Output	Clock Enable channel 2.
mem_2_dq	Bidir	Data (read/write) channel 2.
mem_2_dqs_t	Bidir	Data Strobe (true) channel 2.
mem_2_dqs_c	Bidir	Data Strobe (complement) channel 2.
mem_2_dmi	Bidir	Data Mask/Data Inversion channel 2.

4.3.22. mem_ck_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Clock pin to the memory (channel 2).

Table 82. Interface: mem_ck_2

Interface type: conduit

Port Name	Direction	Description
mem_2_ck_t	Output	CK Clock (true) channel 2.
mem_2_ck_c	Output	CK Clock (complement) channel 2.

4.3.23. mem_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Interface to the memory (channel 3), including all CA pins, DQ pins, and DQS pins.

Table 83. Interface: mem_3

Interface type: conduit

Port Name	Direction	Description
mem_3_cs	Output	Chip Select channel 3.
mem_3_ca	Output	Command/Address Bus channel 3.
mem_3_cke	Output	Clock Enable channel 3.
mem_3_dq	Bidir	Data (read/write) channel 3.
mem_3_dqs_t	Bidir	Data Strobe (true) channel 3.
mem_3_dqs_c	Bidir	Data Strobe (complement) channel 3.
mem_3_dmi	Bidir	Data Mask/Data Inversion channel 3.

4.3.24. mem_ck_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Clock pin to the memory (channel 3).

Table 84. Interface: mem_ck_3

Interface type: conduit

Port Name	Direction	Description
mem_3_ck_t	Output	CK Clock (true) channel 3.
mem_3_ck_c	Output	CK Clock (complement) channel 3.

4.3.25. mem_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Reset pin to the memory. Must always be placed along with channel 0, but shared for entire interface (all channels within one EMIF).

Table 85. Interface: mem_reset_n

Interface type: conduit

Port Name	Direction	Description
mem_0_reset_n	Output	Asynchronous Reset channel 0.

4.3.26. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).

Table 86. Interface: oct_0

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_0	Input	Calibrated On-Chip Termination (OCT) input pin channel 0.

4.3.27. oct_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

On-Chip Termination (OCT) interface, representing RZQ pin (channel 1).

Table 87. Interface: oct_1

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_1	Input	Calibrated On-Chip Termination (OCT) input pin channel 1.

4.3.28. oct_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

On-Chip Termination (OCT) interface, representing RZQ pin (channel 2).

Table 88. Interface: oct_2

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_2	Input	Calibrated On-Chip Termination (OCT) input pin channel 2.

4.3.29. oct_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

On-Chip Termination (OCT) interface, representing RZQ pin (channel 3).

Table 89. Interface: oct_3

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_3	Input	Calibrated On-Chip Termination (OCT) input pin channel 3.

4.3.30. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR4

Reference clock used by the EMIF PLL.

Table 90. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	Input	PLL reference clock input.

4.4. IP Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

The interfaces in the Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5 each have signals that can be connected in Platform Designer. The following table lists the interfaces and corresponding interface types.

Table 91. Interfaces for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Interface Name	Interface Type	Description
s0_axi4_clock_in	clock	Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.
core_init_n	reset	An input to indicate that core configuration is complete.
s0_axi4_ctrl_ready	reset	Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.
s0_axi4_clock_out	clock	Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.
s1_axi4_ctrl_ready	reset	Reset for mainband, from secondary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.
s1_axi4_clock_out	clock	Output user clock for mainband (from CPA of secondary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.
s0_axi4	axi4	Mainband AXI4 from fabric to controller, channel 0.
s1_axi4	axi4	Mainband AXI4 from fabric to controller, channel 1.
s2_axi4	axi4	Mainband AXI4 from fabric to controller, channel 2.
s3_axi4	axi4	Mainband AXI4 from fabric to controller, channel 3.
s0_axi4lite_clock	clock	Clock for sideband interface (primary I/O bank).
s0_axi4lite_reset_n	reset	Reset for sideband interface (primary I/O bank).
s0_axi4lite	axi4lite	Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.
s1_axi4lite_clock	clock	Clock for sideband interface (secondary I/O bank).
s1_axi4lite_reset_n	reset	Reset for sideband interface (secondary I/O bank).
s1_axi4lite	axi4lite	Sideband interface (secondary I/O bank) that will connect to the IOSSM, through a gearbox in the core.
mem_0	conduit	Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.
mem_ck_0	conduit	Clock pin to the memory (channel 0).
mem_1	conduit	Interface to the memory (channel 1), including all CA pins, DQ pins, and DQS pins.
mem_ck_1	conduit	Clock pin to the memory (channel 1).
mem_2	conduit	Interface to the memory (channel 2), including all CA pins, DQ pins, and DQS pins.
mem_ck_2	conduit	Clock pin to the memory (channel 2).
continued...		

Interface Name	Interface Type	Description
mem_3	conduit	Interface to the memory (channel 3), including all CA pins, DQ pins, and DQS pins.
mem_ck_3	conduit	Clock pin to the memory (channel 3).
mem_reset_n	conduit	Reset pin to the memory. Must always be placed along with channel 0, but shared for entire interface (all channels within one EMIF).
oct_0	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).
oct_1	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 1).
oct_2	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 2).
oct_3	conduit	On-Chip Termination (OCT) interface, representing RZQ pin (channel 3).
ref_clk	clock	Reference clock used by the EMIF PLL.

4.4.1. s0_axi4_clock_in for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Input user clock for mainband; for MAINBAND_ACCESS_MODE = ASYNC only.

Table 92. Interface: s0_axi4_clock_in

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_in	Input	User clock for mainband axi. Input clock to the EMIF IP, no relationship to PHY clock.

4.4.2. core_init_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

An input to indicate that core configuration is complete.

Table 93. 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. When high, indicates core initialization is complete.

4.4.3. s0_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Reset for mainband, from primary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.

Table 94. Interface: s0_axi4_ctrl_ready

Interface type: reset

Port Name	Direction	Description
s0_axi4_reset_n	Output	Output signal from EMIF IP (primary I/O bank), indicating that Calibration of the channels in this I/O bank is complete, and controllers in this I/O bank are ready for use.

4.4.4. s0_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Output user clock for mainband (from CPA of primary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.

Table 95. Interface: s0_axi4_clock_out

Interface type: clock

Port Name	Direction	Description
s0_axi4_clock_out	Output	User clock for maiband axi (primary I/O bank). Output clock from the EMIF IP (output from CPA block, synchronous to PHY clock).

4.4.5. s1_axi4_ctrl_ready for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Reset for mainband, from secondary I/O bank, indicating the calibration is complete. Only available if mainband is accessed through fabric.

Table 96. Interface: s1_axi4_ctrl_ready

Interface type: reset

Port Name	Direction	Description
s1_axi4_reset_n	Output	Output signal from EMIF IP (secondary I/O bank), indicating that Calibration of the channels in this I/O bank is complete, and controllers in this I/O bank are ready for use.

4.4.6. s1_axi4_clock_out for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Output user clock for mainband (from CPA of secondary I/O bank); for MAINBAND_ACCESS_MODE = SYNC only.

Table 97. Interface: s1_axi4_clock_out

Interface type: clock

Port Name	Direction	Description
s1_axi4_clock_out	Output	User clock for maiband axi (secondary I/O bank). Output clock from the EMIF IP (output from CPA block, synchronous to PHY clock).

4.4.7. s0_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Mainband AXI4 from fabric to controller, channel 0.

Table 98. Interface: s0_axi4

Interface type: axi4

Port Name	Direction	Description
s0_axi4_awaddr	Input	Write Address , channel 0.
s0_axi4_awburst	Input	Write Burst Type, channel 0.
s0_axi4_awid	Input	Write Address ID, channel 0.
s0_axi4_awlen	Input	Write Burst Length, channel 0.
s0_axi4_awlock	Input	Write Lock Type, channel 0.
s0_axi4_awqos	Input	Write Quality of Service, channel 0.
s0_axi4_awsz	Input	Write Burst Size, channel 0.
s0_axi4_awvalid	Input	Write Address Valid, channel 0.
s0_axi4_awuser	Input	Write Address User Signal, channel 0.
s0_axi4_awprot	Input	Write Protection Type, channel 0.
s0_axi4_awready	Output	Write Address Ready, channel 0.
s0_axi4_araddr	Input	Read Address , channel 0.
s0_axi4_arburst	Input	Read Burst Type, channel 0.
s0_axi4_arid	Input	Read Address ID, channel 0.
s0_axi4_arlen	Input	Read Burst Length, channel 0.
s0_axi4_arlock	Input	Read Lock Type, channel 0.
s0_axi4_arqos	Input	Read Quality of Service, channel 0.
s0_axi4_arsz	Input	Read Burst Size, channel 0.
s0_axi4_arvalid	Input	Read Address Valid, channel 0.
s0_axi4_aruser	Input	Read Address User Signal, channel 0.
s0_axi4_arprot	Input	Read Protection Type, channel 0.
s0_axi4_arready	Output	Read Address Ready, channel 0.
s0_axi4_wdata	Input	Write Data , channel 0.
s0_axi4_wstrb	Input	Write Strobes, channel 0.
s0_axi4_wlast	Input	Write Last, channel 0.
s0_axi4_wvalid	Input	Write Valid, channel 0.
s0_axi4_wready	Output	Write Ready, channel 0.
s0_axi4_bready	Input	Write Response Ready, channel 0.
s0_axi4_bid	Output	Write Response ID, channel 0.
s0_axi4_bresp	Output	Write Response , channel 0.
<i>continued...</i>		

Port Name	Direction	Description
s0_axi4_bvalid	Output	Write Response Valid, channel 0.
s0_axi4_rready	Input	Read Ready, channel 0.
s0_axi4_rdata	Output	Read Data, channel 0.
s0_axi4_rid	Output	Read ID , channel 0.
s0_axi4_rlast	Output	Read Last, channel 0.
s0_axi4_rresp	Output	Read Response, channel 0.
s0_axi4_rvalid	Output	Read Valid, channel 0.

4.4.8. s1_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Mainband AXI4 from fabric to controller, channel 1.

Table 99. Interface: s1_axi4

Interface type: axi4

Port Name	Direction	Description
s1_axi4_awaddr	Input	Write Address , channel 1.
s1_axi4_awburst	Input	Write Burst Type, channel 1.
s1_axi4_awid	Input	Write Address ID, channel 1.
s1_axi4_awlen	Input	Write Burst Length, channel 1.
s1_axi4_awlock	Input	Write Lock Type, channel 1.
s1_axi4_awqos	Input	Write Quality of Service, channel 1.
s1_axi4_awsiz	Input	Write Burst Size, channel 1.
s1_axi4_awvalid	Input	Write Address Valid, channel 1.
s1_axi4_awuser	Input	Write Address User Signal, channel 1.
s1_axi4_awprot	Input	Write Protection Type, channel 1.
s1_axi4_awready	Output	Write Address Ready, channel 1.
s1_axi4_araddr	Input	Read Address , channel 1.
s1_axi4_arburst	Input	Read Burst Type, channel 1.
s1_axi4_arid	Input	Read Address ID, channel 1.
s1_axi4_arlen	Input	Read Burst Length, channel 1.
s1_axi4_arlock	Input	Read Lock Type, channel 1.
s1_axi4_arqos	Input	Read Quality of Service, channel 1.
s1_axi4_arsiz	Input	Read Burst Size, channel 1.
s1_axi4_arvalid	Input	Read Address Valid, channel 1.
s1_axi4_aruser	Input	Read Address User Signal, channel 1.
s1_axi4_arprot	Input	Read Protection Type, channel 1.

continued...

Port Name	Direction	Description
s1_axi4_arready	Output	Read Address Ready, channel 1.
s1_axi4_wdata	Input	Write Data , channel 1.
s1_axi4_wstrb	Input	Write Strobes, channel 1.
s1_axi4_wlast	Input	Write Last, channel 1.
s1_axi4_wvalid	Input	Write Valid, channel 1.
s1_axi4_wready	Output	Write Ready, channel 1.
s1_axi4_bready	Input	Write Response Ready, channel 1.
s1_axi4_bid	Output	Write Response ID, channel 1.
s1_axi4_bresp	Output	Write Response , channel 1.
s1_axi4_bvalid	Output	Write Response Valid, channel 1.
s1_axi4_rready	Input	Read Ready, channel 1.
s1_axi4_rdata	Output	Read Data, channel 1.
s1_axi4_rid	Output	Read ID , channel 1.
s1_axi4_rlast	Output	Read Last, channel 1.
s1_axi4_rresp	Output	Read Response, channel 1.
s1_axi4_rvalid	Output	Read Valid, channel 1.

4.4.9. s2_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Mainband AXI4 from fabric to controller, channel 2.

Table 100. Interface: s2_axi4

Interface type: axi4

Port Name	Direction	Description
s2_axi4_awaddr	Input	Write Address , channel 2.
s2_axi4_awburst	Input	Write Burst Type, channel 2.
s2_axi4_awid	Input	Write Address ID, channel 2.
s2_axi4_awlen	Input	Write Burst Length, channel 2.
s2_axi4_awlock	Input	Write Lock Type, channel 2.
s2_axi4_awqos	Input	Write Quality of Service, channel 2.
s2_axi4_awsz	Input	Write Burst Size, channel 2.
s2_axi4_awvalid	Input	Write Address Valid, channel 2.
s2_axi4_awuser	Input	Write Address User Signal, channel 2.
s2_axi4_awprot	Input	Write Protection Type, channel 2.
s2_axi4_awready	Output	Write Address Ready, channel 2.
s2_axi4_araddr	Input	Read Address , channel 2.

continued...

Port Name	Direction	Description
s2_axi4_arburst	Input	Read Burst Type, channel 2.
s2_axi4_arid	Input	Read Address ID, channel 2.
s2_axi4_arlen	Input	Read Burst Length, channel 2.
s2_axi4_arlock	Input	Read Lock Type, channel 2.
s2_axi4_arqos	Input	Read Quality of Service, channel 2.
s2_axi4_arsize	Input	Read Burst Size, channel 2.
s2_axi4_arvalid	Input	Read Address Valid, channel 2.
s2_axi4_aruser	Input	Read Address User Signal, channel 2.
s2_axi4_arprot	Input	Read Protection Type, channel 2.
s2_axi4_arready	Output	Read Address Ready, channel 2.
s2_axi4_wdata	Input	Write Data , channel 2.
s2_axi4_wstrb	Input	Write Strobes, channel 2.
s2_axi4_wlast	Input	Write Last, channel 2.
s2_axi4_wvalid	Input	Write Valid, channel 2.
s2_axi4_wready	Output	Write Ready, channel 2.
s2_axi4_bready	Input	Write Response Ready, channel 2.
s2_axi4_bid	Output	Write Response ID, channel 2.
s2_axi4_bresp	Output	Write Response , channel 2.
s2_axi4_bvalid	Output	Write Response Valid, channel 2.
s2_axi4_rready	Input	Read Ready, channel 2.
s2_axi4_rdata	Output	Read Data, channel 2.
s2_axi4_rid	Output	Read ID , channel 2.
s2_axi4_rlast	Output	Read Last, channel 2.
s2_axi4_rresp	Output	Read Response, channel 2.
s2_axi4_rvalid	Output	Read Valid, channel 2.

4.4.10. s3_axi4 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Mainband AXI4 from fabric to controller, channel 3.

Table 101. Interface: s3_axi4

Interface type: axi4

Port Name	Direction	Description
s3_axi4_awaddr	Input	Write Address , channel 3.
s3_axi4_awburst	Input	Write Burst Type, channel 3.
s3_axi4_awid	Input	Write Address ID, channel 3.
continued...		

Port Name	Direction	Description
s3_axi4_awlen	Input	Write Burst Length, channel 3.
s3_axi4_awlock	Input	Write Lock Type, channel 3.
s3_axi4_awqos	Input	Write Quality of Service, channel 3.
s3_axi4_awsiz	Input	Write Burst Size, channel 3.
s3_axi4_awvalid	Input	Write Address Valid, channel 3.
s3_axi4_awuser	Input	Write Address User Signal, channel 3.
s3_axi4_awprot	Input	Write Protection Type, channel 3.
s3_axi4_awready	Output	Write Address Ready, channel 3.
s3_axi4_araddr	Input	Read Address , channel 3.
s3_axi4_arburst	Input	Read Burst Type, channel 3.
s3_axi4_arid	Input	Read Address ID, channel 3.
s3_axi4_arlen	Input	Read Burst Length, channel 3.
s3_axi4_arlock	Input	Read Lock Type, channel 3.
s3_axi4_arqos	Input	Read Quality of Service, channel 3.
s3_axi4_arsiz	Input	Read Burst Size, channel 3.
s3_axi4_arvalid	Input	Read Address Valid, channel 3.
s3_axi4_aruser	Input	Read Address User Signal, channel 3.
s3_axi4_arprot	Input	Read Protection Type, channel 3.
s3_axi4_arready	Output	Read Address Ready, channel 3.
s3_axi4_wdata	Input	Write Data , channel 3.
s3_axi4_wstrb	Input	Write Strobes, channel 3.
s3_axi4_wlast	Input	Write Last, channel 3.
s3_axi4_wvalid	Input	Write Valid, channel 3.
s3_axi4_wready	Output	Write Ready, channel 3.
s3_axi4_bready	Input	Write Response Ready, channel 3.
s3_axi4_bid	Output	Write Response ID, channel 3.
s3_axi4_bresp	Output	Write Response , channel 3.
s3_axi4_bvalid	Output	Write Response Valid, channel 3.
s3_axi4_rready	Input	Read Ready, channel 3.
s3_axi4_rdata	Output	Read Data, channel 3.
s3_axi4_rid	Output	Read ID , channel 3.
s3_axi4_rlast	Output	Read Last, channel 3.
s3_axi4_rresp	Output	Read Response, channel 3.
s3_axi4_rvalid	Output	Read Valid, channel 3.

4.4.11. s0_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Clock for sideband interface (primary I/O bank).

Table 102. Interface: s0_axi4lite_clock

Interface type: clock

Port Name	Direction	Description
s0_axi4lite_clock	Input	Axi-Lite clock, to primary IOSSM.

4.4.12. s0_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Reset for sideband interface (primary I/O bank).

Table 103. Interface: s0_axi4lite_reset_n

Interface type: reset

Port Name	Direction	Description
s0_axi4lite_reset_n	Input	Axi-Lite reset_n, to primary IOSSM.

4.4.13. s0_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Sideband interface (primary I/O bank) that will connect to the IOSSM, through a gearbox in the core.

Table 104. Interface: s0_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s0_axi4lite_awaddr	Input	Axi-Lite Write Address, to primary IOSSM.
s0_axi4lite_awprot	Input	Axi-Lite Write Address Protection Signal, to primary IOSSM.
s0_axi4lite_awvalid	Input	Axi-Lite Write Address Valid, to primary IOSSM.
s0_axi4lite_awready	Output	Axi-Lite Write Address Ready, to primary IOSSM.
s0_axi4lite_araddr	Input	Axi-Lite Read Address, to primary IOSSM.
s0_axi4lite_arprot	Input	Axi-Lite Read Address Protection Signal, to primary IOSSM.
s0_axi4lite_arvalid	Input	Axi-Lite Read Address Valid, to primary IOSSM.
s0_axi4lite_arready	Output	Axi-Lite Read Address Ready, to primary IOSSM.
s0_axi4lite_wdata	Input	Axi-Lite Write Data, to primary IOSSM.
s0_axi4lite_wstrb	Input	Axi-Lite Write Strobe, to primary IOSSM.
s0_axi4lite_wvalid	Input	Axi-Lite Write Valid, to primary IOSSM.
s0_axi4lite_wready	Output	Axi-Lite Write Ready, to primary IOSSM.

continued...

Port Name	Direction	Description
s0_axi4lite_bready	Input	Axi-Lite Write Response Ready, to primary IOSSM.
s0_axi4lite_bresp	Output	Axi-Lite Write Response, to primary IOSSM.
s0_axi4lite_bvalid	Output	Axi-Lite Write Response Valid, to primary IOSSM.
s0_axi4lite_rready	Input	Axi-Lite Read Ready, to primary IOSSM.
s0_axi4lite_rdata	Output	Axi-Lite Read Data, to primary IOSSM.
s0_axi4lite_rresp	Output	Axi-Lite Read Response, to primary IOSSM.
s0_axi4lite_rvalid	Output	Axi-Lite Read Valid, to primary IOSSM.

4.4.14. s1_axi4lite_clock for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Clock for sideband interface (secondary I/O bank).

Table 105. Interface: s1_axi4lite_clock

Interface type: clock

Port Name	Direction	Description
s1_axi4lite_clock	Input	Axi-Lite clock, to secondary IOSSM.

4.4.15. s1_axi4lite_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Reset for sideband interface (secondary I/O bank).

Table 106. Interface: s1_axi4lite_reset_n

Interface type: reset

Port Name	Direction	Description
s1_axi4lite_reset_n	Input	Axi-Lite reset_n, to secondary IOSSM.

4.4.16. s1_axi4lite for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Sideband interface (secondary I/O bank) that will connect to the IOSSM, through a gearbox in the core.

Table 107. Interface: s1_axi4lite

Interface type: axi4lite

Port Name	Direction	Description
s1_axi4lite_awaddr	Input	Axi-Lite Write Address, to secondary IOSSM.
s1_axi4lite_awprot	Input	Axi-Lite Write Address Protection Signal, to secondary IOSSM.
s1_axi4lite_awvalid	Input	Axi-Lite Write Address Valid, to secondary IOSSM.
<i>continued...</i>		

Port Name	Direction	Description
s1_axi4lite_awready	Output	Axi-Lite Write Address Ready, to secondary IOSSM.
s1_axi4lite_araddr	Input	Axi-Lite Read Address, to secondary IOSSM.
s1_axi4lite_arprot	Input	Axi-Lite Read Address Protection Signal, to secondary IOSSM.
s1_axi4lite_arvalid	Input	Axi-Lite Read Address Valid, to secondary IOSSM.
s1_axi4lite_arready	Output	Axi-Lite Read Address Ready, to secondary IOSSM.
s1_axi4lite_wdata	Input	Axi-Lite Write Data, to secondary IOSSM.
s1_axi4lite_wstrb	Input	Axi-Lite Write Strobe, to secondary IOSSM.
s1_axi4lite_wvalid	Input	Axi-Lite Write Valid, to secondary IOSSM.
s1_axi4lite_wready	Output	Axi-Lite Write Ready, to secondary IOSSM.
s1_axi4lite_bready	Input	Axi-Lite Write Response Ready, to secondary IOSSM.
s1_axi4lite_bresp	Output	Axi-Lite Write Response, to secondary IOSSM.
s1_axi4lite_bvalid	Output	Axi-Lite Write Response Valid, to secondary IOSSM.
s1_axi4lite_rready	Input	Axi-Lite Read Ready, to secondary IOSSM.
s1_axi4lite_rdata	Output	Axi-Lite Read Data, to secondary IOSSM.
s1_axi4lite_rresp	Output	Axi-Lite Read Response, to secondary IOSSM.
s1_axi4lite_rvalid	Output	Axi-Lite Read Valid, to secondary IOSSM.

4.4.17. mem_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Interface to the memory (channel 0), including all CA pins, DQ pins, and DQS pins.

Table 108. Interface: mem_0

Interface type: conduit

Port Name	Direction	Description
mem_0_cs	Output	Chip Select channel 0.
mem_0_ca	Output	Command/Address Bus channel 0.
mem_0_dq	Bidir	Data (read/write) channel 0.
mem_0_rdqs_t	Bidir	Read Data Strobe (true) channel 0.
mem_0_rdqs_c	Bidir	Read Data Strobe (complement) channel 0.
mem_0_dmi	Bidir	Data Mask/Data Inversion channel 0.
mem_0_wck_t	Output	Write Clock (true) channel 0.
mem_0_wck_c	Output	Write Clock (complement) channel 0.

4.4.18. mem_ck_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Clock pin to the memory (channel 0).

Table 109. Interface: mem_ck_0

Interface type: conduit

Port Name	Direction	Description
mem_0_ck_t	Output	CK Clock (true) channel 0.
mem_0_ck_c	Output	CK Clock (complement) channel 0.

4.4.19. mem_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Interface to the memory (channel 1), including all CA pins, DQ pins, and DQS pins.

Table 110. Interface: mem_1

Interface type: conduit

Port Name	Direction	Description
mem_1_cs	Output	Chip Select channel 1.
mem_1_ca	Output	Command/Address Bus channel 1.
mem_1_dq	Bidir	Data (read/write) channel 1.
mem_1_rdqs_t	Bidir	Read Data Strobe (true) channel 1.
mem_1_rdqs_c	Bidir	Read Data Strobe (complement) channel 1.
mem_1_dmi	Bidir	Data Mask/Data Inversion channel 1.
mem_1_wck_t	Output	Write Clock (true) channel 1.
mem_1_wck_c	Output	Write Clock (complement) channel 1.

4.4.20. mem_ck_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Clock pin to the memory (channel 1).

Table 111. Interface: mem_ck_1

Interface type: conduit

Port Name	Direction	Description
mem_1_ck_t	Output	CK Clock (true) channel 1.
mem_1_ck_c	Output	CK Clock (complement) channel 1.

4.4.21. mem_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Interface to the memory (channel 2), including all CA pins, DQ pins, and DQS pins.

Table 112. Interface: mem_2

Interface type: conduit

Port Name	Direction	Description
mem_2_cs	Output	Chip Select channel 2.
mem_2_ca	Output	Command/Address Bus channel 2.
mem_2_dq	Bidir	Data (read/write) channel 2.
mem_2_rdqs_t	Bidir	Read Data Strobe (true) channel 2.
mem_2_rdqs_c	Bidir	Read Data Strobe (complement) channel 2.
mem_2_dmi	Bidir	Data Mask/Data Inversion channel 2.
mem_2_wck_t	Output	Write Clock (true) channel 2.
mem_2_wck_c	Output	Write Clock (complement) channel 2.

4.4.22. mem_ck_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Clock pin to the memory (channel 2).

Table 113. Interface: mem_ck_2

Interface type: conduit

Port Name	Direction	Description
mem_2_ck_t	Output	CK Clock (true) channel 2.
mem_2_ck_c	Output	CK Clock (complement) channel 2.

4.4.23. mem_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Interface to the memory (channel 3), including all CA pins, DQ pins, and DQS pins.

Table 114. Interface: mem_3

Interface type: conduit

Port Name	Direction	Description
mem_3_cs	Output	Chip Select channel 3.
mem_3_ca	Output	Command/Address Bus channel 3.
mem_3_dq	Bidir	Data (read/write) channel 3.
mem_3_rdqs_t	Bidir	Read Data Strobe (true) channel 3.
mem_3_rdqs_c	Bidir	Read Data Strobe (complement) channel 3.
mem_3_dmi	Bidir	Data Mask/Data Inversion channel 3.
mem_3_wck_t	Output	Write Clock (true) channel 3.
mem_3_wck_c	Output	Write Clock (complement) channel 3.

4.4.24. mem_ck_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Clock pin to the memory (channel 3).

Table 115. Interface: mem_ck_3

Interface type: conduit

Port Name	Direction	Description
mem_3_ck_t	Output	CK Clock (true) channel 3.
mem_3_ck_c	Output	CK Clock (complement) channel 3.

4.4.25. mem_reset_n for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Reset pin to the memory. Must always be placed along with channel 0, but shared for entire interface (all channels within one EMIF).

Table 116. Interface: mem_reset_n

Interface type: conduit

Port Name	Direction	Description
mem_0_reset_n	Output	Asynchronous Reset channel 0.

4.4.26. oct_0 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

On-Chip Termination (OCT) interface, representing RZQ pin (channel 0).

Table 117. Interface: oct_0

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_0	Input	Calibrated On-Chip Termination (OCT) input pin channel 0.

4.4.27. oct_1 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

On-Chip Termination (OCT) interface, representing RZQ pin (channel 1).

Table 118. Interface: oct_1

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_1	Input	Calibrated On-Chip Termination (OCT) input pin channel 1.

4.4.28. oct_2 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

On-Chip Termination (OCT) interface, representing RZQ pin (channel 2).

Table 119. Interface: oct_2

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_2	Input	Calibrated On-Chip Termination (OCT) input pin channel 2.

4.4.29. oct_3 for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

On-Chip Termination (OCT) interface, representing RZQ pin (channel 3).

Table 120. Interface: oct_3

Interface type: conduit

Port Name	Direction	Description
oct_rzqin_3	Input	Calibrated On-Chip Termination (OCT) input pin channel 3.

4.4.30. ref_clk for Agilex 5 E-Series External Memory Interfaces (EMIF) IP - LPDDR5

Reference clock used by the EMIF PLL.

Table 121. Interface: ref_clk

Interface type: clock

Port Name	Direction	Description
ref_clk	Input	PLL reference clock input.

5. Agilix 5 FPGA EMIF IP – Simulating Memory IP

To simulate your design you require the following components:

- A simulator—The simulator must be an Altera-supported Verilog HDL simulator:
 - Siemens EDA* ModelSim
 - Synopsys* VCS/VCS-MX
- A design using Altera'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 Altera 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 Agilix 5 FPGA IP Design Example User Guide*.

Memory Simulation Models

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

- Altera-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: Altera 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\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.

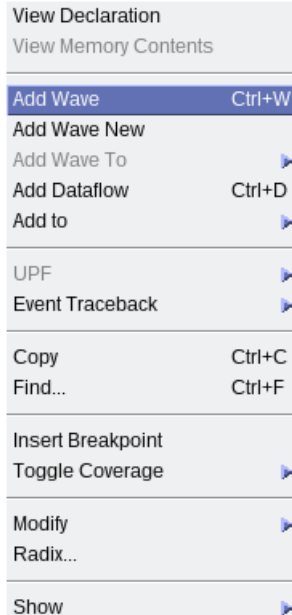
3. Run the following command in the terminal window:

```
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 VCSMX Simulator

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

```
<example_design_directory>/sim/ed_sim/synopsys/vcsmx
```

Non-interactive Mode

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

1. Type the following command on a single line:

```
sh vcsmx_setup.sh USER_DEFINED_COMPILE_OPTIONS=" "
USER_DEFINED_ELAB_OPTIONS="-xlrn\ 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 `vcsmx` folder.

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

```
ELAB_OPTIONS="$ELAB_OPTIONS $QUARTUS_INSTALL_DIR/eda/sim_lib/simsv_dpi.cpp"

run_vcs="vcs -lca -timescale=1ps/1ps -sverilog +verilog2001ext+.v -debug_access+r $ELAB_OPTIONS $USER_DEFINED_ELAB_OPTIONS \
-v $QUARTUS_INSTALL_DIR/eda/sim_lib/220model.v \
-v $QUARTUS_INSTALL_DIR/eda/sim_lib/sgate.v \
-v $QUARTUS_INSTALL_DIR/eda/sim_lib/altera_primitives.v \
-v $QUARTUS_INSTALL_DIR/eda/sim_lib/altera_mf.v \
$QUARTUS_INSTALL_DIR/eda/sim_lib/altera_lnsim.v \
$QUARTUS_INSTALL_DIR/eda/sim_lib/tennm_atoms.v \
$QUARTUS_INSTALL_DIR/eda/sim_lib/synopsys/tennm_atoms_ncrypt.v \
$QUARTUS_INSTALL_DIR/eda/sim_lib/fmca_atoms_ncrypt.v \
$common_design_files \
$design_files \
$USER_DEFINED_ELAB_OPTIONS_APPEND \
-top $TOP_LEVEL_NAME"

eval $run_vcs
```

2. Compile the design example by typing:

```
sh vcs_setup.sh USER_DEFINED_ELAB_OPTIONS="-xlrn\ 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. Agilex 5 FPGA EMIF IP - DDR4 Support

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

6.1. External Memory Interfaces (EMIF) IP - DDR4 Component 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.

Table 122. Group: High-level Configuration / Memory Device

Parameter Name	Description
Data DQ Width	Number of DQ pins per memory channel, used for data. Default value is 32 Legal values are: 16, 32, 40 (Identifier: MEM_CHANNEL_DATA_DQ_WIDTH)
ECC DQ Width	Number of additional DQ pins per memory channel, used for out-of-band ECC. If bigger than 0, controller will enable out-of-band ECC. Otherwise, out-of-band ECC will be disabled. Default value is 0 Legal values are: 0, 8 (Identifier: MEM_CHANNEL_ECC_DQ_WIDTH)
Die DQ Width	Number of DQ pins in each die that makes up the interface. For dual-die packages, this is the width of the die, not the width of full the package. Default value is 16 Legal values are: 8, 16 (Identifier: MEM_DIE_DQ_WIDTH)
Die Density	Capacity of each memory die (in Gbits), per channel per die. For dual-die packages, this is the density of each die, not the density of the full package. Default value is 8 Legal values are: 2, 4, 8, 16 (Identifier: MEM_DIE_DENSITY_GBITS)
CS Width	Specifies the total number of CS pins used by each channel. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_CHANNEL_CS_WIDTH)
Memory Speedbin	Specifies the speedbin of the memory device(s) of which the interface consists. Default value is 3200W Legal values are: 1600J, 1600K, 1600L, 1866L, 1866M, 1866N, 2133N, 2133P, 2133R, 2400P, 2400R, 2400T, 2400U, 2666T, 2666U, 2666V, 2666W, 2933V, 2933W, 2933Y, 2933AA, 3200W, 3200AA, 3200AC
<i>continued...</i>	

© Altera Corporation. Altera, the Altera logo, the 'a' logo, and other Altera marks are trademarks of Altera Corporation. Altera and Intel warrant performance of its FPGA and semiconductor products to current specifications in accordance with Altera's or Intel's standard warranty as applicable, but reserves the right to make changes to any products and services at any time without notice. Altera and Intel assume no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera or Intel. Altera and Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

Parameter Name	Description
	(Identifier: MEM_SPEEDBIN)
Use AC Mirroring	Enable command-address mirroring for multi-rank DDR4 interfaces per JEDEC Standard. Default value is false (Identifier: MEM_AC_MIRRORING_EN)
Share CK Pins Between Ranks	Specifies whether all the ranks in the same channel should share one pair of memory interface differential clock. Default value is false (Identifier: MEM_RANKS_SHARE_CK_EN)
Use AC Parity	Specifies whether address-command parity is enabled. If enabled then command latency is increased by the value of parameter "Address-Command Latency Mode". Default value is false (Identifier: MEM_AC_PARITY_EN)
Auto-set Memory Operating Frequency	if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_OPERATING_FREQ_MHZ_AUTOSET_EN)
Memory Operating Frequency	Specifies the frequency at which the memory interface will run. Legal values are: 666.667, 800, 933.333, 1066.667, 1200, 1333.333, 1466.667, 1600 (Identifier: MEM_OPERATING_FREQ_MHZ)

Table 123. Group: High-level Configuration / PHY

Parameter Name	Description
Auto-set PLL Reference Clock Frequency	if true, let IP select max PLL refclk frequency that this configuration can support. If false, user can set custom value for PLL refclk frequency. Default value is true (Identifier: PHY_REFCLK_FREQ_MHZ_AUTOSET_EN)
Enable Advanced List of PLL Reference Clock Frequencies	If true, provide extended list of possible refclk values. Otherwise, prune possible list of refclk values to a more reasonable length. Default value is false (Identifier: PHY_REFCLK_ADVANCED_SELECT_EN)
Reference Clock Frequency	Specifies the reference clock frequency for the EMIF IOPLL. (Identifier: PHY_REFCLK_FREQ_MHZ)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms I/O BANK and TOP vs BOT part of the I/O BANK). Legal ranges are derived from device floorplan. Default value is BOT Legal values are: BOT, TOP (Identifier: PHY_AC_PLACEMENT)
Alert_n AC-Lane Index	Specifies the AC lane index in which to place the ALERT_N pin. Default value is AC2 Legal values are: AC2, AC3 (Identifier: PHY_ALERT_N_PLACEMENT)
Force Using 4 AC Lanes	Specifies if the minimum number of AC lanes for the memory interface should be forced to 4.
<i>continued...</i>	

Parameter Name	Description
	Default value is false (Identifier: PHY_FORCE_MIN_4_AC_LANES_EN)
Auto-set Mainband Access Mode	if true, let IP select most likely usecase for the PHY_MAINBAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_MAINBAND_ACCESS_MODE_AUTOSSET_EN)
Mainband Access Mode	Specifies the path through which the EMIF QHIP mainband interface is exposed to the user. The mainband interface is the AXI4 interface to the memory controller. Legal values are: NOC, ASYNC, SYNC (Identifier: PHY_MAINBAND_ACCESS_MODE)
Auto-set Sideband Access Mode	if true, let IP select most likely usecase for the PHY_SIDEHAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_SIDEHAND_ACCESS_MODE_AUTOSSET_EN)
Sideband Access Mode	Specifies the path through which the EMIF QHIP sideband interface is exposed to the user. The sideband interface is the AXI4-Lite interface to the IOSSM. Legal values are: NOC, FABRIC (Identifier: PHY_SIDEHAND_ACCESS_MODE)
Pin Swizzle Map	Specifies the swizzle map for the data lanes and pins. (Identifier: PHY_SWIZZLE_MAP)
Use Debug Toolkit	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. Default value is false (Identifier: DEBUG_TOOLS_EN)
Instance ID	Instance ID of the EMIF IP. This is useful when using a discovery mechanism over the side-band interface, to identify which EMIF instance's mailbox is at which offset. If expecting to use a discovery mechanism in hardware, this parameter must be set uniquely for all EMIFs that share a sideband. Otherwise, this parameter can be ignored / kept at the default value. Default value is 0 Legal values are: from 0 to 6 (Identifier: INSTANCE_ID)

Table 124. Group: High-level Configuration / Controller

Parameter Name	Description
Use ECC Autocorrection	If ECC is enabled, specifies whether single-bit-errors (SBEs) should be corrected or just reported. Default value is true (Identifier: CTRL_ECC_AUTOCORRECT_EN)
Use Data Masking	Specifies whether Data Masking is enabled by the controller. When ECC is enabled, RMWs will occur (to recompute / write ECC), regardless of whether this is enabled. Default value is false
<i>continued...</i>	

Parameter Name	Description
	(Identifier: CTRL_DM_EN)
Use WDBI	Specifies whether write Data-bus-inversion is enabled by the controller. Default value is false (Identifier: CTRL_WR_DBI_EN)
Use RDBI	Specifies whether read Data-bus-inversion is enabled by the controller. Default value is false (Identifier: CTRL_RD_DBI_EN)

Table 125. Group: Advanced: Memory Timing / Overrides / JEDEC_TABLE

Parameter Name	Description
JEDEC Parameter	Name of JEDEC Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: MEM_WR_PREAMBLE_MODE, MEM_RD_PREAMBLE_MODE, MEM_CL_CYC, MEM_CWL_CYC, MEM_TREFI_NS, MEM_TRAS_NS, MEM_TRCD_NS, MEM_TRP_NS, MEM_TRC_NS, MEM_TCCD_L_NS, MEM_TCCD_S_NS, MEM_TRRD_L_NS, MEM_TRRD_S_NS, MEM_TFAW_NS, MEM_TWTR_L_NS, MEM_TWTR_S_NS, MEM_TWR_NS, MEM_TMRD_NS, MEM_TCKSRE_NS, MEM_TCKSRX_NS, MEM_TCKE_NS, MEM_TCKESR_CYC, MEM_TMPRR_NS, MEM_TRFC_NS, MEM_TDQCK_NS, MEM_TDQSS_CYC, MEM_TDSH_NS, MEM_TDSS_NS, MEM_TIH_NS, MEM_TIS_NS, MEM_TQSH_NS, MEM_TWLH_NS, MEM_TWLS_NS, MEM_TRFC_DLR_NS, MEM_TRRD_DLR_NS, MEM_TFAW_DLR_NS, MEM_TCCD_DLR_NS, MEM_TXP_NS, MEM_TXS_NS, MEM_TXS_DLL_NS, MEM_TCPDED_NS, MEM_TMOD_NS, MEM_TZQCS_NS, MEM_TZQINIT_CYC, MEM_TZQOPER_CYC (Identifier: JEDEC_OVERRIDE_TABLE_PARAM_NAME)

Table 126. Group: Advanced: Memory Timing / Values

Parameter Name	Description
Write Preamble Length	Specifies the write preamble length in cycles. (Identifier: MEM_WR_PREAMBLE_MODE)
Read Preamble Length	Specifies the read preamble length in cycles. (Identifier: MEM_RD_PREAMBLE_MODE)
Read Latency	Read Latency of the memory device in clock cycles. (Identifier: MEM_CL_CYC)
Write Latency	Write Latency in clock cycles. (Identifier: MEM_CWL_CYC)
tREFI	Specifies the average refresh interval in nanoseconds. (Identifier: MEM_TREFI_NS)
tRAS	Specifies the activation-to-precharge command period in nanoseconds. (Identifier: MEM_TRAS_NS)
tRCD	Specifies the activation to interval read or write delay interval in nanoseconds. (Identifier: MEM_TRCD_NS)
tRP	Specifies the precharge command period in nanoseconds. (Identifier: MEM_TRP_NS)
<i>continued...</i>	

Parameter Name	Description
tRC	Specifies the activate-to-activate or activate-to-refresh command period in nanoseconds. (Identifier: MEM_TRC_NS)
tCCD_L	Specifies the CAS-to-CAS command delay for the same bank group in nanoseconds. (Identifier: MEM_TCCD_L_NS)
tCCD_S	Specifies the CAS-to-CAS command delay for different bank groups in nanoseconds. (Identifier: MEM_TCCD_S_NS)
tRRD_L	Specifies the activation-to-activation command delay for the same bank group in nanoseconds. (Identifier: MEM_TRRD_L_NS)
tRRD_S	Specifies the activation-to-activation command delay for different bank groups in nanoseconds. (Identifier: MEM_TRRD_S_NS)
tFAW	Specifies the four-activate-window in nanoseconds. (Identifier: MEM_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 nanoseconds. (Identifier: MEM_TWTR_L_NS)
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 nanoseconds. (Identifier: MEM_TWTR_S_NS)
tWR	Specifies the write recovery time in nanoseconds. (Identifier: MEM_TWR_NS)
tMRD	Specifies the mode-register command cycle time in nanoseconds. (Identifier: MEM_TMRD_NS)
tCKSRE	Specifies the amount of time, in nanoseconds, required after self-refresh entry or power-down entry. (Identifier: MEM_TCKSRE_NS)
tCKSRX	Specifies the amount of time, in nanoseconds, required before self-refresh exit, power-down exit, or reset exit. (Identifier: MEM_TCKSRX_NS)
tCKE	Specifies the minimum CKE low pulse width from self-refresh entry to self-refresh exit in nanoseconds. (Identifier: MEM_TCKE_NS)
tCKESR	Specifies the minimum CKE low pulse width from self-refresh entry to self-refresh exit in memory clock cycles. (Identifier: MEM_TCKESR_CYC)
tMPRR	Specifies the multi-purpose register recovery time measured in nanoseconds. (Identifier: MEM_TMPRR_NS)
tRFC	Specifies the refresh-to-activate or refresh-to-refresh command period in nanoseconds. (Identifier: MEM_TRFC_NS)
continued...	

Parameter Name	Description
tDQSCK	Specifies the minimum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in nanoseconds. (Identifier: MEM_TDQSCK_NS)
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). (Identifier: MEM_TDQSS_CYC)
tDSH	Specifies the write DQS hold time, in nanoseconds. This is the time difference between the rising CK edge and the falling edge of DQS, measured as a percentage of tCK. (Identifier: MEM_TDSH_NS)
tDSS	Describes the time, in nanoseconds, between the falling edge of DQS to the rising edge of the next CK transition. (Identifier: MEM_TDSS_NS)
tIH (Base)	Refers to the hold time for the Address/Command bus after the rising edge of CK in nanoseconds. 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"). (Identifier: MEM_TIH_NS)
tIS (Base)	Refers to the setup time for the Address/Command/Control bus to the rising edge of CK in nanoseconds. (Identifier: MEM_TIS_NS)
tQSH	Specifies the write DQS hold time in nanoseconds. This is the time difference between the rising CK edge and the falling edge of DQS, measured as a percentage of tCK. (Identifier: MEM_TQSH_NS)
tWLH	Describes the write leveling hold time in nanoseconds. It is measured from the rising edge of DQS to the rising edge of CK. (Identifier: MEM_TWLH_NS)
tWLS	Describes the write leveling setup time in nanoseconds. It is measured from the rising edge of CK to the rising edge of DQS. (Identifier: MEM_TWLS_NS)
tRFC_DLR	Specifies the refresh cycle time across different logical rank in nanoseconds. Only applicable to 3DS devices. (Identifier: MEM_TRFC_DLR_NS)
tRRD_DLR	Specifies the activation-to-activation time across different logical rank in nanoseconds. Only applicable to 3DS devices. (Identifier: MEM_TRRD_DLR_NS)
tFAW_DLR	Specifies the four-activate-window across different logical ranks in nanoseconds. (Identifier: MEM_TFAW_DLR_NS)
tCCD_DLR	Specifies the CAS-to-CAS delay across different logical ranks in nanoseconds. (Identifier: MEM_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 nanoseconds. (Identifier: MEM_TXP_NS)
<i>continued...</i>	

Parameter Name	Description
tXS	Specifies the delay from self refresh exit to commands not requiring a locked DLL in nanoseconds. (Identifier: MEM_TXS_NS)
tXSDLL	Specifies the delay from self refresh exit to commands requiring a locked DLL in nanoseconds. (Identifier: MEM_TXS_DLL_NS)
tCPDED	Specifies the command pass disable delay measured in nanoseconds. (Identifier: MEM_TCPDED_NS)
tMOD	Specifies the mode register set command update delay in nanoseconds. (Identifier: MEM_TMOD_NS)
tZQCS	Specifies the normal operation short calibration time in nanoseconds. (Identifier: MEM_TZQCS_NS)
tZQINIT	Specifies the power-up and reset calibration time in cycles. (Identifier: MEM_TZQINIT_CYC)
tZQOPER	Specifies the normal operation full calibration time in cycles. (Identifier: MEM_TZQOPER_CYC)

Table 127. Group: Advanced: Analog Overrides / Overrides / ANALOG_TABLE

Parameter Name	Description
Analog Parameter	Name of Analog Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: PHY_TERM_X_R_S_AC_OUTPUT_OHM, PHY_TERM_X_R_S_CK_OUTPUT_OHM, PHY_TERM_X_R_S_DQ_OUTPUT_OHM, PHY_TERM_X_DQ_SLEW_RATE, PHY_TERM_X_R_T_DQ_INPUT_OHM, PHY_TERM_X_DQ_VREF, PHY_TERM_X_R_T_REFCLK_INPUT_OHM, MEM_ODT_DQ_X_TGT_WR, MEM_ODT_DQ_X_NON_TGT_WR, MEM_ODT_DQ_X_NON_TGT_RD, MEM_ODT_DQ_X_RON, MEM_VREF_DQ_X_RANGE, MEM_VREF_DQ_X_VALUE (Identifier: ANALOG_PARAM_DERIVATION_PARAM_NAME)

Table 128. Group: Advanced: Analog Overrides / Values

Parameter Name	Description
AC Drive Strength	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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_AC_OUTPUT_OHM)
CK 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_CK_OUTPUT_OHM)
FPGA DQ 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_DQ_OUTPUT_OHM)
<i>continued...</i>	

Parameter Name	Description
DQ 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. <i>Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.</i> Legal values are: SLOW, MEDIUM, FAST, FASTEST (Identifier: PHY_TERM_X_DQ_SLEW_RATE)
DQ 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. Legal values are: RT_40_OHM_CAL, RT_50_OHM_CAL, RT_60_OHM_CAL (Identifier: PHY_TERM_X_R_T_DQ_INPUT_OHM)
DQ 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. Legal values are: from 0 to 100 (Identifier: PHY_TERM_X_DQ_VREF)
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. Legal values are: RT_OFF, RT_DIFF (Identifier: PHY_TERM_X_R_T_REFCLK_INPUT_OHM)
Target Write Termination	Specifies the target termination to be used during a write. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: MEM_ODT_DQ_X_NON_TGT_RD)
Memory DQ Drive Strength	Specifies the termination to be used when driving read data from memory. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: 7, 5 (Identifier: MEM_ODT_DQ_X_RON)
VrefDQ Range	Specifies which of the memory protocol defined ranges will be used. Legal values are: 1, 2 (Identifier: MEM_VREF_DQ_X_RANGE)
VrefDQ Value	Specifies the initial VrefDQ value to be used. Legal values are: from 60.00 to 92.50, from 45.00 to 75.00 (Identifier: MEM_VREF_DQ_X_VALUE)

Table 129. Group: Example Design / Fileset Types

Parameter 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. Default value is VERILOG Legal values are: VERILOG, VHDL (Identifier: EX_DESIGN_HDL_FORMAT)
Generate Synthesis Fileset	Generate Synthesis Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SYNTH)
Generate Simulation Fileset	Generate Simulation Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SIM)

Table 130. Group: Example Design / User PLL

Parameter Name	Description
Auto-set User PLL Output Clock Frequency	if true, let IP select a reference clock frequency for the user PLL in the example design; if false, let user set a custom value for this parameter. Default value is true (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ_AUTOSSET_EN)
User PLL Output Clock Frequency	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode). Default value is 570 (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ)
User PLL Reference Clock Frequency	PLL reference clock frequency in MHz for PLL supplying the core clock. Default value is 100 (Identifier: EX_DESIGN_USER_PLL_REFCLK_FREQ_MHZ)
NOC Reference Clock Frequency	Reference Clock Frequency for the NOC control IP. Default value is 100 Legal values are: 25, 100, 125 (Identifier: EX_DESIGN_NOC_PLL_REFCLK_FREQ_MHZ)

Table 131. Group: Example Design / Traffic Generator

Parameter Name	Description
Traffic Generator Remote Access	Specifies whether the Traffic Generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled. Default value is JTAG Legal values are: EXPORT, JTAG (Identifier: EX_DESIGN_TG_CSR_ACCESS_MODE)
Traffic Generator Program	Specifies the traffic pattern to be run. Default value is MEDIUM Legal values are: SHORT, MEDIUM, LONG, INFINITE (Identifier: EX_DESIGN_TG_PROGRAM)

Table 132. Group: Example Design / Performance Monitor

Parameter Name	Description
Enable Performance Monitor for Channel 0	If true, example design will include a Performance Monitor instance connected to Channel 0. Default value is false (Identifier: EX_DESIGN_PMON_CH0_EN)

6.2. External Memory Interfaces (EMIF) IP - DDR4 DIMM 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.

Table 133. Group: High-level Configuration / Memory Device

Parameter Name	Description
Data DQ Width	Number of DQ pins per memory channel, used for data. Default value is 64 Legal values are: 64, 72 (Identifier: MEM_CHANNEL_DATA_DQ_WIDTH)
ECC DQ Width	Number of additional DQ pins per memory channel, used for out-of-band ECC. If bigger than 0, controller will enable out-of-band ECC. Otherwise, out-of-band ECC will be disabled. Default value is 0 Legal values are: 0, 8 (Identifier: MEM_CHANNEL_ECC_DQ_WIDTH)
DIMM Type	Specifies the type of DIMM that is used with this interface. Default value is UDIMM Legal values are: UDIMM, SODIMM, RDIMM (Identifier: MEM_DIMM_TYPE)
Die DQ Width	Number of DQ pins in each die that makes up the interface. For dual-die packages, this is the width of the die, not the width of full the package. Default value is 8 Legal values are: 4, 8, 16 (Identifier: MEM_DIE_DQ_WIDTH)
Die Density	Capacity of each memory die (in Gbits), per channel per die. For dual-die packages, this is the density of each die, not the density of the full package. Default value is 8 Legal values are: 2, 4, 8, 16, 32 (Identifier: MEM_DIE_DENSITY_GBITS)
CS Width	Specifies the total number of CS pins used by each channel. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_CHANNEL_CS_WIDTH)
Memory Speedbin	Specifies the speedbin of the memory device(s) of which the interface consists. Default value is 3200W Legal values are: 1600J, 1600K, 1600L, 1866L, 1866M, 1866N, 2133N, 2133P, 2133R, 2400P, 2400R, 2400T, 2400U, 2666T, 2666U, 2666V, 2666W, 2933V, 2933W, 2933Y, 2933AA, 3200W, 3200AA, 3200AC, 1600J-3DS2B, 1600K-3DS2B, 1600L-3DS2B, 1866L-3DS2B, 1866M-3DS2B, 1866N-3DS2B, <i>continued...</i>

Parameter Name	Description
	2133P-3DS2A, 2133P-3DS3A, 2133R-3DS4A, 2400P-3DS3B, 2400P-3DS2A, 2400U-3DS2A, 2400U-3DS4A, 2666T-3DS3A, 2666V-3DS3A, 2666W-3DS4A, 2933W-3DS3A, 2933Y-3DS3A, 2933AA-3DS4A, 3200W-3DS4A, 3200AA-3DS4A, 3200AC-3DS4A (Identifier: MEM_SPEEDBIN)
Use AC Mirroring	Enable command-address mirroring for multi-rank DDR4 interfaces per JEDEC Standard. Default value is false (Identifier: MEM_AC_MIRRORING_EN)
Share CK Pins Between Ranks	Specifies whether all the ranks in the same channel should share one pair of memory interface differential clock. Default value is false (Identifier: MEM_RANKS_SHARE_CK_EN)
Use AC Parity	Specifies whether address-command parity is enabled. If enabled then command latency is increased by the value of parameter "Address-Command Latency Mode". Default value is false (Identifier: MEM_AC_PARITY_EN)
Auto-set Memory Operating Frequency	if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_OPERATING_FREQ_MHZ_AUTOSET_EN)
Memory Operating Frequency	Specifies the frequency at which the memory interface will run. Legal values are: 666.667, 800, 933.333, 1066.667, 1200, 1333.333, 1466.667, 1600 (Identifier: MEM_OPERATING_FREQ_MHZ)

Table 134. Group: High-level Configuration / PHY

Parameter Name	Description
Auto-set PLL Reference Clock Frequency	if true, let IP select max PLL refclk frequency that this configuration can support. If false, user can set custom value for PLL refclk frequency. Default value is true (Identifier: PHY_REFCLK_FREQ_MHZ_AUTOSET_EN)
Enable Advanced List of PLL Reference Clock Frequencies	If true, provide extended list of possible refclk values. Otherwise, prune possible list of refclk values to a more reasonable length. Default value is false (Identifier: PHY_REFCLK_ADVANCED_SELECT_EN)
Reference Clock Frequency	Specifies the reference clock frequency for the EMIF IOPLL. (Identifier: PHY_REFCLK_FREQ_MHZ)
Alert_n AC-Lane Index	Specifies the AC lane index in which to place the ALERT_N pin. Default value is AC2 Legal values are: AC2, AC3 (Identifier: PHY_ALERT_N_PLACEMENT)
Force Using 4 AC Lanes	Specifies if the minimum number of AC lanes for the memory interface should be forced to 4. Default value is false (Identifier: PHY_FORCE_MIN_4_AC_LANES_EN)
<i>continued...</i>	

Parameter Name	Description
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms I/O BANK and TOP vs BOT part of the I/O BANK). Legal ranges are derived from device floorplan. Default value is BOT_BOT Legal values are: BOT_BOT, TOP_TOP, BOT_TOP, TOP_TOP_M, TOP_BOT_M, BOT_BOT_M (Identifier: PHY_AC_PLACEMENT)
Mainband Access Mode	Specifies the path through which the EMIF QHIP mainband interface is exposed to the user. The mainband interface is the AXI4 interface to the memory controller. Legal values are: SYNC (Identifier: PHY_MAINBAND_ACCESS_MODE)
Auto-set Sideband Access Mode	if true, let IP select most likely usecase for the PHY_SIDEHAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_SIDEHAND_ACCESS_MODE_AUTOSSET_EN)
Sideband Access Mode	Specifies the path through which the EMIF QHIP sideband interface is exposed to the user. The sideband interface is the AXI4-Lite interface to the IOSSM. Legal values are: FABRIC (Identifier: PHY_SIDEHAND_ACCESS_MODE)
Pin Swizzle Map	Specifies the swizzle map for the data lanes and pins. (Identifier: PHY_SWIZZLE_MAP)
Use Debug Toolkit	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. Default value is false (Identifier: DEBUG_TOOLS_EN)
Instance ID	Instance ID of the EMIF IP. This is useful when using a discovery mechanism over the side-band interface, to identify which EMIF instance's mailbox is at which offset. If expecting to use a discovery mechanism in hardware, this parameter must be set uniquely for all EMIFs that share a sideband. Otherwise, this parameter can be ignored / kept at the default value. Default value is 0 Legal values are: from 0 to 6 (Identifier: INSTANCE_ID)

Table 135. Group: High-level Configuration / Controller

Parameter Name	Description
Use ECC Autocorrection	If ECC is enabled, specifies whether single-bit-errors (SBEs) should be corrected or just reported. Default value is true (Identifier: CTRL_ECC_AUTOCORRECT_EN)
Use Data Masking	Specifies whether Data Masking is enabled by the controller. When ECC is enabled, RMWs will occur (to recompute / write ECC), regardless of whether this is enabled. Default value is false (Identifier: CTRL_DM_EN)
Use WDBI	Specifies whether write Data-bus-inversion is enabled by the controller.
<i>continued...</i>	

Parameter Name	Description
	Default value is false (Identifier: CTRL_WR_DBI_EN)
Use RDBI	Specifies whether read Data-bus-inversion is enabled by the controller. Default value is false (Identifier: CTRL_RD_DBI_EN)

Table 136. Group: Advanced: Memory Timing / Overrides / JEDEC_TABLE

Parameter Name	Description
JEDEC Parameter	Name of JEDEC Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: MEM_WR_PREAMBLE_MODE, MEM_RD_PREAMBLE_MODE, MEM_CL_CYC, MEM_CWL_CYC, MEM_TREFI_NS, MEM_TRAS_NS, MEM_TRCD_NS, MEM_TRP_NS, MEM_TRC_NS, MEM_TCCD_L_NS, MEM_TCCD_S_NS, MEM_TRRD_L_NS, MEM_TRRD_S_NS, MEM_TFAW_NS, MEM_TWTR_L_NS, MEM_TWTR_S_NS, MEM_TWR_NS, MEM_TMRD_NS, MEM_TCKSRE_NS, MEM_TCKSRX_NS, MEM_TCKE_NS, MEM_TCKESR_CYC, MEM_TMPRR_NS, MEM_TRFC_NS, MEM_TDQSCCK_NS, MEM_TDQSS_CYC, MEM_TDSH_NS, MEM_TDSS_NS, MEM_TIH_NS, MEM_TIS_NS, MEM_TQSH_NS, MEM_TWLH_NS, MEM_TWLS_NS, MEM_TRFC_DLR_NS, MEM_TRRD_DLR_NS, MEM_TFAW_DLR_NS, MEM_TCCD_DLR_NS, MEM_TXP_NS, MEM_TXS_NS, MEM_TXS_DLL_NS, MEM_TCPDED_NS, MEM_TMOD_NS, MEM_TZQCS_NS, MEM_TZQINIT_CYC, MEM_TZQOPER_CYC, MEM_SPD137_RCD_CA_DRV, MEM_SPD138_RCD_CK_DRV (Identifier: JEDEC_OVERRIDE_TABLE_PARAM_NAME)

Table 137. Group: Advanced: Memory Timing / Values

Parameter Name	Description
Write Preamble Length	Specifies the write preamble length in cycles. (Identifier: MEM_WR_PREAMBLE_MODE)
Read Preamble Length	Specifies the read preamble length in cycles. (Identifier: MEM_RD_PREAMBLE_MODE)
Read Latency	Read Latency of the memory device in clock cycles. (Identifier: MEM_CL_CYC)
Write Latency	Write Latency in clock cycles. (Identifier: MEM_CWL_CYC)
tREFI	Specifies the average refresh interval in nanoseconds. (Identifier: MEM_TREFI_NS)
tRAS	Specifies the activation-to-precharge command period in nanoseconds. (Identifier: MEM_TRAS_NS)
tRCD	Specifies the activation to interval read or write delay interval in nanoseconds. (Identifier: MEM_TRCD_NS)
tRP	Specifies the precharge command period in nanoseconds. (Identifier: MEM_TRP_NS)
tRC	Specifies the activate-to-activate or activate-to-refresh command period in nanoseconds.
<i>continued...</i>	

Parameter Name	Description
	(Identifier: MEM_TRC_NS)
tCCD_L	Specifies the CAS-to-CAS command delay for the same bank group in nanoseconds. (Identifier: MEM_TCCD_L_NS)
tCCD_S	Specifies the CAS-to-CAS command delay for different bank groups in nanoseconds. (Identifier: MEM_TCCD_S_NS)
tRRD_L	Specifies the activation-to-activation command delay for the same bank group in nanoseconds. (Identifier: MEM_TRRD_L_NS)
tRRD_S	Specifies the activation-to-activation command delay for different bank groups in nanoseconds. (Identifier: MEM_TRRD_S_NS)
tFAW	Specifies the four-activate-window in nanoseconds. (Identifier: MEM_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 nanoseconds. (Identifier: MEM_TWTR_L_NS)
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 nanoseconds. (Identifier: MEM_TWTR_S_NS)
tWR	Specifies the write recovery time in nanoseconds. (Identifier: MEM_TWR_NS)
tMRD	Specifies the mode-register command cycle time in nanoseconds. (Identifier: MEM_TMRD_NS)
tCKSRE	Specifies the amount of time, in nanoseconds, required after self-refresh entry or power-down entry. (Identifier: MEM_TCKSRE_NS)
tCKSRX	Specifies the amount of time, in nanoseconds, required before self-refresh exit, power-down exit, or reset exit. (Identifier: MEM_TCKSRX_NS)
tCKE	Specifies the minimum CKE low pulse width from self-refresh entry to self-refresh exit in nanoseconds. (Identifier: MEM_TCKE_NS)
tCKESR	Specifies the minimum CKE low pulse width from self-refresh entry to self-refresh exit in memory clock cycles. (Identifier: MEM_TCKESR_CYC)
tMPRR	Specifies the multi-purpose register recovery time measured in nanoseconds. (Identifier: MEM_TMPRR_NS)
tRFC	Specifies the refresh-to-activate or refresh-to-refresh command period in nanoseconds. (Identifier: MEM_TRFC_NS)
<i>continued...</i>	

Parameter Name	Description
tDQSCK	Specifies the minimum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in nanoseconds. (Identifier: MEM_TDQSCK_NS)
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). (Identifier: MEM_TDQSS_CYC)
tDSH	Specifies the write DQS hold time, in nanoseconds. This is the time difference between the rising CK edge and the falling edge of DQS, measured as a percentage of tCK. (Identifier: MEM_TDSH_NS)
tDSS	Describes the time, in nanoseconds, between the falling edge of DQS to the rising edge of the next CK transition. (Identifier: MEM_TDSS_NS)
tIH (Base)	Refers to the hold time for the Address/Command bus after the rising edge of CK in nanoseconds. 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"). (Identifier: MEM_TIH_NS)
tIS (Base)	Refers to the setup time for the Address/Command/Control bus to the rising edge of CK in nanoseconds. (Identifier: MEM_TIS_NS)
tQSH	Specifies the write DQS hold time in nanoseconds. This is the time difference between the rising CK edge and the falling edge of DQS, measured as a percentage of tCK. (Identifier: MEM_TQSH_NS)
tWLH	Describes the write leveling hold time in nanoseconds. It is measured from the rising edge of DQS to the rising edge of CK. (Identifier: MEM_TWLH_NS)
tWLS	Describes the write leveling setup time in nanoseconds. It is measured from the rising edge of CK to the rising edge of DQS. (Identifier: MEM_TWLS_NS)
tRFC_DLR	Specifies the refresh cycle time across different logical rank in nanoseconds. Only applicable to 3DS devices. (Identifier: MEM_TRFC_DLR_NS)
tRRD_DLR	Specifies the activation-to-activation time across different logical rank in nanoseconds. Only applicable to 3DS devices. (Identifier: MEM_TRRD_DLR_NS)
tFAW_DLR	Specifies the four-activate-window across different logical ranks in nanoseconds. Only applicable to 3DS devices. (Identifier: MEM_TFAW_DLR_NS)
tCCD_DLR	Specifies the CAS-to-CAS delay across different logical ranks in nanoseconds. Only applicable to 3DS devices. (Identifier: MEM_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 nanoseconds. (Identifier: MEM_TXP_NS)
continued...	

Parameter Name	Description
tXS	Specifies the delay from self refresh exit to commands not requiring a locked DLL in nanoseconds. (Identifier: MEM_TXS_NS)
tXSDLL	Specifies the delay from self refresh exit to commands requiring a locked DLL in nanoseconds. (Identifier: MEM_TXS_DLL_NS)
tCPDED	Specifies the command pass disable delay measured in nanoseconds. (Identifier: MEM_TCPDED_NS)
tMOD	Specifies the mode register set command update delay in nanoseconds. (Identifier: MEM_TMOD_NS)
tZQCS	Specifies the normal operation short calibration time in nanoseconds. (Identifier: MEM_TZQCS_NS)
tZQINIT	Specifies the power-up and reset calibration time in cycles. (Identifier: MEM_TZQINIT_CYC)
tZQOPER	Specifies the normal operation full calibration time in cycles. (Identifier: MEM_TZQOPER_CYC)
RDIMM Serial Presence Detect (SPD) Byte 137	Specifies the value of SPD Byte 137 as an integer in decimal. (Identifier: MEM_SPD137_RCD_CA_DRV)
RDIMM Serial Presence Detect (SPD) Byte 138	Specifies the value of SPD Byte 138 as an integer in decimal. (Identifier: MEM_SPD138_RCD_CK_DRV)
DDR4 RDIMM RCD FORC0E Value	Specifies the value of DDR4 RDIMM RCD control word FORC0E - Parity, NV Mode Enable, and Alert_N Configuration. Internal use only, and value is default to 13. (Identifier: MEM_RCD_PARITY_LATENCY_CONTROL_WORD)

Table 138. Group: Advanced: Analog Overrides / Overrides / ANALOG_TABLE

Parameter Name	Description
Analog Parameter	Name of Analog Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: PHY_TERM_X_R_S_AC_OUTPUT_OHM, PHY_TERM_X_R_S_CK_OUTPUT_OHM, PHY_TERM_X_R_S_DQ_OUTPUT_OHM, PHY_TERM_X_DQ_SLEW_RATE, PHY_TERM_X_R_T_DQ_INPUT_OHM, PHY_TERM_X_DQ_VREF, PHY_TERM_X_R_T_REFCLK_INPUT_OHM, MEM_ODT_DQ_X_TGT_WR, MEM_ODT_DQ_X_NON_TGT_WR, MEM_ODT_DQ_X_NON_TGT_RD, MEM_ODT_DQ_X_RON, MEM_VREF_DQ_X_RANGE, MEM_VREF_DQ_X_VALUE, MEM_RCD_DCA_IBT, MEM_RCD_DCS_IBT, MEM_RCD_DCKE_IBT, MEM_RCD_DODT_IBT (Identifier: ANALOG_PARAM_DERIVATION_PARAM_NAME)

Table 139. Group: Advanced: Analog Overrides / Values

Parameter Name	Description
AC Drive Strength	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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL

continued...

Parameter Name	Description
	(Identifier: PHY_TERM_X_R_S_AC_OUTPUT_OHM)
CK 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_CK_OUTPUT_OHM)
FPGA DQ 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_DQ_OUTPUT_OHM)
DQ 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. <i>Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.</i> Legal values are: SLOW, MEDIUM, FAST, FASTEST (Identifier: PHY_TERM_X_DQ_SLEW_RATE)
DQ 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. Legal values are: RT_40_OHM_CAL, RT_50_OHM_CAL, RT_60_OHM_CAL (Identifier: PHY_TERM_X_R_T_DQ_INPUT_OHM)
DQ 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. Legal values are: from 0 to 100 (Identifier: PHY_TERM_X_DQ_VREF)
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. Legal values are: RT_OFF, RT_DIFF (Identifier: PHY_TERM_X_R_T_REFCLK_INPUT_OHM)
Target Write Termination	Specifies the target termination to be used during a write. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: MEM_ODT_DQ_X_NON_TGT_RD)
Memory DQ Drive Strength	Specifies the termination to be used when driving read data from memory. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: 7, 5 (Identifier: MEM_ODT_DQ_X_RON)
continued...	

Parameter Name	Description
VrefDQ Range	Specifies which of the memory protocol defined ranges will be used. Legal values are: 1, 2 (Identifier: MEM_VREF_DQ_X_RANGE)
VrefDQ Value	Specifies the initial VrefDQ value to be used. Legal values are: from 60.00 to 92.50, from 45.00 to 75.00 (Identifier: MEM_VREF_DQ_X_VALUE)
RCD CA Input Termination	This parameter allows you to select the input bus termination of RCD CA input bus. See JEDEC specifications on DDR4CD02 for the pins covered. The values map to specific termination strengths: 0 -> "100 Ohm", 1 -> "150 Ohm", 2 -> "300 Ohm", 3 -> "OFF". Legal values are: 0, 1, 2, 3 (Identifier: MEM_RCD_DCA_IBT)
RCD CS Input Termination	This parameter allows you to select the input bus termination of RCD CS input bus. See JEDEC specifications on DDR4CD02 for the pins covered. The values map to specific termination strengths: 0 -> "100 Ohm", 1 -> "150 Ohm", 2 -> "300 Ohm", 3 -> "OFF". Legal values are: 0, 1, 2, 3 (Identifier: MEM_RCD_DCS_IBT)
RCD CKE Input Termination	This parameter allows you to select the input bus termination of CKE input bus. See JEDEC specifications on DDR4CD02 for the pins covered. The values map to specific termination strengths: 0 -> "100 Ohm", 1 -> "150 Ohm", 2 -> "300 Ohm", 3 -> "OFF". Legal values are: 0, 1, 2, 3 (Identifier: MEM_RCD_DCKE_IBT)
RCD ODT Input Termination	This parameter allows you to select the input bus termination of ODT input bus. See JEDEC specifications on DDR4CD02 for the pins covered. The values map to specific termination strengths: 0 -> "100 Ohm", 1 -> "150 Ohm", 2 -> "300 Ohm", 3 -> "OFF". Legal values are: 0, 1, 2, 3 (Identifier: MEM_RCD_DODT_IBT)

Table 140. Group: Example Design / Fileset Types

Parameter 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. Default value is VERILOG Legal values are: VERILOG, VHDL (Identifier: EX_DESIGN_HDL_FORMAT)
Generate Synthesis Fileset	Generate Synthesis Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SYNTH)
Generate Simulation Fileset	Generate Simulation Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SIM)

Table 141. Group: Example Design / User PLL

Parameter Name	Description
Auto-set User PLL Output Clock Frequency	if true, let IP select a reference clock frequency for the user PLL in the example design; if false, let user set a custom value for this parameter. Default value is true (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ_AUTOSET_EN)
User PLL Output Clock Frequency	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode). Default value is 570 (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ)
User PLL Reference Clock Frequency	PLL reference clock frequency in MHz for PLL supplying the core clock. Default value is 100 (Identifier: EX_DESIGN_USER_PLL_REFCLK_FREQ_MHZ)
NOC Reference Clock Frequency	Reference Clock Frequency for the NOC control IP. Default value is 100 Legal values are: 25, 100, 125 (Identifier: EX_DESIGN_NOC_PLL_REFCLK_FREQ_MHZ)

Table 142. Group: Example Design / Traffic Generator

Parameter Name	Description
Traffic Generator Remote Access	Specifies whether the Traffic Generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled. Default value is JTAG Legal values are: EXPORT, JTAG (Identifier: EX_DESIGN_TG_CSR_ACCESS_MODE)
Traffic Generator Program	Specifies the traffic pattern to be run. Default value is MEDIUM Legal values are: SHORT, MEDIUM, LONG, INFINITE (Identifier: EX_DESIGN_TG_PROGRAM)

Table 143. Group: Example Design / Performance Monitor

Parameter Name	Description
Enable Performance Monitor for Channel 0	If true, example design will include a Performance Monitor instance connected to Channel 0. Default value is false (Identifier: EX_DESIGN_PMON_CH0_EN)

6.3. 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.

6.3.1. Agilex 5 FPGA EMIF IP Resources

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

6.3.1.1. 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 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.

6.3.1.2. 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. Agilex 5 devices support only differential I/O standard on dedicated PLL clock input pin for EMIF IP.

Altera 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.

6.3.2. Pin Guidelines for Agilex 5 FPGA EMIF IP

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

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.

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 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).

6.3.2.1. 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.

6.3.2.1.1. Agilex 5 FPGA EMIF IP Interface Pins

All HSIO banks in 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 Altera recommends that you always generate a test project of your desired configuration and confirm that it meets timing requirements.

6.3.2.1.2. 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.

6.3.2.1.3. 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](#).

Table 144. Maximum Number of DDR4 Interfaces

Device	Package	Component Interface	DIMM Interface
A5EA005B/ A5EA007B	B15A	1	—
A5EA005B / A5EA007B	B23B	1	—
A5EG005B / A5EG007B	B18A	—	—
A5EC013B ES / A5ED013B ES	B23A	1	—
A5EC008B / A5EC013A / A5EC013B / A5ED008B / A5ED013A / A5ED013B	B23A	1	—
A5EC008B / A5EC013A / A5EC013B / A5ED008B / A5ED013A / A5ED013B	B32A	2	—
A5EC008B / A5EC013B / A5ED008B / A5ED013B	M16A	2	—
A5EA008B / A5EA013B A5EB008B / A5EB013B A5EE008B / A5EE013B	B23B	2	—
A5EC028A / A5EC028B / A5ED028A / A5ED028B	B23A	1	—
A5EC028A / A5EC028B / A5ED028A / A5ED028B	B32A	2	—
<i>continued...</i>			

Device	Package	Component Interface	DIMM Interface
A5EC028B / A5ED028B	M16A	2	—
A5EA028B / A5EB028B / A5EE028B	B23B	2	—
A5EC065B ES / A5ED065B ES	B23A	1	—
A5EC043A / A5EC043B / A5EC052A / A5EC052B / A5EC065A / A5EC065B / A5ED043A / A5ED043B / A5ED052A / A5ED052B / A5ED065A / A5ED065B	B23A	1	—
A5EC065B ES / A5ED065B ES	B32A	4	—
A5EC043A / A5EC043B / A5EC052A / A5EC052B / A5EC065A / A5EC065B / A5ED043A / A5ED043B / A5ED052A / A5ED052B / A5ED065A / A5ED065B	B32A	4	—
A5DC064A ES / A5DD064A ES	B32B	4	2
A5DC051A / A5DC064A / A5DD051A / A5DD064A	B32B	4	2

Component Interface refers to x16, x16 + ECC, x32 or x32+ ECC which can be implemented within a single IO96B bank. The B15A package can support only one instance of x16 / x16 + ECC DDR4 interface using 3AC lanes.

1 DIMM interface requires two adjacent IO96B banks located on the same edge of the device; this is supported only on D-Series devices.

6.3.3. Pin Placements for Agilex 5 FPGA DDR4 EMIF IP

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

6.3.3.1. Address and Command Pin Placement for DDR4

Table 146. Address and Command Pin Placement for DDR4 IP

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

continued...

Address/Command Lane	Index Within Byte Lane	DDR4		
		Scheme 1	Scheme 1A	Scheme 2
	2			
	1			
	0			
AC2	11	BG[0]	BG[0]	BG[0]
	10	BA[1]	BA[1]	BA[1]
	9	BA[0]	BA[0]	BA[0]
	8	ALERT_N	A[17]	ALERT_N
	7	A[16]	A[16]	A[16]
	6	A[15]	A[15]	A[15]
	5	A[14]	A[14]	A[14]
	4	A[13]	A[13]	A[13]
	3	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]
	10	A[10]	A[10]	A[10]
	9	A[9]	A[9]	A[9]
	8	A[8]	A[8]	A[8]
	7	A[7]	A[7]	A[7]
	6	A[6]	A[6]	A[6]
	5	A[5]	A[5]	A[5]
	4	A[4]	A[4]	A[4]
	3	A[3]	A[3]	A[3]
	2	A[2]	A[2]	A[2]
	1	A[1]	A[1]	A[1]
	0	A[0]	A[0]	A[0]
AC0	11	PAR[0]	PAR[0]	PAR[0]
	10	CS_N[1]	CS_N[1]	CS_N[1]
	9	CK_C[0]	CK_C[0]	CK_C[0]
	8	CK_T[0]	CK_T[0]	CK_T[0]
	7	CKE[1]	CKE[1]	CKE[1]
	6	CKE[0]	CKE[0]	CKE[0]
	5	ODT[1]	ODT[1]	ODT[1]
	4	ODT[0]	ODT[0]	ODT[0]
continued...				

Address/Command Lane	Index Within Byte Lane	DDR4		
		Scheme 1	Scheme 1A	Scheme 2
	3	ACT_N[0]	ACT_N[0]	ACT_N[0]
	2	CS_N[0]	CS_N[0]	CS_N[0]
	1	RESET_N[0]	RESET_N[0]	RESET_N[0]
	0	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.
- The Agilex 5 FPGA EMIF IP for DDR4 does not support 3DS.

6.3.3.2. 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 148. 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
	continued...									

Controller	Address / Command Scheme	Data Width Usage	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
	Scheme 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]
<p>Note: 1. This configuration is not supported on E-Series devices. DDR4 x40 requires both controllers within an I/O bank in a lockstep configuration, and AXI user data.</p> <p>2. GPIO – available for GPIO/PHYLite.</p> <p>3. DQ[ECC] – DQ/DQS group used as ECC.</p> <p>4. ES0 silicon supports address/command pin placement for DDR4 on the bottom sub-bank (BL0-BL3) only.</p>										

Table 149. D-Series Supported DDR4 Mapping for Lockstep Configuration for x64

A/C	A/C Placement Option	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
		Primary								Secondary							
3	AC_PRI_T OP / SEC_DQ_ BOT	DQ[0]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	GPIO	DQ[7]	DQ[6]	DQ[5]	DQ[4]	GPIO	GPIO	GPIO	GPIO
	AC_PRI_B OT / SEC_DQ_ BOT	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	GPIO	DQ[7]	DQ[6]	DQ[5]	DQ[4]	GPIO	GPIO	GPIO	GPIO
	AC_PRI_T OP / SEC_DQ_ TOP(m)	DQ[0]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	GPIO	GPIO	X	X	X	DQ[4]	DQ[5]	DQ[6]	DQ[7]
4	AC_PRI_T OP / SEC_DQ_ BOT(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	DQ[7]	DQ[6]	DQ[5]	DQ[4]	GPIO	GPIO	GPIO	GPIO
	AC_PRI_T OP / SEC_DQ_ TOP(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	GPIO	X	X	X	DQ[4]	DQ[5]	DQ[6]	DQ[7]
	AC_PRI_B OT / SEC_DQ_ TOP(m)	AC3	AC0	AC1	AC2	DQ[0]	DQ[1]	DQ[2]	DQ[3]	GPIO	X	X	X	DQ[4]	DQ[5]	DQ[6]	DQ[7]

Note:

- GPIO = available for GPIO/PHYLite.
- X = not available for GPIO/PHYLite.
- AC_PRI_TOP/SEC_DQ_BOT = AC Primary Top Sub-Bank/Secondary DQ Bot Sub-Bank.
- AC_PRI_BOT/SEC_DQ_BOT = AC Primary Bot Sub-Bank/Secondary DQ Bot Sub-Bank.
- AC_PRI_TOP/SEC_DQ_TOP = AC Primary Top Sub-Bank/Secondary DQ Top Sub-Bank.
- AC_PRI_BOT/SEC_DQ_TOP = AC Primary Bot Sub-Bank/Secondary DQ Top Sub-Bank.

Table 150. D-Series Supported Lockstep configuration for DDR4 x72 or x64 (with ECC)

A/C	A/C Placement Option	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
		Primary								Secondary							
3	AC_PRI_T OP / SEC_DQ_ BOT	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0] _J	DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
	AC_Pri Bot / SEC_DQ_ BOT	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	sDQ[4] _J	DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
	AC_PRI_T OP / SEC_DQ_ TOP(m)	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0] _J	GPIO	X	X	X	DQ[5]	DQ[6]	DQ[7]	DQ[8]
4	AC_PRI_T OP / SEC_DQ_ BOT(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	DQ[8]	DQ[7]	DQ[6]	DQ[5]	X	X	GPIO	sDQ[4] _J
	AC_PRI_T OP / SEC_DQ_ TOP(m)	DQ[3]	DQ[2]	DQ[1]	DQ[0]	AC1	AC2	AC0	AC3	DQ[4]	X	X	X	DQ[5]	DQ[6]	DQ[7]	sDQ[8] _J
	AC_PRI_T OP / SEC_DQ_ TOP(m)	AC3	AC0	AC1	AC2	DQ[0]	DQ[1]	DQ[2]	DQ[3]	DQ[4]	X	X	X	DQ[5]	DQ[6]	DQ[7]	sDQ[8] _J
<ul style="list-style-type: none"> GPIO = available for GPIO/PHYLite. X = not available for GPIO/PHYLite. DQ lane with an s prefix is the lane used for the following: <ul style="list-style-type: none"> WUSER/RUSER in x72 configuration or ECC Lane for x64 + ECC configuration AC_PRI_TOP/SEC_DQ_BOT = AC Primary Top Sub-Bank/Secondary DQ Bot Sub-Bank. AC_PRI_BOT/SEC_DQ_BOT = AC Primary Bot Sub-Bank/Secondary DQ Bot Sub-Bank. AC_PRI_TOP/SEC_DQ_TOP = AC Primary Top Sub-Bank/Secondary DQ Top Sub-Bank. AC_PRI_BOT/SEC_DQ_TOP = AC Primary Bot Sub-Bank/Secondary DQ Top Sub-Bank. 																	

Figure 15. x16 + ECC DDR4, Single Rank using x8 Memory Component

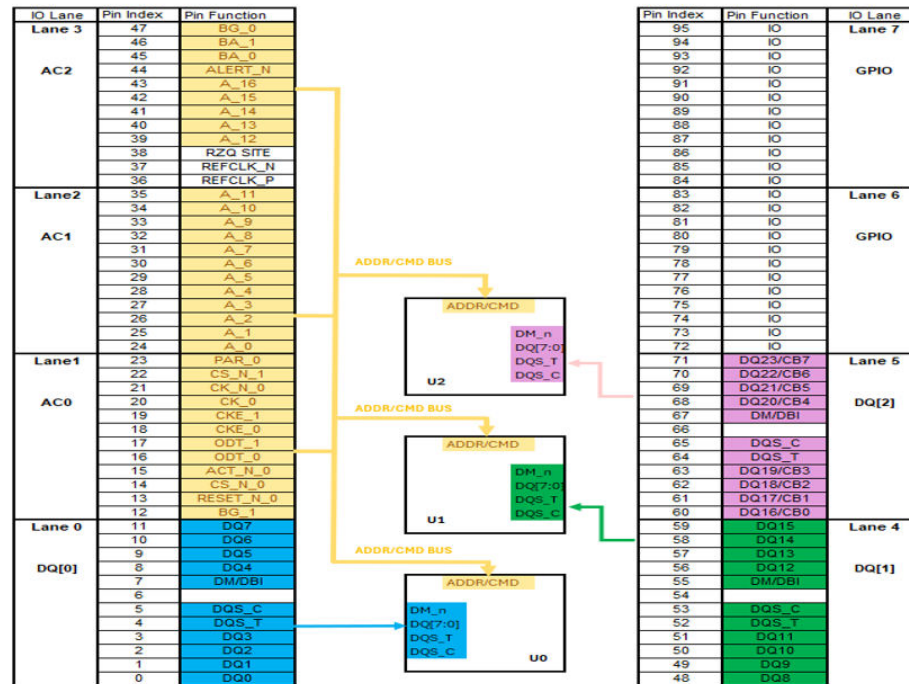


Figure 16. x32 + ECC DDR4, Single Rank using x8 Memory Component

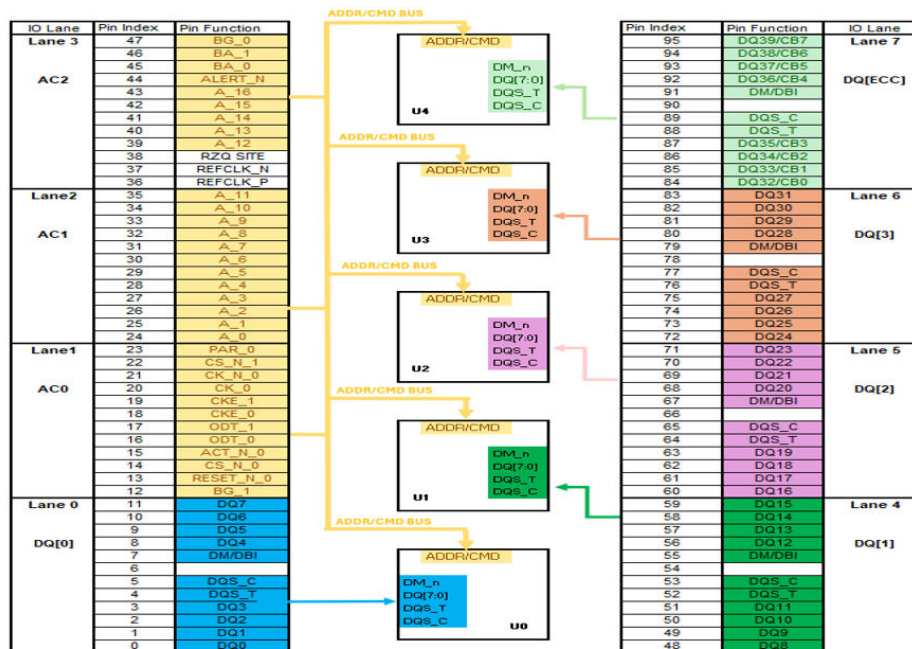
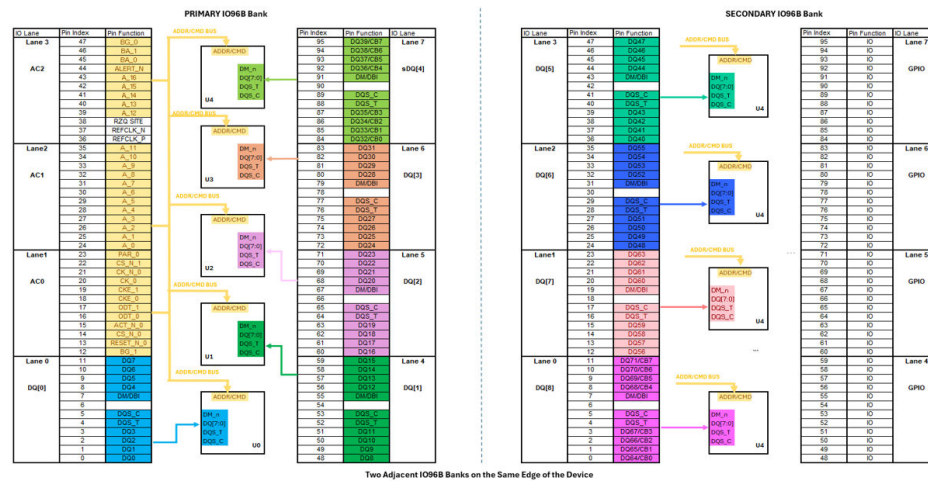


Figure 17. x64 + ECC DDR4 DIMM Interface (3AC - AC_PRI_BOT/SEC_DQ_BOT), Single Rank using x8 Memory Component



6.3.3.3. Clamshell Topology

In a DDR4 clamshell topology, SDRAM is arranged in two layers along either side of the chip, with individual memory devices opposite one another. This configuration allows for a smaller footprint than with fly-by topology, where memory devices are arranged on a single layer.

(Clamshell topology is not supported on the following ES0 devices:
A5EX065BB23AEXSR0, A5EX065BB32AEXSR0, A5EX013BB23AEXSR0.)

The small footprint of the clamshell topology requires less board space than fly-by topology; however, the close proximity of the memory devices in clamshell topology increases the complexity of the required device routing to prevent signal integrity problems. Clamshell topology uses Address Mirroring to minimize undesired effects such as cross-talk, by splitting the chip select signal for each rank, as follows:

- A chip select that accesses the top layer of components, which have not been mirrored.
- A chip select that accesses the bottom layer of components, which have been mirrored.

The total number of chip selects required is double the interface's rank — for example, a single-rank memory interface requires two chip selects. The two chip selects are required for proper calibration of the interface, as a means of accounting for address mirroring. DDR4 component EMIF IP on Agilex 5 devices can support only single rank in clamshell topology.

The JEDEC specification JESD21-C defines address mirroring for DDR4 as shown in the table below:

Table 151. Address Mirroring

Memory Controller Pin	DRAM Pin (Non-mirrored)	DRAM Pin (Mirrored)
A3	A3	A4
A4	A4	A3
A5	A5	A6
A6	A6	A5
A7	A7	A8
A8	A8	A7
A11	A11	A13
A13	A13	A11
BA0	BA0	BA1
BA1	BA1	BA0
BG0 ¹	BG0	BG1
BG1 ¹	BG1	BG0
¹ BG0 and BG1 can be mirrored only when BG1 is present on the memory device.		

Enabling Clamshell Topology in Your External Memory Interface

1. Configure a single memory interface according to your requirements.
2. Enable **Use Clamshell Topology** on the Memory Device section on the High-level Configuration tab.

Figure 18. Enabling Clamshell Topology

External Memory Interfaces (EMIF) IP - DDR4 Component
emif_io96b_ddr4comp

High-level Configuration | Advanced: Memory Timing | Advanced: Analog Overrides | Example Design

Memory Device

Data DQ Width: 32

ECC DQ Width: 8

Die DQ Width: 8

Die Density: 16Gbits

CS Width: 1

Memory Speedbin: 3200AA (CL22)

☒ Use Clamshell Topology Enable this option to use Clamshell Topology

☐ Use AC Parity

☒ Auto-set Memory Operating Frequency

Memory Operating Frequency: 933.333 MHz

Table 152. CS Pin Mapping

Rank	Top/Bottom of Memory Device	CS Pin on Memory Device	CS Pin on FPGA
0	Top	CS0	CS0
0	Bottom	CS0	CS1

6.3.3.4. General Guidelines

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 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 can be used as general purpose I/O pins.
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.

Note: EMIF IP pin-out requirements for the 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.

6.3.3.5. x4 DIMM Implementation

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

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

Table 153. Mapping of DQS Signals Between DIMM and the EMIF IP

DIMM			Quartus Prime EMIF IP	
DQS0_A	DQ[3:0]_A		DQS0	DQ[3:0]_A
DQS5_A	DQ[7:4]_A		DQS1	DQ[7:4]_A
DQS1_A	DQ[11:8]_A		DQS2	DQ[11:8]_A
DQS6_A	DQ[15:12]_A		DQS3	DQ[15:12]_A
DQS2_A	DQ[19:16]_A		DQS4	DQ[19:16]_A
DQS7_A	DQ[23:20]_A		DQS5	DQ[23:20]_A
DQS3_A	DQ[27:24]_A		DQS6	DQ[27:24]_A
DQS8_A	DQ[31:28]_A		DQS7	DQ[31:28]_A
DQS4_A	CB[3:0]_A		DQS8	CB[3:0]_A
DQS9_A	CB[7:4]_A		DQS9	CB[7:4]_A
continued...				

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

Data Bus Connection Mapping Flow

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

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

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

6.3.3.6. 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. Altera recommends that you consider connecting these clock input pins to a reference clock source to facilitate greater system implementation flexibility.

Note: 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.

6.3.3.7. 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.

6.3.3.8. 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 $\pm 25\%$ ($\pm 90^\circ$) of the positive edge of the SDRAM clock input. Therefore, you should generate the CK_T and 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.

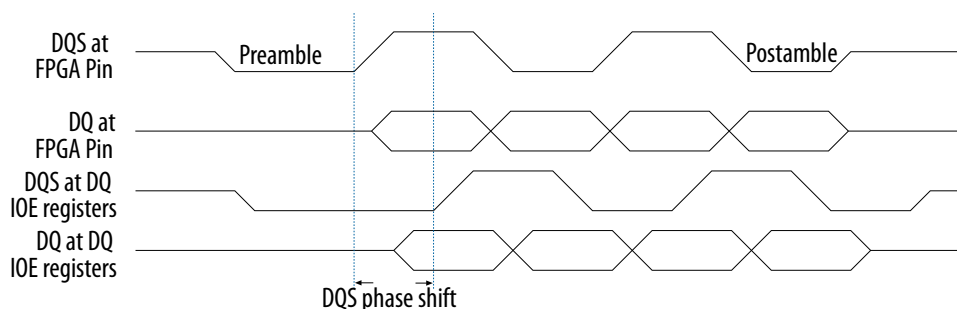
6.3.3.9. Data, Data Strobes, DM/DBI, and Optional ECC Signals

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 $\times 4$ or $\times 8$ mode DQS groups, depending on your chosen memory device or DIMM, regardless of interface width. The $\times 4$ and $\times 8$ 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 $\times 16$ configurations. A group of DQ pins must remain associated with its respective DQS and DQSn pins.

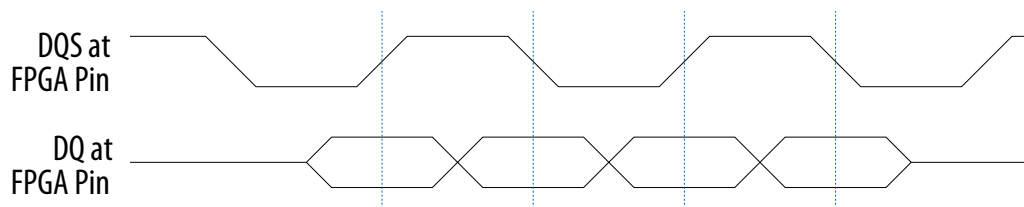
The DQ signals are edge-aligned with the DQS signal during a read from the memory and are center-aligned with the DQS signal during a write to the memory. The memory controller shifts the DQ signals by -90 degrees during a write operation to center align the DQ and DQS signals. The PHY IP delays the DQS signal during a read, so that the DQ and DQS signals are center aligned at the capture register. Altera 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 19. 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 20. 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, Altera 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.

6.4. 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).

6.4.1. DDR4 Byte Lane Swapping

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

The rules for swapping DQ byte lane are as follows:

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

Table 154. Byte Lane Swapping

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

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

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

Example: DDR4 x 40 implemented with AC Scheme 2

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

Table 155. Byte Lane Swapping for Lockstep Configuration

A/C Placem ent Option	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7	BL0	BL1	BL2	BL3	BL4	BL5	BL6	BL7
	Primary								Secondary							
AC_PR I_TOP / SEC_D Q_BOT	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]	DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
AC_PR I_BOT / SEC_D Q_BOT	DQ[0]	AC0	AC1	AC2	DQ[1]	DQ[2]	DQ[3]	sDQ[4]	DQ[8]	DQ[7]	DQ[6]	DQ[5]	GPIO	GPIO	GPIO	GPIO
AC_PR I_TOP / SEC_D Q_TO P(m)	DQ[4]	DQ[3]	DQ[2]	DQ[1]	AC1	AC2	AC0	sDQ[0]	GPIO	X	X	X	DQ[5]	DQ[6]	DQ[7]	DQ[8]

Example: x72 or x64+ECC Configuration with AC_PRI_TOP/SEC_DQ_BOT Placement

In Primary IO96 Bank: sDQ[0] is used for ECC or RUSER/WUSER. It cannot be swapped to other byte lanes.

Byte lane swapping between BL0, 1,2 and 3 is allowed.

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

Example: x72 or x64+ECC Configuration with AC_PRI_BOT/SEC_DQ_BOT Placement

In Primary IO96 Bank: sDQ[4] is used for ECC or RUSER/WUSER. It cannot be swapped to other byte lanes.

Byte lane swapping between BL0, 4,5 and 6 is allowed.

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

Example: x72 or x64+ECC Configuration with AC_PRI_TOP/SEC_DQ_TOP(m)

In Primary IO96 Bank: sDQ[0] is used for ECC or RUSER/WUSER. It cannot be swapped to other byte lanes.

Byte lane swapping between BL0, 1,2 and 3 is allowed.

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

6.4.2. 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.

6.4.3. 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 x8 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 x8 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. Altera 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 x4 or x8 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 156. Pin Swapping Rules for DDR4 x8 Interfaces

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

6.4.4. 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 157. Pin Swapping Rules for DDR4 x4

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

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

Note:

- Although the current version of the Quartus Prime software may not enforce all of the rules listed in the above table, be aware that all of these rules may be enforced in later versions of the software.
- At present, the Quartus Prime software checks the following:
 - Address and command pin placement, per the table in the [Address and Command Pin Placement for DDR4](#) topic.
 - 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].

For guidelines on designing your PCB, refer to the *EMIF PCB Routing Guidelines* section in the *PCB Design Guidelines (HSSI, EMIF, MIPI, True Differential, PDN) User Guide: Agilex 5 FPGAs and SoCs* document.

7. Agilex 5 FPGA EMIF IP - DDR5 Support

This chapter contains IP parameter descriptions and pin planning information for Agilex 5 FPGA external memory interface IP for DDR5.

7.1. External Memory Interfaces (EMIF) IP - DDR5 Component 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.

Table 158. Group: High-level Configuration / Memory Device

Parameter Name	Description
Number of Channels	Specifies the number of channels that the interface should implement. For multi-channel devices, this should always match the number of channels on the device. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_NUM_CHANNELS)
Data DQ Width	Number of DQ pins per memory channel, used for data. Default value is 32 Legal values are: 16, 32, 40 (Identifier: MEM_CHANNEL_DATA_DQ_WIDTH)
ECC DQ Width	Number of additional DQ pins per memory channel, used for out-of-band ECC. If bigger than 0, controller will enable out-of-band ECC. Otherwise, out-of-band ECC will be disabled. Default value is 0 Legal values are: 0, 8 (Identifier: MEM_CHANNEL_ECC_DQ_WIDTH)
Die DQ Width	Number of DQ pins in each die that makes up the interface. For dual-die packages, this is the width of the die, not the width of full the package. Default value is 16 Legal values are: 8, 16 (Identifier: MEM_DIE_DQ_WIDTH)
Die Density	Capacity of each memory die (in Gbits), per channel per die. For dual-die packages, this is the density of each die, not the density of the full package. Default value is 8 Legal values are: 8, 16, 24, 32 (Identifier: MEM_DIE_DENSITY_GBITS)
CS Width	Specifies the total number of CS pins used by each channel. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_CHANNEL_CS_WIDTH)
<i>continued...</i>	

© Altera Corporation. Altera, the Altera logo, the 'a' logo, and other Altera marks are trademarks of Altera Corporation. Altera and Intel warrant performance of its FPGA and semiconductor products to current specifications in accordance with Altera's or Intel's standard warranty as applicable, but reserves the right to make changes to any products and services at any time without notice. Altera and Intel assume no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera or Intel. Altera and Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

Parameter Name	Description
Memory Speedbin	Specifies the speedbin of the memory device(s) of which the interface consists. Default value is 5600AN Legal values are: 3200AN, 3200B, 3200BN, 3200C, 3600AN, 3600B, 3600BN, 3600C, 4000AN, 4000B, 4000BN, 4000C, 4400AN, 4400B, 4400BN, 4400C, 4800AN, 4800B, 4800BN, 4800C, 5200AN, 5200B, 5200BN, 5200C, 5600AN, 5600B, 5600BN, 5600C (Identifier: MEM_SPEEDBIN)
Auto-set Memory Operating Frequency	if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_OPERATING_FREQ_MHZ_AUTOSET_EN)
Memory Operating Frequency	Specifies the frequency at which the memory interface will run. Legal values are: 1600, 1800, 2000, 2200, 2400, 2600, 2800 (Identifier: MEM_OPERATING_FREQ_MHZ)

Table 159. Group: High-level Configuration / PHY

Parameter Name	Description
Auto-set PLL Reference Clock Frequency	if true, let IP select max PLL refclk frequency that this configuration can support. If false, user can set custom value for PLL refclk frequency. Default value is true (Identifier: PHY_REFCLK_FREQ_MHZ_AUTOSET_EN)
Enable Advanced List of PLL Reference Clock Frequencies	If true, provide extended list of possible refclk values. Otherwise, prune possible list of refclk values to a more reasonable length. Default value is false (Identifier: PHY_REFCLK_ADVANCED_SELECT_EN)
Reference Clock Frequency	Specifies the reference clock frequency for the EMIF IOPLL. (Identifier: PHY_REFCLK_FREQ_MHZ)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms I/O BANK and TOP vs BOT part of the I/O BANK). Legal ranges are derived from device floorplan. Default value is BOT Legal values are: BOT, TOP, FULL (Identifier: PHY_AC_PLACEMENT)
Auto-set Mainband Access Mode	if true, let IP select most likely usecase for the PHY_MAINBAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_MAINBAND_ACCESS_MODE_AUTOSET_EN)
Mainband Access Mode	Specifies the path through which the EMIF QHIP mainband interface is exposed to the user. The mainband interface is the AXI4 interface to the memory controller. Legal values are: NOC, ASYNC, SYNC (Identifier: PHY_MAINBAND_ACCESS_MODE)
Auto-set Sideband Access Mode	if true, let IP select most likely usecase for the PHY_SIDEHAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_SIDEHAND_ACCESS_MODE_AUTOSET_EN)
continued...	

Parameter Name	Description
Sideband Access Mode	Specifies the path through which the EMIF QHIP sideband interface is exposed to the user. The sideband interface is the AXI4-Lite interface to the IOSSM. Legal values are: NOC, FABRIC (Identifier: PHY_SIDEHAND_ACCESS_MODE)
Pin Swizzle Map	Specifies the swizzle map for the data lanes and pins. (Identifier: PHY_SWIZZLE_MAP)
Use Debug Toolkit	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. Default value is false (Identifier: DEBUG_TOOLS_EN)
Instance ID	Instance ID of the EMIF IP. This is useful when using a discovery mechanism over the side-band interface, to identify which EMIF instance's mailbox is at which offset. If expecting to use a discovery mechanism in hardware, this parameter must be set uniquely for all EMIFs that share a sideband. Otherwise, this parameter can be ignored / kept at the default value. Default value is 0 Legal values are: from 0 to 6 (Identifier: INSTANCE_ID)

Table 160. Group: High-level Configuration / Controller

Parameter Name	Description
Use ECC Autocorrection	If ECC is enabled, specifies whether single-bit-errors (SBEs) should be corrected or just reported. Default value is true (Identifier: CTRL_ECC_AUTOCORRECT_EN)
Use Data Masking	Specifies whether Data Masking is enabled by the controller. When ECC is enabled, RMWs will occur (to recompute / write ECC), regardless of whether this is enabled. Default value is false (Identifier: CTRL_DM_EN)

Table 161. Group: Advanced: Memory Timing / Overrides / JEDEC_TABLE

Parameter Name	Description
JEDEC Parameter	Name of JEDEC Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: MEM_OPERATING_SPEEDBIN, MEM_CL_CYC, MEM_CWL_CYC, MEM_WR_PREAMBLE_MODE, MEM_RD_PREAMBLE_MODE, MEM_WR_POSTAMBLE_MODE, MEM_RD_POSTAMBLE_MODE, MEM_FINE_GRANULARITY_REFRESH_MODE, MEM_TREFI1_NS, MEM_TREFI2_NS, MEM_TREFISB_NS, MEM_TCCD_S_CYC, MEM_TCCD_L_NS, MEM_TCCD_L_WR_NS, MEM_TCCD_L_WR2_NS, MEM_TRRD_S_CYC, MEM_TRRD_L_NS, MEM_TFAW_NS, MEM_TRFC1_NS, MEM_TRFC2_NS, MEM_TRFCB_NS, MEM_TRCD_NS, MEM_TRP_NS, MEM_TRAS_NS, MEM_TRC_NS, MEM_TREFSBRD_NS, MEM_TWR_NS, MEM_TZQLAT_NS, MEM_TZQCAL_NS, MEM_TMRR_NS, MEM_TMRR_P_NS, MEM_TMRW_NS, MEM_TMRD_NS, MEM_TDFE_NS, MEM_TDCLK_NS, MEM_TWTR_S_NS, MEM_TWTR_L_NS, MEM_TRTP_NS, MEM_TPPD_CYC, MEM_TPD_NS, MEM_TACTPDEN_CYC, MEM_TPRPDEN_CYC, MEM_TREFPDEN_CYC, MEM_TXP_NS, MEM_TCPDED_CYC, MEM_TCSSL_NS,

Parameter Name	Description
	MEM_TCKSRX_NS, MEM_TCSH_SREXIT_NS, MEM_TDQSK_MIN_CYC, MEM_TDQSK_MAX_CYC, MEM_TDQSK_CYC, MEM_TWPRE_EN_CYC, MEM_TDQSS_CYC, MEM_TCKLCS_CYC, MEM_TWTRA_NS (Identifier: JEDEC_OVERRIDE_TABLE_PARAM_NAME)

Table 162. Group: Advanced: Memory Timing / Values

Parameter Name	Description
Operating Speedbin	Specifies the operating speedbin of the memory device(s) for the current operating frequency and device speedbin. (Identifier: MEM_OPERATING_SPEEDBIN)
Read Latency	Read Latency of the memory device in clock cycles. (Identifier: MEM_CL_CYC)
Write Latency	Write Latency in clock cycles. (Identifier: MEM_CWL_CYC)
Write Preamble Mode	Specifies the write preamble mode of the memory interface (0: not supported, 1: 2-cycle preamble, 2: 3-cycle preamble, 3: 4-cycle preamble). (Identifier: MEM_WR_PREAMBLE_MODE)
Read Preamble Mode	Specifies the read preamble mode of the memory interface (0: 1-cycle preamble, 1: 2-cycle preamble, 2: 2-cycle DDR4-style preamble, 3: 3-cycle preamble, 4: 4-cycle preamble). (Identifier: MEM_RD_PREAMBLE_MODE)
Write Postamble Mode	Specifies the write postamble mode of the memory interface (0: 0.5-cycle postamble, 1: 1.5-cycle postamble). (Identifier: MEM_WR_POSTAMBLE_MODE)
Read Postamble Mode	Specifies the read postamble mode of the memory interface (0: 0.5-tCK postamble, 1: 1.5-tCK postamble). (Identifier: MEM_RD_POSTAMBLE_MODE)
Memory Fine Granularity Refresh Mode	Specifies the Fine Granularity Refresh (FGR) mode of the memory interface. (Identifier: MEM_FINE_GRANULARITY_REFRESH_MODE)
tREFI1	Specifies the maximum average refresh interval in normal refresh mode in nanoseconds. (Identifier: MEM_TREFI1_NS)
tREFI2	Specifies the maximum average refresh interval in fine granularity refresh mode in nanoseconds. (Identifier: MEM_TREFI2_NS)
tREFISB	Specifies the maximum average refresh interval in fine granularity and same bank refresh mode in nanoseconds. (Identifier: MEM_TREFISB_NS)
tCCD_S	Specifies the CAS_n to CAS_n command delay for different bank group in cycles. (Identifier: MEM_TCCD_S_CYC)
tCCD_L	Specifies the CAS_n to CAS_n command delay for same bank group in nanoseconds. (Identifier: MEM_TCCD_L_NS)
tCCD_L_WR	Specifies the write CAS_n to write CAS_n command delay for same bank group in nanoseconds.

continued...

Parameter Name	Description
	(Identifier: MEM_TCCD_L_WR_NS)
tCCD_L_WR2	Specifies the write CAS_n to write CAS_n command delay for same bank group and the second write is not RMW, in nanoseconds. (Identifier: MEM_TCCD_L_WR2_NS)
tRRD_S	Specifies the Activate-to-Activate command delay to different bank group for 1KB page size in nanoseconds. (Identifier: MEM_TRRD_S_CYC)
tRRD_L	Specifies the Activate-to-Activate command delay to same bank group for 1KB page size in nanoseconds. (Identifier: MEM_TRRD_L_NS)
tFAW	Specifies the four activate window for 1KB page size in nanoseconds. (Identifier: MEM_TFAW_NS)
tRFC1	Specifies the refresh operation delay in normal refresh mode in nanoseconds. (Identifier: MEM_TRFC1_NS)
tRFC2	Specifies the refresh operation delay in fine granularity refresh mode in nanoseconds. (Identifier: MEM_TRFC2_NS)
tRFCSB	Specifies the refresh operation delay in fine granularity and same bank refresh mode in nanoseconds. (Identifier: MEM_TRFCSB_NS)
tRCD	Specifies the Activate-to-internal-Read-or-Write delay in nanoseconds. (Identifier: MEM_TRCD_NS)
tRP	Specifies the row precharge time in nanoseconds. (Identifier: MEM_TRP_NS)
tRAS	Specifies the Activate-to-Precharge command period in nanoseconds. (Identifier: MEM_TRAS_NS)
tRC (tRAS+tRP)	Specifies the Activate-to-Activate or Refresh command period in nanoseconds. (Identifier: MEM_TRC_NS)
tREFSBRD	Specifies the same bank refresh to activate delay in nanoseconds. (Identifier: MEM_TREFSBRD_NS)
tWR	Specifies the write recovery time in nanoseconds. (Identifier: MEM_TWR_NS)
tZQLAT	Specifies the ZQ calibration latch time in nanoseconds. (Identifier: MEM_TZQLAT_NS)
tZQCAL	Specifies the ZQ calibration time in nanoseconds. (Identifier: MEM_TZQCAL_NS)
tMRR	Specifies the Mode Register Read (MRR) command period in nanoseconds. (Identifier: MEM_TMRR_NS)
tMRR_P	Specifies the Mode Register Read (MRR) pattern to mode register read pattern command spacing in nanoseconds. (Identifier: MEM_TMRR_P_NS)
<i>continued...</i>	

Parameter Name	Description
tMRW	Specifies the Mode Register Write (MRW) command period in nanoseconds. (Identifier: MEM_TMRW_NS)
tMRD	Specifies the Mode Register Set (MRS) command delay in nanoseconds. (Identifier: MEM_TMRD_NS)
tDFE	Specifies the Decision Feedback Equalization (DFE) Mode Register Write update delay time in nanoseconds. (Identifier: MEM_TDFE_NS)
tDLLK	Specifies the timing of DLLK in nanoseconds. (Identifier: MEM_TDLLK_NS)
tWTR_S	Specifies the delay from start of internal write transaction to internal read command for different bank group in nanoseconds. (Identifier: MEM_TWTR_S_NS)
tWTR_L	Specifies the delay from start of internal write transaction to internal read command for same bank group in nanoseconds. (Identifier: MEM_TWTR_L_NS)
tRTP	Specifies the internal read command to precharge command delay in nanoseconds. (Identifier: MEM_TRTP_NS)
tPPD	Specifies the Precharge-to-Precharge delay in cycles. (Identifier: MEM_TPPD_CYC)
tPD	Specifies the minimum power down time in nanoseconds. (Identifier: MEM_TPD_NS)
tACTPDEN	Specifies the timing of Activate command to power down entry command in cycles. (Identifier: MEM_TACTPDEN_CYC)
tPRPDEN	Specifies the timing of Precharge All Banks (PREab), Precharge Same Bank (PREsb), or Normal Precharge (PREpb) to power down entry command in cycles. (Identifier: MEM_TPRPDEN_CYC)
tREFPDEN	Specifies the timing of Refresh All Banks (REFab) or Refresh Same Bank (REFsb) command to power down entry command in cycles. (Identifier: MEM_TREFPDEN_CYC)
tXP	Specifies the exit power down to next valid command in nanoseconds. (Identifier: MEM_TXP_NS)
tCPDED	Specifies the command pass disable delay in nanoseconds. (Identifier: MEM_TCPDED_CYC)
tCSL	Specifies the Self-Refresh CS_n low pulse width in nanoseconds. (Identifier: MEM_TCSL_NS)
tCKSRX	Specifies the valid clock requirement before SRX in nanoseconds. (Identifier: MEM_TCKSRX_NS)
tCSH_SREXIT	Specifies the self-refresh exit CS_n high pulse width in nanoseconds. (Identifier: MEM_TCSH_SREXIT_NS)
tDQCK_MIN	Specifies the minimum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles.
<i>continued...</i>	

Parameter Name	Description
	(Identifier: MEM_TDQSK_MIN_CYC)
tDQSK_MAX	Specifies the maximum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles. (Identifier: MEM_TDQSK_MAX_CYC)
tDQSK	Specifies the DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles. (Identifier: MEM_TDQSK_CYC)
tWPRE_EN	Specifies the write preamble enable window in cycles. The window size depends on the write preamble mode. (Identifier: MEM_TWPRE_EN_CYC)
tDQSS	Specifies the host and system voltage/temperature drift window of first rising DQS_t preamble edge relative to CAS Write Latency (CWL) CK_t-CK_c edge in cycles. (Identifier: MEM_TDQSS_CYC)
tCKLCS	Specifies the valid clock requirement after SRE in cycles. (Identifier: MEM_TCKLCS_CYC)
twTRA	Specifies the delay from start of internal write transaction to internal read with auto precharge command for same bank in nanoseconds. (Identifier: MEM_TWTRA_NS)

Table 163. Group: Advanced: Analog Overrides / Overrides / ANALOG_TABLE

Parameter Name	Description
Analog Parameter	Name of Analog Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: PHY_TERM_X_R_S_AC_OUTPUT_OHM, PHY_TERM_X_R_S_CK_OUTPUT_OHM, PHY_TERM_X_R_S_DQ_OUTPUT_OHM, PHY_TERM_X_DQ_SLEW_RATE, PHY_TERM_X_R_T_DQ_INPUT_OHM, PHY_TERM_X_DQ_VREF, PHY_TERM_X_R_T_REFCLK_INPUT_OHM, PHY_DFE_X_TAP_1, PHY_DFE_X_TAP_2, PHY_DFE_X_TAP_3, PHY_DFE_X_TAP_4, MEM_ODT_DQ_X_TGT_WR, MEM_ODT_DQ_X_NON_TGT_WR, MEM_ODT_DQ_X_NON_TGT_RD, MEM_ODT_DQ_X_IDLE, MEM_ODT_DQ_X_RON, MEM_VREF_DQ_X_VALUE, MEM_ODT_CA_X_CA, MEM_ODT_CA_X_CS, MEM_ODT_CA_X_CK, MEM_VREF_CA_X_CA_VALUE, MEM_VREF_CA_X_CS_VALUE, MEM_DFE_X_TAP_1, MEM_DFE_X_TAP_2, MEM_DFE_X_TAP_3, MEM_DFE_X_TAP_4 (Identifier: ANALOG_PARAM_DERIVATION_PARAM_NAME)

Table 164. Group: Advanced: Analog Overrides / Values

Parameter Name	Description
AC Drive Strength	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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_AC_OUTPUT_OHM)
CK 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_CK_OUTPUT_OHM)
<i>continued...</i>	

Parameter Name	Description
FPGA DQ 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_DQ_OUTPUT_OHM)
DQ 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. <i>Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.</i> Legal values are: SLOW, MEDIUM, FAST, FASTEST (Identifier: PHY_TERM_X_DQ_SLEW_RATE)
DQ 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. Legal values are: RT_40_OHM_CAL, RT_50_OHM_CAL, RT_60_OHM_CAL (Identifier: PHY_TERM_X_R_T_DQ_INPUT_OHM)
DQ 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. Legal values are: from 0 to 100 (Identifier: PHY_TERM_X_DQ_VREF)
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. Legal values are: RT_OFF, RT_DIFF (Identifier: PHY_TERM_X_R_T_REFCLK_INPUT_OHM)
PHY DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_1)
PHY DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_2)
PHY DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_3)
PHY DFE Tap 4	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_4)
Target Write Termination	Specifies the target termination to be used during a write. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$.

continued...

Parameter Name	Description
	Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: MEM_ODT_DQ_X_NON_TGT_RD)
DQ Idle Termination	Specifies the termination to be used for RTT_PARK and DQS_RTT_PARK. For power savings it is recommended to leave this as disabled. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: MEM_ODT_DQ_X_IDLE)
Memory DQ Drive Strength	Specifies the termination to be used when driving read data from memory. Legal values are: 7, 6, 5 (Identifier: MEM_ODT_DQ_X_RON)
VrefDQ Value	Specifies the initial VrefDQ value to be used. Legal values are: from 35.00 to 97.50 (Identifier: MEM_VREF_DQ_X_VALUE)
CA Termination	Specifies the termination to be used for the CA bus. This setting only applies to Group B, Group A will always be unterminated. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 0p5, 1, 2, 3, 4, 6 (Identifier: MEM_ODT_CA_X_CA)
CS Termination	Specifies the termination to be used for the CS bus. This setting only applies to Group B, Group A will always be unterminated. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 0p5, 1, 2, 3, 4, 6 (Identifier: MEM_ODT_CA_X_CS)
CK Termination	Specifies the termination to be used for the CK bus. This setting only applies to Group B, Group A will always be unterminated. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 0p5, 1, 2, 3, 4, 6 (Identifier: MEM_ODT_CA_X_CK)
VrefCA Value	Specifies the initial VrefCA value to be used. Legal values are: from 35.00 to 97.50 (Identifier: MEM_VREF_CA_X_CA_VALUE)
VrefCS Value	Specifies the initial VrefCS value to be used. Legal values are: from 35.00 to 97.50 (Identifier: MEM_VREF_CA_X_CS_VALUE)
MEM DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the memory DFE. (Identifier: MEM_DFE_X_TAP_1)
MEM DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the memory DFE. (Identifier: MEM_DFE_X_TAP_2)
MEM DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the memory DFE. (Identifier: MEM_DFE_X_TAP_3)
MEM DFE Tap 4	This parameter allows you to select the amount of bias used on tap 4 of the memory DFE. (Identifier: MEM_DFE_X_TAP_4)

Table 165. Group: Example Design / Fileset Types

Parameter 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. Default value is VERILOG Legal values are: VERILOG, VHDL (Identifier: EX_DESIGN_HDL_FORMAT)
Generate Synthesis Fileset	Generate Synthesis Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SYNTH)
Generate Simulation Fileset	Generate Simulation Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SIM)

Table 166. Group: Example Design / User PLL

Parameter Name	Description
Auto-set User PLL Output Clock Frequency	if true, let IP select a reference clock frequency for the user PLL in the example design; if false, let user set a custom value for this parameter. Default value is true (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ_AUTOSSET_EN)
User PLL Output Clock Frequency	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode). Default value is 570 (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ)
User PLL Reference Clock Frequency	PLL reference clock frequency in MHz for PLL supplying the core clock. Default value is 100 (Identifier: EX_DESIGN_USER_PLL_REFCLK_FREQ_MHZ)
NOC Reference Clock Frequency	Reference Clock Frequency for the NOC control IP. Default value is 100 Legal values are: 25, 100, 125 (Identifier: EX_DESIGN_NOC_PLL_REFCLK_FREQ_MHZ)

Table 167. Group: Example Design / Traffic Generator

Parameter Name	Description
Traffic Generator Remote Access	Specifies whether the Traffic Generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled. Default value is JTAG Legal values are: EXPORT, JTAG (Identifier: EX_DESIGN_TG_CSR_ACCESS_MODE)
Traffic Generator Program	Specifies the traffic pattern to be run. Default value is MEDIUM Legal values are: SHORT, MEDIUM, LONG, INFINITE (Identifier: EX_DESIGN_TG_PROGRAM)

Table 168. Group: Example Design / Performance Monitor

Parameter Name	Description
Enable Performance Monitor for Channel 0	If true, example design will include a Performance Monitor instance connected to Channel 0. Default value is false (Identifier: EX_DESIGN_PMON_CH0_EN)
Enable Performance Monitor for Channel 1	If true, example design will include a Performance Monitor instance connected to Channel 1. Default value is false (Identifier: EX_DESIGN_PMON_CH1_EN)

7.2. External Memory Interfaces (EMIF) IP - DDR5 DIMM 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.

Table 169. Group: High-level Configuration / Memory Device

Parameter Name	Description
DIMM Type	Specifies the type of DIMM that is used with this interface. Default value is UDIMM Legal values are: UDIMM, SODIMM, RDIMM (Identifier: MEM_DIMM_TYPE)
Data DQ Width	Number of DQ pins per memory channel, used for data. Default value is 32 Legal values are: 32, 40 (Identifier: MEM_CHANNEL_DATA_DQ_WIDTH)
ECC DQ Width	Number of additional DQ pins per memory channel, used for out-of-band ECC. If bigger than 0, controller will enable out-of-band ECC. Otherwise, out-of-band ECC will be disabled. Default value is 4 Legal values are: 0, 4, 8 (Identifier: MEM_CHANNEL_ECC_DQ_WIDTH)
Die DQ Width	Number of DQ pins in each die that makes up the interface. For dual-die packages, this is the width of the die, not the width of full the package. Default value is 8 Legal values are: 4, 8, 16 (Identifier: MEM_DIE_DQ_WIDTH)
Die Density	Capacity of each memory die (in Gbits), per channel per die. For dual-die packages, this is the density of each die, not the density of the full package. Default value is 8 Legal values are: 8, 16, 24, 32 (Identifier: MEM_DIE_DENSITY_GBITS)
CS Width	Specifies the total number of CS pins used by each channel. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_CHANNEL_CS_WIDTH)
Memory Speedbin	Specifies the speedbin of the memory device(s) of which the interface consists.
<i>continued...</i>	

Parameter Name	Description
	<p>Default value is 5600AN</p> <p>Legal values are: 3200AN, 3200B, 3200BN, 3200C, 3600AN, 3600B, 3600BN, 3600C, 4000AN, 4000B, 4000BN, 4000C, 4400AN, 4400B, 4400BN, 4400C, 4800AN, 4800B, 4800BN, 4800C, 5200AN, 5200B, 5200BN, 5200C, 5600AN, 5600B, 5600BN, 5600C, 3DS_3200AN, 3DS_3200B, 3DS_3200BN, 3DS_3200C, 3DS_3600AN, 3DS_3600B, 3DS_3600BN, 3DS_3600C, 3DS_4000AN, 3DS_4000B, 3DS_4000BN, 3DS_4000C, 3DS_4400AN, 3DS_4400B, 3DS_4400BN, 3DS_4400C, 3DS_4800AN, 3DS_4800B, 3DS_4800BN, 3DS_4800C, 3DS_5200AN, 3DS_5200B, 3DS_5200BN, 3DS_5200C, 3DS_5600AN, 3DS_5600B, 3DS_5600BN, 3DS_5600C</p> <p>(Identifier: MEM_SPEEDBIN)</p>
Auto-set Memory Operating Frequency	<p>if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency.</p> <p>Default value is true</p> <p>(Identifier: MEM_OPERATING_FREQ_MHZ_AUTOSSET_EN)</p>
Memory Operating Frequency	<p>Specifies the frequency at which the memory interface will run.</p> <p>Legal values are: 1600, 1800, 2000, 2200, 2400, 2600, 2800</p> <p>(Identifier: MEM_OPERATING_FREQ_MHZ)</p>

Table 170. Group: High-level Configuration / PHY

Parameter Name	Description
Auto-set PLL Reference Clock Frequency	<p>if true, let IP select max PLL refclk frequency that this configuration can support. If false, user can set custom value for PLL refclk frequency.</p> <p>Default value is true</p> <p>(Identifier: PHY_REFCLK_FREQ_MHZ_AUTOSSET_EN)</p>
Enable Advanced List of PLL Reference Clock Frequencies	<p>If true, provide extended list of possible refclk values. Otherwise, prune possible list of refclk values to a more reasonable length.</p> <p>Default value is false</p> <p>(Identifier: PHY_REFCLK_ADVANCED_SELECT_EN)</p>
Reference Clock Frequency	<p>Specifies the reference clock frequency for the EMIF IOPLL.</p> <p>(Identifier: PHY_REFCLK_FREQ_MHZ)</p>
AC Placement	<p>Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms I/O BANK and TOP vs BOT part of the I/O BANK). Legal ranges are derived from device floorplan.</p> <p>Default value is BOT_BOT</p> <p>Legal values are: BOT_TOP, TOP_BOT, BOT_BOT</p> <p>(Identifier: PHY_AC_PLACEMENT)</p>
Auto-set Mainband Access Mode	<p>if true, let IP select most likely usecase for the PHY_MAINBAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode.</p> <p>Default value is true</p> <p>(Identifier: PHY_MAINBAND_ACCESS_MODE_AUTOSSET_EN)</p>
Mainband Access Mode	<p>Specifies the path through which the EMIF QHIP mainband interface is exposed to the user. The mainband interface is the AXI4 interface to the memory controller.</p> <p>Legal values are: NOC, ASYNC, SYNC</p> <p>(Identifier: PHY_MAINBAND_ACCESS_MODE)</p>
Auto-set Sideband Access Mode	<p>if true, let IP select most likely usecase for the PHY_SIDEHAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode.</p>

continued...

Parameter Name	Description
	Default value is true (Identifier: PHY_SIDEHAND_ACCESS_MODE_AUTOSET_EN)
Sideband Access Mode	Specifies the path through which the EMIF QHIP sideband interface is exposed to the user. The sideband interface is the AXI4-Lite interface to the IOSSM. Legal values are: NOC, FABRIC (Identifier: PHY_SIDEHAND_ACCESS_MODE)
Pin Swizzle Map	Specifies the swizzle map for the data lanes and pins. (Identifier: PHY_SWIZZLE_MAP)
Use Debug Toolkit	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. Default value is false (Identifier: DEBUG_TOOLS_EN)
Instance ID	Instance ID of the EMIF IP. This is useful when using a discovery mechanism over the side-band interface, to identify which EMIF instance's mailbox is at which offset. If expecting to use a discovery mechanism in hardware, this parameter must be set uniquely for all EMIFs that share a sideband. Otherwise, this parameter can be ignored / kept at the default value. Default value is 0 Legal values are: from 0 to 6 (Identifier: INSTANCE_ID)

Table 171. Group: High-level Configuration / Controller

Parameter Name	Description
Use ECC Autocorrection	If ECC is enabled, specifies whether single-bit-errors (SBEs) should be corrected or just reported. Default value is true (Identifier: CTRL_ECC_AUTOCORRECT_EN)
Use Data Masking	Specifies whether Data Masking is enabled by the controller. When ECC is enabled, RMWs will occur (to recompute / write ECC), regardless of whether this is enabled. Default value is false (Identifier: CTRL_DM_EN)

Table 172. Group: Advanced: Memory Timing / Overrides / JEDEC_TABLE

Parameter Name	Description
JEDEC Parameter	Name of JEDEC Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: MEM_OPERATING_SPEEDBIN, MEM_CL_CYC, MEM_CWL_CYC, MEM_WR_PREAMBLE_MODE, MEM_RD_PREAMBLE_MODE, MEM_WR_POSTAMBLE_MODE, MEM_RD_POSTAMBLE_MODE, MEM_FINE_GRANULARITY_REFRESH_MODE, MEM_TREFI1_NS, MEM_TREFI2_NS, MEM_TREFISB_NS, MEM_TCCD_S_CYC, MEM_TCCD_L_NS, MEM_TCCD_L_WR_NS, MEM_TCCD_L_WR2_NS, MEM_TRRD_S_CYC, MEM_TRRD_L_NS, MEM_TFAW_NS, MEM_TRFC1_NS, MEM_TRFC2_NS, MEM_TRFCB_NS, MEM_TRCD_NS, MEM_TRP_NS, MEM_TRAS_NS, MEM_TRC_NS, MEM_TREFSBRD_NS, MEM_TWR_NS, MEM_TZQLAT_NS, MEM_TZQCAL_NS, MEM_TMRD_NS, MEM_TMRP_NS, MEM_TMRW_NS, MEM_TMRD_NS, MEM_TDFE_NS, MEM_TDLLK_NS, MEM_TWTR_S_NS, MEM_TWTR_L_NS, MEM_TRTP_NS, MEM_TPPD_CYC,

Parameter Name	Description
	MEM_TPD_NS, MEM_TACTPDEN_CYC, MEM_TPRPDEN_CYC, MEM_TREFPDEN_CYC, MEM_TXP_NS, MEM_TCPDED_CYC, MEM_TCSL_NS, MEM_TCKSRX_NS, MEM_TCSH_SREXIT_NS, MEM_TDQSK_MIN_CYC, MEM_TDQSK_MAX_CYC, MEM_TDQSK_CYC, MEM_TWPRE_EN_CYC, MEM_TDQSS_CYC, MEM_TCKLCS_CYC, MEM_TWTRA_NS, MEM_SPD248_CK_CONTROL_ENABLE, MEM_SPD249_QCA_CS_ENABLE, MEM_SPD250_QCK_SIGNAL_DRIVER_STRENGTH, MEM_SPD252_QCA_QCS_SIGNAL_DRIVER_STRENGTH, MEM_SPD254_CK_CA_CS_SLEW_RATE, MEM_TRRD_DLR_NS, MEM_TFAW_DLR_NS, MEM_TCCD_DLR_NS (Identifier: JEDEC_OVERRIDE_TABLE_PARAM_NAME)

Table 173. Group: Advanced: Memory Timing / Values

Parameter Name	Description
Operating Speedbin	Specifies the operating speedbin of the memory device(s) for the current operating frequency and device speedbin. (Identifier: MEM_OPERATING_SPEEDBIN)
Read Latency	Read Latency of the memory device in clock cycles. (Identifier: MEM_CL_CYC)
Write Latency	Write Latency in clock cycles. (Identifier: MEM_CWL_CYC)
Write Preamble Mode	Specifies the write preamble mode of the memory interface (0: not supported, 1: 2-cycle preamble, 2: 3-cycle preamble, 3: 4-cycle preamble). (Identifier: MEM_WR_PREAMBLE_MODE)
Read Preamble Mode	Specifies the read preamble mode of the memory interface (0: 1-cycle preamble, 1: 2-cycle preamble, 2: 2-cycle DDR4-style preamble, 3: 3-cycle preamble, 4: 4-cycle preamble). (Identifier: MEM_RD_PREAMBLE_MODE)
Write Postamble Mode	Specifies the write postamble mode of the memory interface (0: 0.5-cycle postamble, 1: 1.5-cycle postamble). (Identifier: MEM_WR_POSTAMBLE_MODE)
Read Postamble Mode	Specifies the read postamble mode of the memory interface (0: 0.5-tCK postamble, 1: 1.5-tCK postamble). (Identifier: MEM_RD_POSTAMBLE_MODE)
Memory Fine Granularity Refresh Mode	Specifies the Fine Granularity Refresh (FGR) mode of the memory interface. (Identifier: MEM_FINE_GRANULARITY_REFRESH_MODE)
tREFI1	Specifies the maximum average refresh interval in normal refresh mode in nanoseconds. (Identifier: MEM_TREFI1_NS)
tREFI2	Specifies the maximum average refresh interval in fine granularity refresh mode in nanoseconds. (Identifier: MEM_TREFI2_NS)
tREFISB	Specifies the maximum average refresh interval in fine granularity and same bank refresh mode in nanoseconds. (Identifier: MEM_TREFISB_NS)
tCCD_S	Specifies the CAS_n to CAS_n command delay for different bank group in cycles. (Identifier: MEM_TCCD_S_CYC)
<i>continued...</i>	

Parameter Name	Description
tCCD_L	Specifies the CAS _n to CAS _n command delay for same bank group in nanoseconds. (Identifier: MEM_TCCD_L_NS)
tCCD_L_WR	Specifies the write CAS _n to write CAS _n command delay for same bank group in nanoseconds. (Identifier: MEM_TCCD_L_WR_NS)
tCCD_L_WR2	Specifies the write CAS _n to write CAS _n command delay for same bank group and the second write is not RMW, in nanoseconds. (Identifier: MEM_TCCD_L_WR2_NS)
tRRD_S	Specifies the Activate-to-Activate command delay to different bank group for 1KB page size in nanoseconds. (Identifier: MEM_TRRD_S_CYC)
tRRD_L	Specifies the Activate-to-Activate command delay to same bank group for 1KB page size in nanoseconds. (Identifier: MEM_TRRD_L_NS)
tFAW	Specifies the four activate window for 1KB page size in nanoseconds. (Identifier: MEM_TFAW_NS)
tRFC1	Specifies the refresh operation delay in normal refresh mode in nanoseconds. (Identifier: MEM_TRFC1_NS)
tRFC2	Specifies the refresh operation delay in fine granularity refresh mode in nanoseconds. (Identifier: MEM_TRFC2_NS)
tRFCSB	Specifies the refresh operation delay in fine granularity and same bank refresh mode in nanoseconds. (Identifier: MEM_TRFCSB_NS)
tRCD	Specifies the Activate-to-internal-Read-or-Write delay in nanoseconds. (Identifier: MEM_TRCD_NS)
tRP	Specifies the row precharge time in nanoseconds. (Identifier: MEM_TRP_NS)
tRAS	Specifies the Activate-to-Precharge command period in nanoseconds. (Identifier: MEM_TRAS_NS)
tRC (tRAS+tRP)	Specifies the Activate-to-Activate or Refresh command period in nanoseconds. (Identifier: MEM_TRC_NS)
tREFSBRD	Specifies the same bank refresh to activate delay in nanoseconds. (Identifier: MEM_TREFSBRD_NS)
tWR	Specifies the write recovery time in nanoseconds. (Identifier: MEM_TWR_NS)
tZQLAT	Specifies the ZQ calibration latch time in nanoseconds. (Identifier: MEM_TZQLAT_NS)
tZQCAL	Specifies the ZQ calibration time in nanoseconds. (Identifier: MEM_TZQCAL_NS)
tMRR	Specifies the Mode Register Read (MRR) command period in nanoseconds.
continued...	

Parameter Name	Description
	(Identifier: MEM_TMRR_NS)
tMRR_P	Specifies the Mode Register Read (MRR) pattern to mode register read pattern command spacing in nanoseconds. (Identifier: MEM_TMRR_P_NS)
tMRW	Specifies the Mode Register Write (MRW) command period in nanoseconds. (Identifier: MEM_TMRW_NS)
tMRD	Specifies the Mode Register Set (MRS) command delay in nanoseconds. (Identifier: MEM_TMRD_NS)
tDFE	Specifies the Decision Feedback Equalization (DFE) Mode Register Write update delay time in nanoseconds. (Identifier: MEM_TDFE_NS)
tDLLK	Specifies the timing of DLLK in nanoseconds. (Identifier: MEM_TDLLK_NS)
tWTR_S	Specifies the delay from start of internal write transaction to internal read command for different bank group in nanoseconds. (Identifier: MEM_TWTR_S_NS)
tWTR_L	Specifies the delay from start of internal write transaction to internal read command for same bank group in nanoseconds. (Identifier: MEM_TWTR_L_NS)
tRTP	Specifies the internal read command to precharge command delay in nanoseconds. (Identifier: MEM_TRTP_NS)
tPPD	Specifies the Precharge-to-Precharge delay in cycles. (Identifier: MEM_TPPD_CYC)
tPD	Specifies the minimum power down time in nanoseconds. (Identifier: MEM_TPD_NS)
tACTPDEN	Specifies the timing of Activate command to power down entry command in cycles. (Identifier: MEM_TACTPDEN_CYC)
tPRPDEN	Specifies the timing of Precharge All Banks (PREab), Precharge Same Bank (PREsb), or Normal Precharge (PREpb) to power down entry command in cycles. (Identifier: MEM_TPRPDEN_CYC)
tREFPDEN	Specifies the timing of Refresh All Banks (REFab) or Refresh Same Bank (REFsb) command to power down entry command in cycles. (Identifier: MEM_TREFPDEN_CYC)
tXP	Specifies the exit power down to next valid command in nanoseconds. (Identifier: MEM_TXP_NS)
tCPDED	Specifies the command pass disable delay in nanoseconds. (Identifier: MEM_TCPDED_CYC)
tCSL	Specifies the Self-Refresh CS_n low pulse width in nanoseconds. (Identifier: MEM_TCSL_NS)
tCKSRX	Specifies the valid clock requirement before SRX in nanoseconds. (Identifier: MEM_TCKSRX_NS)
continued...	

Parameter Name	Description
tCSH_SREXIT	Specifies the self-refresh exit CS_n high pulse width in nanoseconds. (Identifier: MEM_TCSH_SREXIT_NS)
tDQSCK_MIN	Specifies the minimum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles. (Identifier: MEM_TDQSCK_MIN_CYC)
tDQSCK_MAX	Specifies the maximum DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles. (Identifier: MEM_TDQSCK_MAX_CYC)
tDQSCK	Specifies the DQS_t, DQS_c rising edge output timing location from rising CK_t, CK_c in cycles. (Identifier: MEM_TDQSCK_CYC)
tWPRE_EN	Specifies the write preamble enable window in cycles. The window size depends on the write preamble mode. (Identifier: MEM_TWPRE_EN_CYC)
tDQSS	Specifies the host and system voltage/temperature drift window of first rising DQS_t preamble edge relative to CAS Write Latency (CWL) CK_t-CK_c edge in cycles. (Identifier: MEM_TDQSS_CYC)
tCKLCS	Specifies the valid clock requirement after SRE in cycles. (Identifier: MEM_TCKLCS_CYC)
tWTRA	Specifies the delay from start of internal write transaction to internal read with auto precharge command for same bank in nanoseconds. (Identifier: MEM_TWTRA_NS)
RDIMM Serial Presence Detect (SPD) Byte 248	Specifies the value of SPD Byte 248 as an integer in decimal. (Identifier: MEM_SPD248_CK_CONTROL_ENABLE)
RDIMM Serial Presence Detect (SPD) Byte 249	Specifies the value of SPD Byte 249 as an integer in decimal. (Identifier: MEM_SPD249_QCA_CS_ENABLE)
RDIMM Serial Presence Detect (SPD) Byte 250	Specifies the value of SPD Byte 250 as an integer in decimal. (Identifier: MEM_SPD250_QCK_SIGNAL_DRIVER_STRENGTH)
RDIMM Serial Presence Detect (SPD) Byte 252	Specifies the value of SPD Byte 252 as an integer in decimal. (Identifier: MEM_SPD252_QCA_QCS_SIGNAL_DRIVER_STRENGTH)
RDIMM Serial Presence Detect (SPD) Byte 254	Specifies the value of SPD Byte 254 as an integer in decimal. (Identifier: MEM_SPD254_CK_CA_CS_SLEW_RATE)
tRRD_DLR	Specifies the Activate-to-Activate command delay to different logical ranks in nanoseconds. Only applicable to 3D stacked devices. (Identifier: MEM_TRRD_DLR_NS)
tFAW_DLR	Specifies the four activate window for different logical ranks in nanoseconds. Only applicable to 3D stacked devices. (Identifier: MEM_TFAW_DLR_NS)
tCCD_DLR	Specifies the write CAS_n to write CAS_n command delay in different logical ranks, in nanoseconds. Only applicable to 3D stacked devices device. (Identifier: MEM_TCCD_DLR_NS)

Table 174. Group: Advanced: Analog Overrides / Overrides / ANALOG_TABLE

Parameter Name	Description
Analog Parameter	<p>Name of Analog Parameter to explicitly override; the values will be applied and appear in the list below.</p> <p>Default value is</p> <p>Legal values are: PHY_TERM_X_R_S_AC_OUTPUT_OHM, PHY_TERM_X_R_S_CK_OUTPUT_OHM, PHY_TERM_X_R_S_DQ_OUTPUT_OHM, PHY_TERM_X_DQ_SLEW_RATE, PHY_TERM_X_R_T_DQ_INPUT_OHM, PHY_TERM_X_DQ_VREF, PHY_TERM_X_R_T_REFCLK_INPUT_OHM, PHY_DFE_X_TAP_1, PHY_DFE_X_TAP_2, PHY_DFE_X_TAP_3, PHY_DFE_X_TAP_4, MEM_ODT_DQ_X_TGT_WR, MEM_ODT_DQ_X_NON_TGT_WR, MEM_ODT_DQ_X_NON_TGT_RD, MEM_ODT_DQ_X_IDLE, MEM_ODT_DQ_X_RON, MEM_VREF_DQ_X_VALUE, MEM_ODT_CA_X_CA, MEM_ODT_CA_X_CS, MEM_ODT_CA_X_CK, MEM_VREF_CA_X_CA_VALUE, MEM_VREF_CA_X_CS_VALUE, MEM_DFE_X_TAP_1, MEM_DFE_X_TAP_2, MEM_DFE_X_TAP_3, MEM_DFE_X_TAP_4, MEM_RCD_DCA_IBT, MEM_RCD_DCS_IBT, MEM_RCD_DCK_IBT, MEM_RCD_DERROR_IBT (Identifier: ANALOG_PARAM_DERIVATION_PARAM_NAME)</p>

Table 175. Group: Advanced: Analog Overrides / Values

Parameter Name	Description
AC Drive Strength	<p>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.</p> <p>Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_AC_OUTPUT_OHM)</p>
CK Drive Strength	<p>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.</p> <p>Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_CK_OUTPUT_OHM)</p>
FPGA DQ Drive Strength	<p>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.</p> <p>Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_DQ_OUTPUT_OHM)</p>
DQ Slew Rate	<p>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. <i>Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.</i></p> <p>Legal values are: SLOW, MEDIUM, FAST, FASTEST (Identifier: PHY_TERM_X_DQ_SLEW_RATE)</p>
DQ Input Termination	<p>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.</p> <p>Legal values are: RT_40_OHM_CAL, RT_50_OHM_CAL, RT_60_OHM_CAL (Identifier: PHY_TERM_X_R_T_DQ_INPUT_OHM)</p>
DQ Initial Vrefin	<p>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.</p> <p>Legal values are: from 0 to 100 (Identifier: PHY_TERM_X_DQ_VREF)</p>
<i>continued...</i>	

Parameter 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. Legal values are: RT_OFF, RT_DIFF (Identifier: PHY_TERM_X_R_T_REFCLK_INPUT_OHM)
PHY DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_1)
PHY DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_2)
PHY DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_3)
PHY DFE Tap 4	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_4)
Target Write Termination	Specifies the target termination to be used during a write. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: 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. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: MEM_ODT_DQ_X_NON_TGT_RD)
DQ Idle Termination	Specifies the termination to be used for RTT_PARK and DQS_RTT_PARK. For power savings it is recommended to leave this as disabled. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6, 7 (Identifier: MEM_ODT_DQ_X_IDLE)
Memory DQ Drive Strength	Specifies the termination to be used when driving read data from memory. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: 7, 6, 5 (Identifier: MEM_ODT_DQ_X_RON)
VrefDQ Value	Specifies the initial VrefDQ value to be used. Legal values are: from 35.00 to 97.50 (Identifier: MEM_VREF_DQ_X_VALUE)
CA Termination	Specifies the termination to be used for the CA bus. This setting only applies to Group B, Group A will always be unterminated. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 0p5, 1, 2, 3, 4, 6 (Identifier: MEM_ODT_CA_X_CA)
continued...	

Parameter Name	Description
CS Termination	Specifies the termination to be used for the CS bus. This setting only applies to Group B, Group A will always be unterminated. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 0p5, 1, 2, 3, 4, 6 (Identifier: MEM_ODT_CA_X_CS)
CK Termination	Specifies the termination to be used for the CK bus. This setting only applies to Group B, Group A will always be unterminated. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 0p5, 1, 2, 3, 4, 6 (Identifier: MEM_ODT_CA_X_CK)
VrefCA Value	Specifies the initial VrefCA value to be used. Legal values are: from 35.00 to 97.50 (Identifier: MEM_VREF_CA_X_CA_VALUE)
VrefCS Value	Specifies the initial VrefCS value to be used. Legal values are: from 35.00 to 97.50 (Identifier: MEM_VREF_CA_X_CS_VALUE)
MEM DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the memory DFE. (Identifier: MEM_DFE_X_TAP_1)
MEM DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the memory DFE. (Identifier: MEM_DFE_X_TAP_2)
MEM DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the memory DFE. (Identifier: MEM_DFE_X_TAP_3)
MEM DFE Tap 4	This parameter allows you to select the amount of bias used on tap 4 of the memory DFE. (Identifier: MEM_DFE_X_TAP_4)
RCD CA Input Termination	This parameter allows you to select the input bus termination of RCD CA input bus. See JEDEC specifications on DDR4CD02 and DDR5RCD04 for the pins covered. The values map to specific termination strengths: 0 -> "60 Ohm", 1 -> "48 Ohm", 3 -> "OFF". Legal values are: 0, 1, 3 (Identifier: MEM_RCD_DCA_IBT)
RCD CS Input Termination	This parameter allows you to select the input bus termination of RCD CS input bus. See JEDEC specifications on DDR4CD02 and DDR5RCD04 for the pins covered. The values map to specific termination strengths: 0 -> "60 Ohm", 1 -> "48 Ohm", 3 -> "OFF". Legal values are: 0, 1, 3 (Identifier: MEM_RCD_DCS_IBT)
RCD CK Input Termination	This parameter allows you to select the input bus termination of CK input bus. See JEDEC specifications on DDR5RCD04 for the pins covered. The values map to specific termination strengths: 0 -> "60 Ohm", 1 -> "48 Ohm", 3 -> "OFF". Legal values are: 0, 1, 3 (Identifier: MEM_RCD_DCK_IBT)
RCD Error Input Termination	This parameter allows you to select the input bus termination of DERROR input bus. See JEDEC specification on DDR5RCD04 for the pins covered. The values map to specific termination strengths: 0 -> "60 Ohm", 1 -> "48 Ohm", 3 -> "OFF".

Parameter Name	Description
	Legal values are: 0, 1, 3 (Identifier: MEM_RCD_DERROR_IBT)

Table 176. Group: Example Design / Fileset Types

Parameter 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. Default value is VERILOG Legal values are: VERILOG, VHDL (Identifier: EX_DESIGN_HDL_FORMAT)
Generate Synthesis Fileset	Generate Synthesis Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SYNTH)
Generate Simulation Fileset	Generate Simulation Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SIM)

Table 177. Group: Example Design / User PLL

Parameter Name	Description
Auto-set User PLL Output Clock Frequency	if true, let IP select a reference clock frequency for the user PLL in the example design; if false, let user set a custom value for this parameter. Default value is true (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ_AUTOSSET_EN)
User PLL Output Clock Frequency	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode). Default value is 570 (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ)
User PLL Reference Clock Frequency	PLL reference clock frequency in MHz for PLL supplying the core clock. Default value is 100 (Identifier: EX_DESIGN_USER_PLL_REFCLK_FREQ_MHZ)
NOC Reference Clock Frequency	Reference Clock Frequency for the NOC control IP. Default value is 100 Legal values are: 25, 100, 125 (Identifier: EX_DESIGN_NOC_PLL_REFCLK_FREQ_MHZ)

Table 178. Group: Example Design / Traffic Generator

Parameter Name	Description
Traffic Generator Remote Access	Specifies whether the Traffic Generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled. Default value is JTAG Legal values are: EXPORT, JTAG (Identifier: EX_DESIGN_TG_CSR_ACCESS_MODE)
Traffic Generator Program	Specifies the traffic pattern to be run. Default value is MEDIUM Legal values are: SHORT, MEDIUM, LONG, INFINITE (Identifier: EX_DESIGN_TG_PROGRAM)

Table 179. Group: Example Design / Performance Monitor

Parameter Name	Description
Enable Performance Monitor for Channel 0	If true, example design will include a Performance Monitor instance connected to Channel 0. Default value is false (Identifier: EX_DESIGN_PMON_CH0_EN)
Enable Performance Monitor for Channel 1	If true, example design will include a Performance Monitor instance connected to Channel 1. Default value is false (Identifier: EX_DESIGN_PMON_CH1_EN)

7.3. 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.

7.3.1. Agilex 5 FPGA EMIF IP Interface Pins

All HSIO banks in 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.

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.

7.3.1.1. Estimating Pin Requirements

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

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

7.3.1.2. DIMM Options

DDR5 unbuffered DIMMs (UDIMMs) and small outline DIMMs (SODIMMs) command and address pins are clocked at single data rate (SDR). DDR5 registered DIMMs (RDIMMs) command and address pins are clocked at double data rate (DDR).

The table below shows a pin comparison of UDIMM, SODIMM, and RDIMM modules up to dual rank. You should always check your memory vendor's data sheet to be sure.

Table 180. UDIMM, SODIMM, and RDIMM Pin Options for DDR5

Pins	UDIMM Pins	SODIMM Pins	RDIMM Pins
Data	72 bit DQ[31:0]_A DQ[31:0]_B CB[3:0]_A CB[3:0]_B	72 bit DQ[31:0]_A DQ[31:0]_B CB[3:0]_A CB[3:0]_B	80 bit DQ[31:0]_A DQ[31:0]_B CB[7:0]_A CB[7:0]_B
Data Mask	DM[3:0]_A_n ⁽¹⁾ DM[3:0]_B_n ⁽¹⁾	DM[3:0]_A_n ⁽¹⁾ DM[3:0]_B_n ⁽¹⁾	DM[4:0]_A_n ⁽¹⁾ DM[4:0]_B_n ⁽¹⁾
Data Strobe	x8: DQS[4:0]_A_t DQS[4:0]_A_c DQS[4:0]_B_t DQS[4:0]_B_c	x8: DQS[4:0]_A_t DQS[4:0]_A_c DQS[4:0]_B_t DQS[4:0]_B_c	x8: DQS[4:0]_A_t DQS[4:0]_A_c DQS[4:0]_B_t DQS[4:0]_B_c x4: DQS[9:0]_A_t DQS[9:0]_A_c DQS[9:0]_B_t DQS[9:0]_B_c
Command / Address	CA[12:0]_A CA[12:0]_B CA[1:0]_A_n CA[1:0]_B_n	CA[12:0]_A CA[12:0]_B CA[1:0]_A_n CA[1:0]_B_n	CA[6:0]_A CA[6:0]_B CS[1:0]_A_n CS[1:0]_B_n
Clock	CK[1:0]_A_t CK[1:0]_A_c	CK[1:0]_A_t CK[1:0]_A_c	CK_t CK_c
continued...			

Pins	UDIMM Pins	SODIMM Pins	RDIMM Pins
	CK[1:0]_B_t CK[1:0]_B_c	CK[1:0]_B_t CK[1:0]_B_c	
Parity	ALERT_n	ALERT_n	ALERT_n PAR_A PAR_B
Other Pins	RESET_n HSDA, HSCL, HSA	RESET_n HSDA, HSCL, HSA	RESET_n HSDA, HSCL, HSA LBD/RSP_A_n LBS/RSP_B_n
Notes to Table: 1. DM pins are available only for DIMMs constructed using x8 or greater components. 2. The Agilex 5 memory controller supports up to 2 ranks per channel. Agilex 5 devices support only 1 DIMM per channel (1DPC).			

7.3.1.3. 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 more information about timing closure, refer to the *Area and Timing Optimization Techniques* chapter in the *Quartus Prime Handbook*.

Table 181. Maximum Number of DDR5 Interfaces

Device	Package	Component Interface	DIMM Interface
A5EC013A / A5ED013A	B23A	1	—
A5EC013A / A5ED013A	B32A	2	—
A5EC028A / A5ED028A	B23A	1	—
A5EC028A / A5ED028A	B32A	2	—
A5EC043A / A5EC052A / A5EC065A / A5ED043A / A5ED052A / A5ED065A	B23A	1	—
A5EC043A / A5EC052A / A5EC065A / A5ED043A / A5ED052A / A5ED065A	B32A	4	—
A5DC064A ES / A5DD064A ES	B32B	4	2
A5DC051A / A5DC064A / A5DD051A / A5DD064A	B32B	4	2

Component Interface refers to 2ch x16, x16, x16 + ECC, x32 and x32 + ECC which can be implemented within a single IO96B bank.

1 DIMM interface requires two adjacent IO96B banks located on the same edge of the device; this is supported only on D-Series devices.

7.3.2. Agilex 5 FPGA EMIF IP Resources

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

7.3.2.1. 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 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 V_{CCIO_PIO} voltage as the memory interface.
- The OCT calibration block uses a single fixed R_{ZQ} . You must ensure that an external termination resistor is connected to the correct pin for a given OCT block.

For specific pin connection requirements, refer to [Specific Pin Connection Requirements](#).

7.3.2.2. PLL

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

For the clock source, use the clock input pin specifically dedicated to the PLL that you want to use with your external memory interface. The input and output pins are only fully compensated when you use the dedicated PLL clock input pin. Agilex 5 devices support only differential I/O standard on dedicated PLL clock input pin for EMIF IP.

Altera 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.

7.3.3. Pin Guidelines for Agilex 5 FPGA EMIF IP

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

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.

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 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 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).

7.3.3.1. General Guidelines - DDR5

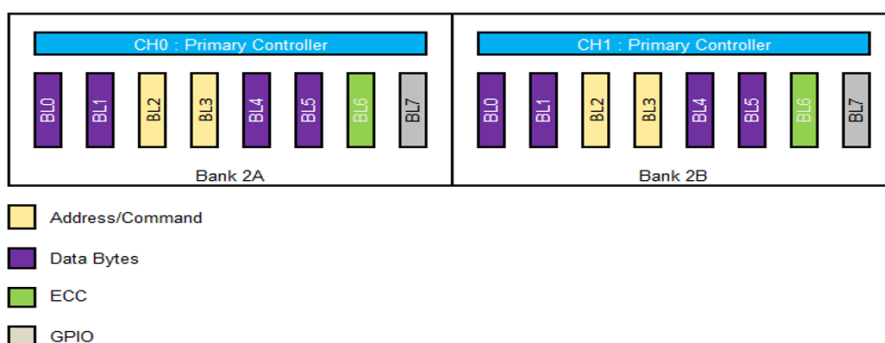
Observe the following general guidelines when placing pins for your Agilex 5 external memory interface:

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 DDR5](#) topic.
3. Not every byte lane can function as an address and command lane or a data lane. The pin assignment must adhere to the DDR5 data width mapping defined in [DDR5 Data Width Mapping](#).
4. A byte lane must not be used by both address and command pins and data pins.
5. An external memory interface can occupy one or more banks on the same edge. When an interface must occupy multiple banks, ensure that those banks are adjacent to one another.
 - If an I/O bank is shared between two interfaces—meaning that two sub-banks belong to two different EMIF interfaces—then both the interfaces must share the same voltage.
 - Sharing of I/O lanes within a sub-bank for two different EMIF interfaces is not permitted; I/O lanes within a sub-bank can be assigned to one EMIF interface only.
6. Any pin in the same bank that is not used by an external memory interface may not be available for use as a general purpose I/O pin:

- For fabric EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same sub-bank, pins in an I/O lane that is not assigned to an EMIF interface, can be used as general-purpose I/O pins.
 - For HPS EMIF, unused pins in an I/O lane assigned to an EMIF interface cannot be used as general-purpose I/O pins. In the same bank, pins in an I/O lane that is not assigned to an EMIF interface can be used as general-purpose I/O pins.
7. All address and command pins and their associated clock pins (CK_t and CK_c) must reside within a single sub-bank. The sub-bank containing the address and command pins is identified as the address and command sub-bank. Refer to the table in [DDR5 Data Width Mapping](#) for the supported address and command and data lane placements for DDR5.
 8. The address and command pins and their associated clock pins in the address and command bank must follow a fixed pin-out scheme, as defined in the *Agilex 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 [DDR5 Data Width Mapping](#). Be aware that not every byte lane can be used as an address and command lane or a data lane.

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

Figure 21. x72 DDR5 Pin Placement using Bank 2A and 2B



10. An unused I/O lane in the address and command sub-bank can serve to implement a data group, such as a x8 DQS group. The data group must be from the same controller as the address and command signals.
11. An I/O lane must not be used by both address and command pins and data pins.
12. Place read data groups according to the DQS grouping in the pin table and Pin Planner. Read data strobes (such as DQS_t and DQS_c) must reside at physical pins capable of functioning as DQS_t and DQS_c for a specific read data group size. You must place the associated read data pins (DQ), within the same group.

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

13. One of the sub-banks in the device (typically the sub-bank within corner bank 3A) may not be available if you use certain device configuration schemes. For some schemes, there may be an I/O lane available for EMIF data group.

- AVST-8 – This is contained entirely within the SDM, therefore all lanes of sub-bank 3A can be used by the external memory interface.
- AVST-16 – Lanes 4, 5, 6, and 7 are all effectively occupied and are not usable by the external memory interface.

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

7.3.3.2. x4 DIMM Implementation

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

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

Table 183. Mapping of DQS Signals Between DIMM and the EMIF IP

DIMM			Quartus Prime EMIF IP	
DQS0_A	DQ[3:0]_A		DQS0	DQ[3:0]_A
DQS5_A	DQ[7:4]_A		DQS1	DQ[7:4]_A
DQS1_A	DQ[11:8]_A		DQS2	DQ[11:8]_A
DQS6_A	DQ[15:12]_A		DQS3	DQ[15:12]_A
DQS2_A	DQ[19:16]_A		DQS4	DQ[19:16]_A
DQS7_A	DQ[23:20]_A		DQS5	DQ[23:20]_A
DQS3_A	DQ[27:24]_A		DQS6	DQ[27:24]_A
DQS8_A	DQ[31:28]_A		DQS7	DQ[31:28]_A
DQS4_A	CB[3:0]_A		DQS8	CB[3:0]_A
DQS9_A	CB[7:4]_A		DQS9	CB[7:4]_A
DQS0_B	DQ[3:0]_B		DQS10	DQ[3:0]_B
DQS5_B	DQ[7:4]_B		DQS11	DQ[7:4]_B
DQS1_B	DQ[11:8]_B		DQS12	DQ[11:8]_B
DQS6_B	DQ[15:12]_B		DQS13	DQ[15:12]_B
DQS2_B	DQ[19:16]_B		DQS14	DQ[19:16]_B
DQS7_B	DQ[23:20]_B		DQS15	DQ[23:20]_B
DQS3_B	DQ[27:24]_B		DQS16	DQ[27:24]_B
DQS8_B	DQ[31:28]_B		DQS17	DQ[31:28]_B
DQS4_B	CB[3:0]_B		DQS18	CB[3:0]_B
DQS9_B	CB[7:4]_B		DQS19	CB[7:4]_B

Data Bus Connection Mapping Flow

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

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

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

7.3.3.3. Specific Pin Connection Requirements

PLL

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

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

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

OCT

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

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

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

Address / Command / Parity

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

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

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

DQS/DQ/DM

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

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

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

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

7.3.3.4. Command and Address Signals

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

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

7.3.3.5. Clock Signals

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

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

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

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

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

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

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

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

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

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

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

7.3.4. Pin Placements for Agilex 5 FPGA DDR5 EMIF IP

The Agilex 5 EMIF IP for DDR5 supports fixed address and command pin placement, and fixed data lane placement.

7.3.4.1. Address and Command Pin Placement for DDR5

Table 184. Address and Command Pin Placement

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

The Agilex 5 EMIF IP for DDR5 supports fixed Address and Command pin placement as shown in the above table. The IP supports up to 2 ranks for the following schemes:

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

7.3.4.2. DDR5 Data Width Mapping

The EMIF IP for Agilex 5 devices does not support flexible data lane placement.

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

Table 185. Component

Controller	Address Command Scheme	Data Width Usage	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Primary	Scheme 1	DDR5x16	GPIO	GPIO	GPIO	GPIO	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Primary	Scheme 1	DDR5x16	DQ[1] ^S	DQ[0] ^S	AC1 ^S	AC0 ^S	X	X	X	GPIO
Primary & Secondary	Scheme 1	DDR5 2x16	DQ[1] ^S	DQ[0] ^S	AC1 ^S	AC0 ^S	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Primary	Scheme 1	DDR5x16 + ECC	GPIO	GPIO	GPIO	DQ[ECC] ^P	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Primary	Scheme 1	DDR5x32	GPIO	GPIO	DQ[3] ^P	DQ[2] ^P	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Primary	Scheme 1	DDR5x32 + ECC	GPIO	DQ[ECC] ^P	DQ[3] ^P	DQ[2] ^P	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Note: <ul style="list-style-type: none"> • X = Not available as GPIO. • ^P = Primary controller. • ^S = Secondary controller. 										

Table 186. DIMM Support

Data Width per Channel		A/C Placement	CH0							CH1								
			BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
DDR5 32		Ch0 Bot Sub-bank / Ch1 Bot Sub-bank	GPIO	GPIO	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]	GPIO	GPIO	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]
DDR5 32 + ECC		Ch0 Bot Sub-bank / Ch1 Bot Sub-bank	GPIO	DQ[EC _C]	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]	GPIO	DQ[EC _C]	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]
DDR5 32		Ch0 Bot Sub-bank / Ch1 Top Sub-bank	GPIO	GPIO	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]	DQ[1]	DQ[0]	AC[1]	AC[0]	DQ[2]	DQ[3]	X	GPIO
DDR5 32 + ECC		Ch0 Bot Sub-bank / Ch1 Top Sub-bank	GPIO	DQ[EC _C]	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]	DQ[1]	DQ[0]	AC[1]	AC[0]	DQ[2]	DQ[3]	DQ[EC _C]	GPIO
DDR5 32		Ch0 Top Sub-bank / Ch1	DQ[1]	DQ[0]	AC[1]	AC[0]	DQ[2]	DQ[3]	X	GPIO	GPIO	DQ[1]	DQ[3]	DQ[2]	AC[1]	AC[0]	DQ[0]	DQ[1]
			continued															

continued...

Data Width per Channel	A/C Placement	CH0							CH1								
		BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0
	Bot Sub-bank																
DDR5 32 + ECC	Ch0 Top Sub-bank / Ch1 Bot Sub-bank	DQ[1]	DQ[0]	AC[1]	AC[0]	DQ[2]	DQ[3]	DQ[EC _C]	GPIO	DQ[EC _C]	DQ[3]	DQ[2]	DQ[3]	AC[1]	AC[0]	DQ[0]	DQ[1]
Note: X means not available as GPIO.																	

Figure 22. DDR5 2chx16, Single Rank using x8 Memory Component

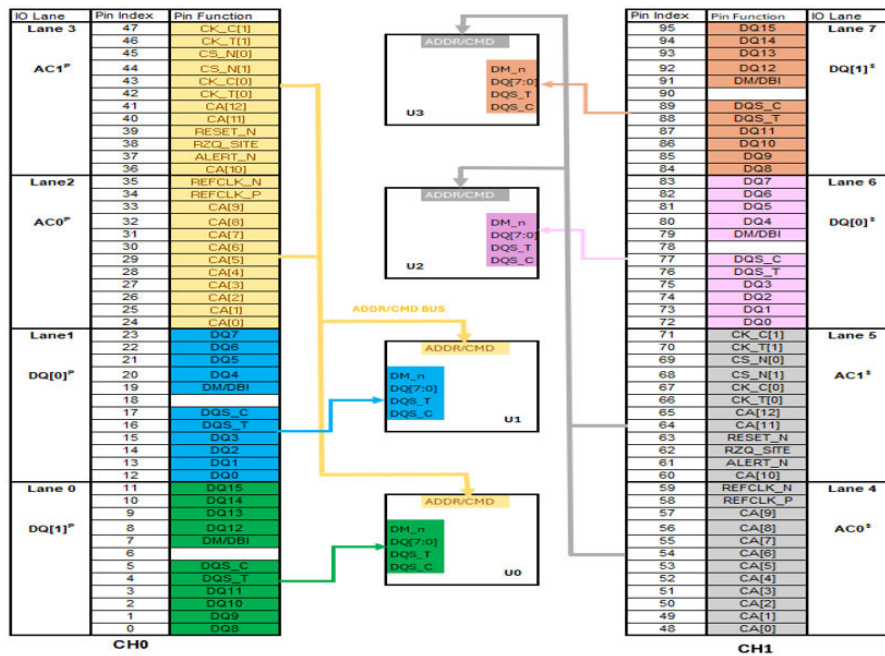


Figure 23. DDR5 x32 + ECC, Single Rank using x8 Memory Component

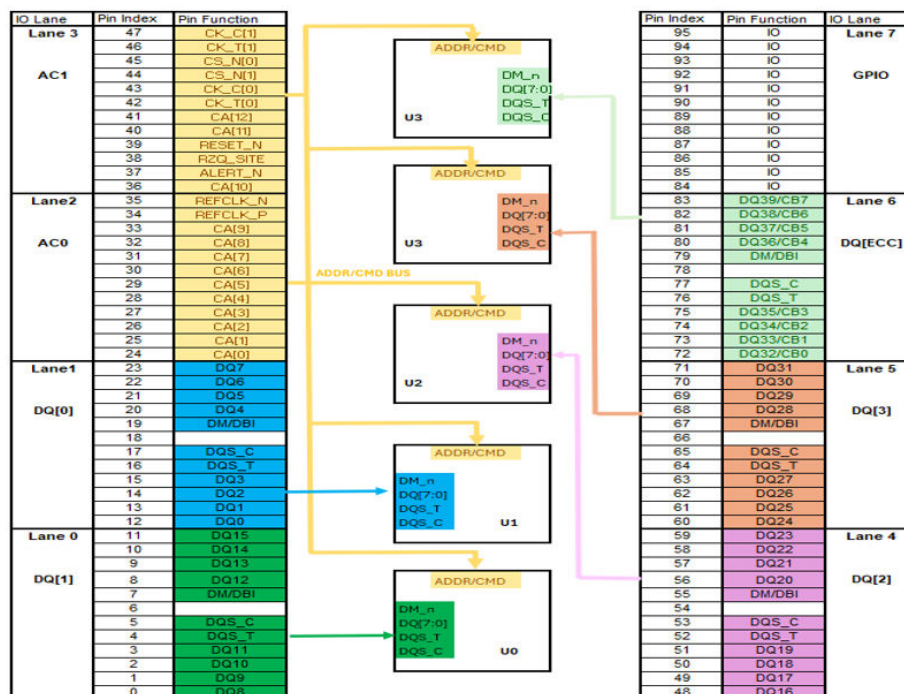
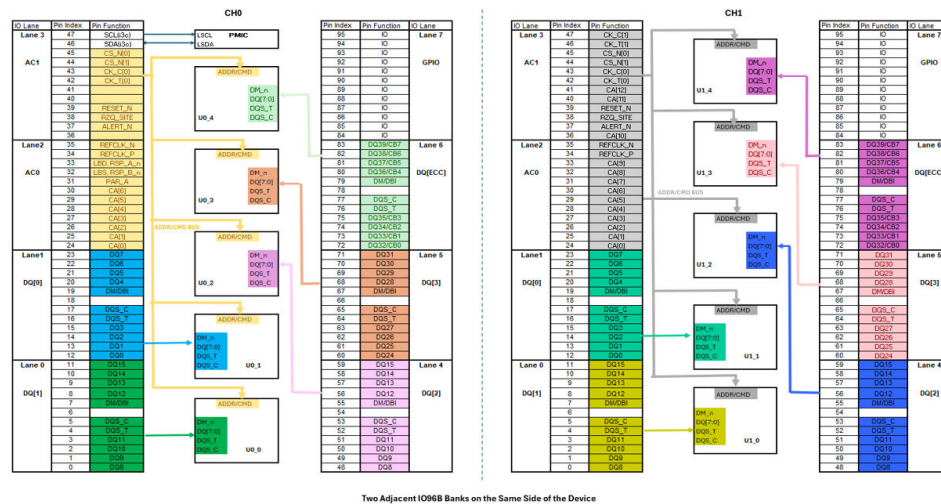


Figure 24. DDR5 x32 + ECC DIMM (CH0 Bottom Sub-bank, CH1 Bottom Sub-bank), Single Rank using x8 Memory Component



Two Adjacent IO66B Banks on the Same Side of the Device

7.3.5. 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).

7.3.5.1. DDR5 Byte Lane Swapping

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

The rules for swapping DQ byte lane are as follows:

- You can only swap between utilized DQ lanes.
- You cannot swap a DQ lane with an AC lane.
- You cannot swap a DQ lane with an ECC lane when out-of-band ECC is enabled. For x40 interfaces, the highest-indexed DQ byte lane cannot be swapped.
- For multi-channel configuration, you can only swap DQ lanes of the same channel.
- 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 187. Byte Lane Swapping

Address/ Command Scheme	Data Width per Channel	BL7 [P95:P84]	BL6 [P83:P72]	BL5 [P71:P60]	BL4 [P59:P48]	BL3 [P47:P36]	BL2 [P35:P24]	BL1 [P23:P12]	BL0 [P11:P0]
Scheme 1	DDR5 x32	GPIO	GPIO	DQ[3] ^P	DQ[2] ^P	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Scheme 1	DDR5 x32+ ECC	GPIO	DQ[ECC] ^P	DQ[3] ^P	DQ[2] ^P	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Scheme 1	DDR5 2x16	DQ[1] ^S	DQ[0] ^S	AC1 ^S	AC0 ^S	AC1 ^P	AC0 ^P	DQ[0] ^P	DQ[1] ^P
Note: • ^P = Primary controller. • ^S = Secondary controller.									

Example 1: DDR5 x32

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

Example 2: DDR5 x32 + ECC

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

Example 3: DDR5 2 x16

For multi-channel configuration, you can only swap DQ lanes of the same channel.

BL0 and BL1 are used as DQ lanes for one channel. Byte lane swapping is allowed on BL0 and BL1.

BL6 and BL7 are used as DQ lanes for another channel. Byte lane swapping is allowed on BL6 and BL7.

7.3.5.2. DDR5 Address and Command and CLK Lane

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

Pin mapping must adhere to the requirements defined in the table in the [Address and Command Pin Placement for DDR5](#) topic.

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

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

7.3.5.3. DDR5 Interface x8 Data Lane

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

For DDR5 interfaces composed of x8 devices, two pins are reserved for DQS-T and DQS-C signals, one pin is reserved for the optional DM signal, one pin must be reserved, and the remaining eight pins are for DQ signals. One-byte data lane must be

assigned for each byte lane, where the byte lane covers DQ [0:7], DQS_T/DQS_C and DM_N. The following are EMIF I/O pin swapping restrictions applicable to a DDR5 interface with a x8 data lane:

- DQS_T must go to pin 4 in IO12 pins.
- DQS_C must go to pin 5 in IO12 pins.
- DM_N must go to pin 6 in IO12 pins. If the interface does not use the DM_N pin, this pin 6 in IO12 lane must remain unconnected.
- Pin 7 in IO12 lane remains unconnected. Altera recommends that you connect this pin 7 to the T_{DQS} dummy load of the memory component and route it as a differential trace along with DM_N (pin 6). This facilitates x4 or x8 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 188. Pin Swapping Rules for DDR5 x8 Interfaces

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

7.3.5.4. DDR5 Interface x4 Data Lane

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

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

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

Table 189. Pin Swapping Rules for DDR5 x4

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

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

Note:

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

You are responsible for ensuring that these conditions are met.

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

For guidelines on designing your PCB, refer to the *EMIF PCB Routing Guidelines* section in the *PCB Design Guidelines (HSSI, EMIF, MIPI, True Differential, PDN) User Guide: Agilex 5 FPGAs and SoCs* document.

7.3.5.5. Pin Swizzling

For information on pin swizzling, refer to [Configuring DQ Pin Swizzling](#) in the *External Memory Interfaces (EMIF) IP Design Example User Guide: Agilex 5 FPGAs and SoCs*.

8. Agilex 5 FPGA EMIF IP - LPDDR4 Support

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

8.1. External Memory Interfaces (EMIF) 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.

Table 190. Group: High-level Configuration / Memory Device

Parameter Name	Description
Use LPDDR4X	If True: using LPDDR4X; If False: using LPDDR4 Default value is false (Identifier: MEM_TECH_IS_X)
Number of Channels	Specifies the number of channels that the interface should implement. For multi-channel devices, this should always match the number of channels on the device. Default value is 2 Legal values are: 1, 2, 4 (Identifier: MEM_NUM_CHANNELS)
Data DQ Width	Number of DQ pins per memory channel, used for data. Default value is 16 Legal values are: 16, 32 (Identifier: MEM_CHANNEL_DATA_DQ_WIDTH)
Die Density	Capacity of each memory die (in Gbits), per channel per die. For dual-die packages, this is the density of each die, not the density of the full package. Default value is 16 Legal values are: 1, 2, 3, 4, 6, 8, 12, 16 (Identifier: MEM_DIE_DENSITY_GBITS)
CS Width	Specifies the total number of CS pins used by each channel. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_CHANNEL_CS_WIDTH)
Auto-set Memory Operating Frequency	if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_OPERATING_FREQ_MHZ_AUTOSSET_EN)
Memory Operating Frequency	Specifies the frequency at which the memory interface will run. Legal values are: 800, 1066.667, 1333.333, 1600, 1866.667, 2133.333
<i>continued...</i>	

© Altera Corporation. Altera, the Altera logo, the 'a' logo, and other Altera marks are trademarks of Altera Corporation. Altera and Intel warrant performance of its FPGA and semiconductor products to current specifications in accordance with Altera's or Intel's standard warranty as applicable, but reserves the right to make changes to any products and services at any time without notice. Altera and Intel assume no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera or Intel. Altera and Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

Parameter Name	Description
	(Identifier: MEM_OPERATING_FREQ_MHZ)

Table 191. Group: High-level Configuration / PHY

Parameter Name	Description
Auto-set PLL Reference Clock Frequency	if true, let IP select max PLL refclk frequency that this configuration can support. If false, user can set custom value for PLL refclk frequency. Default value is true (Identifier: PHY_REFCLK_FREQ_MHZ_AUTOSET_EN)
Enable Advanced List of PLL Reference Clock Frequencies	If true, provide extended list of possible refclk values. Otherwise, prune possible list of refclk values to a more reasonable length. Default value is false (Identifier: PHY_REFCLK_ADVANCED_SELECT_EN)
Reference Clock Frequency	Specifies the reference clock frequency for the EMIF IOPLL. (Identifier: PHY_REFCLK_FREQ_MHZ)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms I/O BANK and TOP vs BOT part of the I/O BANK). Legal ranges are derived from device floorplan. Default value is BOT Legal values are: BOT, TOP, FULL (Identifier: PHY_AC_PLACEMENT)
Auto-set Mainband Access Mode	if true, let IP select most likely usecase for the PHY_MAINBAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_MAINBAND_ACCESS_MODE_AUTOSET_EN)
Mainband Access Mode	Specifies the path through which the EMIF QHIP mainband interface is exposed to the user. The mainband interface is the AXI4 interface to the memory controller. Legal values are: NOC, ASYNC, SYNC (Identifier: PHY_MAINBAND_ACCESS_MODE)
Auto-set Sideband Access Mode	if true, let IP select most likely usecase for the PHY_SIDEHAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_SIDEHAND_ACCESS_MODE_AUTOSET_EN)
Sideband Access Mode	Specifies the path through which the EMIF QHIP sideband interface is exposed to the user. The sideband interface is the AXI4-Lite interface to the IOSSM. Legal values are: NOC, FABRIC (Identifier: PHY_SIDEHAND_ACCESS_MODE)
Pin Swizzle Map	Specifies the swizzle map for the data lanes and pins. (Identifier: PHY_SWIZZLE_MAP)
Use Debug Toolkit	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. Default value is false
<i>continued...</i>	

Parameter Name	Description
	(Identifier: DEBUG_TOOLS_EN)
Instance ID	<p>Instance ID of the EMIF IP. This is useful when using a discovery mechanism over the side-band interface, to identify which EMIF instance's mailbox is at which offset. If expecting to use a discovery mechanism in hardware, this parameter must be set uniquely for all EMIFs that share a sideband. Otherwise, this parameter can be ignored / kept at the default value.</p> <p>Default value is 0 Legal values are: from 0 to 6 (Identifier: INSTANCE_ID)</p>

Table 192. Group: High-level Configuration / Controller

Parameter Name	Description
Use In-Line ECC	<p>Specifies whether in-line ECC is enabled in the controller.</p> <p>Default value is false (Identifier: CTRL_ECC_INLINE_EN)</p>
Use ECC Autocorrection	<p>If ECC is enabled, specifies whether single-bit-errors (SBEs) should be corrected or just reported.</p> <p>Default value is true (Identifier: CTRL_ECC_AUTOCORRECT_EN)</p>
Use Data Masking	<p>Specifies whether Data Masking is enabled by the controller. When ECC is enabled, RMWs will occur (to recompute / write ECC), regardless of whether this is enabled.</p> <p>Default value is false (Identifier: CTRL_DM_EN)</p>
Use WDBI	<p>Specifies whether write Data-bus-inversion is enabled by the controller.</p> <p>Default value is false (Identifier: CTRL_WR_DBI_EN)</p>
Use RDBI	<p>Specifies whether read Data-bus-inversion is enabled by the controller.</p> <p>Default value is false (Identifier: CTRL_RD_DBI_EN)</p>

Table 193. Group: Advanced: Memory Timing / Overrides / JEDEC_TABLE

Parameter Name	Description
JEDEC Parameter	<p>Name of JEDEC Parameter to explicitly override; the values will be applied and appear in the list below.</p> <p>Default value is</p> <p>Legal values are: MEM_CL_CYC, MEM_CWL_CYC, MEM_RD_POSTAMBLE_CYC, MEM_WR_POSTAMBLE_CYC, MEM_TSR_NS, MEM_TRFCAB_NS, MEM_TRFCPB_NS, MEM_TXSR_NS, MEM_TXP_NS, MEM_TCCD_NS, MEM_TRTP_NS, MEM_TRCD_NS, MEM_TRPPB_NS, MEM_TRPAB_NS, MEM_TRAS_NS, MEM_TWR_NS, MEM_TWTR_NS, MEM_TRRD_NS, MEM_TPPD_CYC, MEM_TFAW_NS, MEM_TRC_NS, MEM_TREFW_NS, MEM_MINNUMREFSREQ, MEM_TREFI_NS, MEM_TCKE_NS, MEM_TCMDCKE_NS, MEM_TCKELCK_NS, MEM_TCSCKE_NS, MEM_TCKCKEH_NS, MEM_TCSCKEH_NS, MEM_TMRWCKEL_NS, MEM_TZQCKE_NS, MEM_TMRN_NS, MEM_TMRW_NS, MEM_TMRD_NS, MEM_TESCKE_NS, MEM_TZQCAL_NS, MEM_TZQLAT_NS, MEM_TDQSCK_MAX_NS, MEM_TDQSCK_MIN_NS, MEM_TCKCKEL_NS, MEM_TCKELCMD_NS, MEM_TCKEHCMD_NS (Identifier: JEDEC_OVERRIDE_TABLE_PARAM_NAME)</p>

Table 194. Group: Advanced: Memory Timing / Values

Parameter Name	Description
Read Latency	Read Latency of the memory device in clock cycles. (Identifier: MEM_CL_CYC)
Write Latency	Write Latency in clock cycles. (Identifier: MEM_CWL_CYC)
Read Postamble Cycles	Duration of read postamble in cycles. (Identifier: MEM_RD_POSTAMBLE_CYC)
Write Postamble Cycles	Duration of write postamble in cycles. (Identifier: MEM_WR_POSTAMBLE_CYC)
tSR	Minimum duration (Entry to Exit) of Self Refresh in nanoseconds. (Identifier: MEM_TSR_NS)
tRFCab	All-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_TRFCAB_NS)
tRFCpb	Per-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_TRFCPB_NS)
tXSR	Self-Refresh Exit to Next Valid Command Delay Time in nanoseconds. (Identifier: MEM_TXSR_NS)
tXP	Exit Power-Down to Next Valid Command Delay Time in nanoseconds. (Identifier: MEM_TXP_NS)
tCCD	CAS-to-CAS Delay in nanoseconds. (Identifier: MEM_TCCD_NS)
tRTP	Internal READ to PRECHARGE Command Delay Time in nanoseconds. (Identifier: MEM_TRTP_NS)
tRCD	RAS-to-CAS Delay in nanoseconds. (Identifier: MEM_TRCD_NS)
tRPpb	Per-Bank Precharge Time in nanoseconds. (Identifier: MEM_TRPPB_NS)
tRPab	All-Bank Precharge Time in nanoseconds. (Identifier: MEM_TRPAB_NS)
tRAS	Row Active Time in nanoseconds. (Identifier: MEM_TRAS_NS)
tWR	Write Recovery Time in nanoseconds. (Identifier: MEM_TWR_NS)
tWTR	Write-to-Read Delay in nanoseconds. (Identifier: MEM_TWTR_NS)
tRRD	RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time in nanoseconds. (Identifier: MEM_TRRD_NS)
tFAW	Four-bank ACTIVE window time in nanoseconds. (Identifier: MEM_TFAW_NS)
tRC	Activate-to-Activate command period (same bank) in nanoseconds.
<i>continued...</i>	

Parameter Name	Description
	(Identifier: MEM_TRC_NS)
tREFW	Refresh window time in nanoseconds. (Identifier: MEM_TREFW_NS)
Min Number of Refs Req'd	Minimum Number of Refreshes Required. (Identifier: MEM_MINNUMREFSREQ)
tREFI	Refresh Interval Time in nanoseconds. (Identifier: MEM_TREFI_NS)
tCKE	CKE Minimum Pulse Width time in nanoseconds. (Identifier: MEM_TCKE_NS)
tCMDCKE	Delay from valid command to power-down-entry (CKE low) in nanoseconds. (Identifier: MEM_TCMDCKE_NS)
tCKELCK	Valid clock requirement after power-down-entry in nanoseconds. (Identifier: MEM_TCKELCK_NS)
tCSCKE	Valid CS requirement before power-down-entry (CKE low) in nanoseconds. (Identifier: MEM_TCSCKE_NS)
tCKCKEH	Valid clock requirement before power-down-exit in nanoseconds. (Identifier: MEM_TCKCKEH_NS)
tCSCKEH	Valid CS requirement before power-down-exit (CKE high) in nanoseconds. (Identifier: MEM_TCSCKEH_NS)
tMRWCKEL	Delay from MRW command to power-down-entry (CKE low) in nanoseconds. (Identifier: MEM_TMRWCKEL_NS)
tZQCKE	Delay from ZQCal Start command to power-down-entry (CKE low) in nanoseconds. (Identifier: MEM_TZQCKE_NS)
tMRR	Mode Register Read Command Period Time in nanoseconds. (Identifier: MEM_TMRR_NS)
tMRW	Mode Register Write Command Period Time in nanoseconds. (Identifier: MEM_TMRW_NS)
tMRD	Mode Register Set Command Period Time in nanoseconds. (Identifier: MEM_TMRD_NS)
tESCKE	Delay from SRE command to CKE Input low in nanoseconds. (Identifier: MEM_TESCKE_NS)
tZQCAL	ZQ calibration time in nanoseconds. (Identifier: MEM_TZQCAL_NS)
tZQLAT	ZQcal Latch time in nanoseconds. (Identifier: MEM_TZQLAT_NS)
tDQSCK_MAX	Maximum DQS output access time from CK in nanoseconds. (Identifier: MEM_TDQSCK_MAX_NS)
tDQSCK_MIN	Minimum DQS output access time from CK in nanoseconds. (Identifier: MEM_TDQSCK_MIN_NS)
continued...	

Parameter Name	Description
tCKCKEL	Clock valid requirements after power-down-entry (CKE low) in nanoseconds. (Identifier: MEM_TCKCKEL_NS)
tCKELCMD	Valid command requirement after CKE input low in nanoseconds. (Identifier: MEM_TCKELCMD_NS)
tCKEHCMD	Valid command requirement after CKE input high in nanoseconds. (Identifier: MEM_TCKEHCMD_NS)

Table 195. Group: Advanced: Analog Overrides / Overrides / ANALOG_TABLE

Parameter Name	Description
Analog Parameter	Name of Analog Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: PHY_TERM_X_R_S_AC_OUTPUT_OHM, PHY_TERM_X_R_S_CK_OUTPUT_OHM, PHY_TERM_X_R_S_DQ_OUTPUT_OHM, PHY_TERM_X_DQ_SLEW_RATE, PHY_TERM_X_R_T_DQ_INPUT_OHM, PHY_TERM_X_DQ_VREF, PHY_TERM_X_R_T_REFCLK_INPUT_OHM, MEM_ODT_DQ_X_TGT_WR, MEM_ODT_DQ_X_IDLE, MEM_ODT_DQ_X_RON, MEM_VREF_DQ_X_RANGE, MEM_VREF_DQ_X_VALUE, MEM_ODT_CA_X_CA_COMM, MEM_ODT_CA_X_CA_ENABLE, MEM_ODT_CA_X_CS_ENABLE, MEM_ODT_CA_X_CK_ENABLE, MEM_VREF_CA_X_CA_RANGE, MEM_VREF_CA_X_CA_VALUE (Identifier: ANALOG_PARAM_DERIVATION_PARAM_NAME)

Table 196. Group: Advanced: Analog Overrides / Values

Parameter Name	Description
AC Drive Strength	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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_AC_OUTPUT_OHM)
CK 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_CK_OUTPUT_OHM)
FPGA DQ 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. Legal values are: SERIES_34_OHM_CAL, SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_DQ_OUTPUT_OHM)
DQ 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. <i>Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.</i> Legal values are: SLOW, MEDIUM, FAST, FASTEST (Identifier: PHY_TERM_X_DQ_SLEW_RATE)
DQ 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.
continued...	

Parameter Name	Description
	Legal values are: RT_40_OHM_CAL, RT_50_OHM_CAL, RT_60_OHM_CAL (Identifier: PHY_TERM_X_R_T_DQ_INPUT_OHM)
DQ 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. Legal values are: from 0 to 100 (Identifier: PHY_TERM_X_DQ_VREF)
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. Legal values are: RT_OFF, RT_DIFF (Identifier: PHY_TERM_X_R_T_REFCLK_INPUT_OHM)
Target Write Termination	Specifies the target termination to be used during a write. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6 (Identifier: MEM_ODT_DQ_X_TGT_WR)
DQ Idle Termination	Specifies the termination to be used for RTT_PARK and DQS_RTT_PARK. For power savings it is recommended to leave this as disabled. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. (Identifier: MEM_ODT_DQ_X_IDLE)
Memory DQ Drive Strength	Specifies the termination to be used when driving read data from memory. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: 6, 5, 4, 3, 2, 1 (Identifier: MEM_ODT_DQ_X_RON)
VrefDQ Range	Specifies which of the memory protocol defined ranges will be used. Legal values are: 1, 2 (Identifier: MEM_VREF_DQ_X_RANGE)
VrefDQ Value	Specifies the initial VrefDQ value to be used. Legal values are: from 15.00 to 44.90, from 10.00 to 30.00, from 32.90 to 62.90, from 22.00 to 42.00 (Identifier: MEM_VREF_DQ_X_VALUE)
CA Common Termination	Common termination value that can be applied to CA/CK/CS. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 1, 2, 3, 4, 5, 6 (Identifier: MEM_ODT_CA_X_CA_COMM)
CA Termination Enable	Enable the common termination value on the CA bus. Enabling CA termination will have no effect unless the ODT_CA bond pad is HIGH.. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: false, true (Identifier: MEM_ODT_CA_X_CA_ENABLE)
CS Termination Enable	Enable the common termination value on the CS bus. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: false, true (Identifier: MEM_ODT_CA_X_CS_ENABLE)
CK Termination Enable	Enable the common termination value on the CK bus. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. "off" means this termination is disabled.

continued...

Parameter Name	Description
	Legal values are: false, true (Identifier: MEM_ODT_CA_X_CK_ENABLE)
VrefCA Range	Specifies which of the memory protocol defined ranges will be used. Legal values are: 1, 2 (Identifier: MEM_VREF_CA_X_CA_RANGE)
VrefCA Value	Specifies the initial VrefCA value to be used. Legal values are: from 10.00 to 30.00, from 22.00 to 42.00 (Identifier: MEM_VREF_CA_X_CA_VALUE)

Table 197. Group: Example Design / Fileset Types

Parameter 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. Default value is VERILOG Legal values are: VERILOG, VHDL (Identifier: EX_DESIGN_HDL_FORMAT)
Generate Synthesis Fileset	Generate Synthesis Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SYNTH)
Generate Simulation Fileset	Generate Simulation Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SIM)

Table 198. Group: Example Design / User PLL

Parameter Name	Description
Auto-set User PLL Output Clock Frequency	if true, let IP select a reference clock frequency for the user PLL in the example design; if false, let user set a custom value for this parameter. Default value is true (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ_AUTOSET_EN)
User PLL Output Clock Frequency	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode). Default value is 570 (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ)
User PLL Reference Clock Frequency	PLL reference clock frequency in MHz for PLL supplying the core clock. Default value is 100 (Identifier: EX_DESIGN_USER_PLL_REFCLK_FREQ_MHZ)
NOC Reference Clock Frequency	Reference Clock Frequency for the NOC control IP. Default value is 100 Legal values are: 25, 100, 125 (Identifier: EX_DESIGN_NOC_PLL_REFCLK_FREQ_MHZ)

Table 199. Group: Example Design / Traffic Generator

Parameter Name	Description
Traffic Generator Remote Access	Specifies whether the Traffic Generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled.
<i>continued...</i>	

Parameter Name	Description
	Default value is JTAG Legal values are: EXPORT, JTAG (Identifier: EX_DESIGN_TG_CSR_ACCESS_MODE)
Traffic Generator Program	Specifies the traffic pattern to be run. Default value is MEDIUM Legal values are: SHORT, MEDIUM, LONG, INFINITE (Identifier: EX_DESIGN_TG_PROGRAM)

Table 200. Group: Example Design / Performance Monitor

Parameter Name	Description
Enable Performance Monitor for Channel 0	If true, example design will include a Performance Monitor instance connected to Channel 0. Default value is false (Identifier: EX_DESIGN_PMON_CH0_EN)
Enable Performance Monitor for Channel 1	If true, example design will include a Performance Monitor instance connected to Channel 1. Default value is false (Identifier: EX_DESIGN_PMON_CH1_EN)
Enable Performance Monitor for Channel 2	If true, example design will include a Performance Monitor instance connected to Channel 2. Default value is false (Identifier: EX_DESIGN_PMON_CH2_EN)
Enable Performance Monitor for Channel 3	If true, example design will include a Performance Monitor instance connected to Channel 3. Default value is false (Identifier: EX_DESIGN_PMON_CH3_EN)

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 for your external memory interface are known, 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, 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.

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 by performing the following steps:

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

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. 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 201. Pin Options for LPDDR4 x32 and 2CH 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
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
continued...			

Pins	1 CH x32	2 CH x16	
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
Clock Enable	CKE_A CKE_B	CKE_A	CKE_B
Reset	RESET_n	RESET_n (Resistor jumper to select from mem_0 or mem_1.) <i>Note:</i> The LPDDR4 EMIF IP would generate one reset pin to the memory module for both single channel x16/x32 or dual channel x16 designs. For early board bring up and easy debugging on a dual channel x16 design, you can plan to test each channel independently as a single channel design. You can design the board with two reset pins connected through a resistor jumper to choose the reset from the primary or secondary channel. The example for dual channel x16 LPDDR4 with two reset pins is shown in the two figures below. For mature designs, you can design the board with one reset pin connected from the primary channel.	

Figure 25. Single-Channel x32 LPDDR4, Single Rank

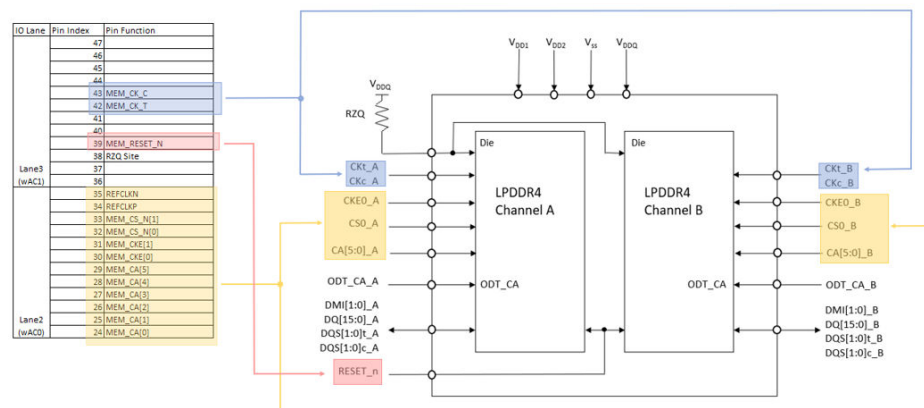
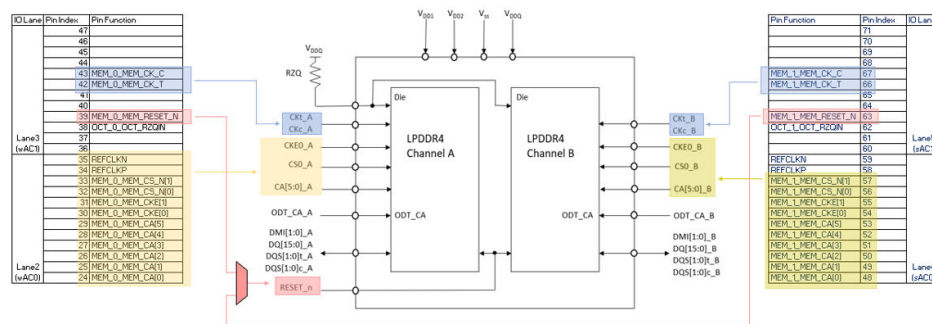


Figure 26. Dual-Channel x16 LPDDR4, Single Rank



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.

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

Table 202. Maximum Number of LPDDR4 Interfaces

Device	Package	1ch x32	2ch x16	4ch x16
A5EA005B/ A5EA007B	B15A ¹	—	—	—
A5EA005B / A5EA007B	B23B	1	1	—
A5EG005B / A5EG007B	B18A	—	—	—
A5EC013B ES / A5ED013B ES	B23A	1	1	—
A5EC008B / A5EC013A / A5EC013B / A5ED008B / A5ED013A / A5ED013B	B23A	1	1	—
A5EC008B / A5EC013A / A5EC013B / A5ED008B / A5ED013A / A5ED013B	B32A	2	2	—
A5EC008B / A5EC013B / A5ED008B / A5ED013B	M16A	2	2	—
A5EA008B / A5EA013B A5EB008B / A5EB013B A5EE008B / A5EE013B	B23B	2	2	—
A5EC028A / A5EC028B / A5ED028A / A5ED028B	B23A	1	1	—
A5EC028A / A5EC028B / A5ED028A / A5ED028B	B32A	2	2	—
A5EC028B / A5ED028B	M16A	2	2	—

continued...

Device	Package	1ch x32	2ch x16	4ch x16
A5EA028B / A5EB028B / A5EE028B	B23B	2	2	—
A5EC065B ES / A5ED065B ES	B23A	1	1	—
A5EC043A / A5EC043B / A5EC052A / A5EC052B / A5EC065A / A5EC065B / A5ED043A / A5ED043B / A5ED052A / A5ED052B / A5ED065A / A5ED065B	B23A	1	1	—
A5EC065B ES / A5ED065B ES	B32A	4	4	2
A5EC043A / A5EC043B / A5EC052A / A5EC052B / A5EC065A / A5EC065B / A5ED043A / A5ED043B / A5ED052A / A5ED052B / A5ED065A / A5ED065B	B32A	4	4	2
A5DC064A ES / A5DD064A ES	B32B	4	4	2
A5DC051A / A5DC064A / A5DD051A / A5DD064A	B32B	4	4	2

¹ The B15A package can support one instance of 1ch x16 LPDDR4 interface.

4ch x16 interface requires two adjacent IO96B banks located on the same edge of the device.

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 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*.

Related Information

[Specific Pin Connection Requirements](#) on page 200

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. 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*.

Related Information

[Specific Pin Connection Requirements](#) on page 200

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.

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).

8.2.3.1. 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. A byte lane must not be used by both address and command pins and data pins.
5. 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.
6. 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 [LPDDR4 Data Width Mapping](#) topic.
7. 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. Refer to [LPDDR4 Data Width Mapping](#) for information on data width mapping.
8. 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 can be used as general-purpose I/O pins.
9. 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.
10. 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.
11. An I/O lane must not be used by both address and command pins and data pins.
12. Place read data groups according to the DQS grouping in the pin table and Pin Planner. Read data strobes (such as DQS_t and DQS_c) must reside at physical pins capable of functioning as DQS_t and DQS_c for a specific read data group size. You must place the associated read data pins (DQ), within the same group.
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 – 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 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.

8.2.3.2. 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: 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

Related Information

- [OCT](#) on page 197
- [PLL](#) on page 197

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

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.

8.2.3.5. 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	
	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]
continued...		

Address/Command Lane	Index Within Byte Lane	LPDDR4
	2	MEM_CA[2]
	1	MEM_CA[1]
	0	MEM_CA[0]

8.2.3.6. LPDDR4 Data Width Mapping

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	GPIO	AC1 P	AC0 P	DQ[1] P	DQ[0] P								
Primary	LPDDR4 1ch x16	GPIO	GPIO	GPIO	GPIO	GPIO	AC1 P	AC0 P	wDQ[1] P	wDQ[0] P								
Primary + Secondary	LPDDR4 2ch x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC1 P	AC0-P P	DQ[1] P	DQ[0] P								
Primary + Secondary	LPDDR4 4ch x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC1 P	AC0-P P	DQ[1] P	DQ[0] P								DQ[0] P
Note:										<ul style="list-style-type: none"> P Primary controller S Secondary controller 								

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 27. Dual-Channel x16 LPDDR4, Single Rank

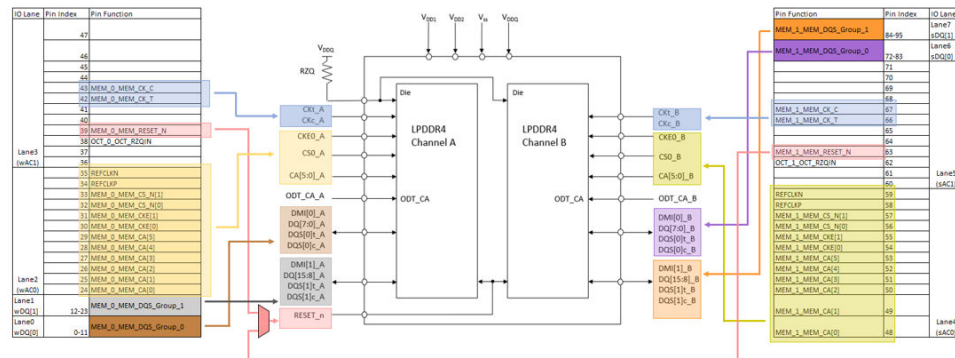
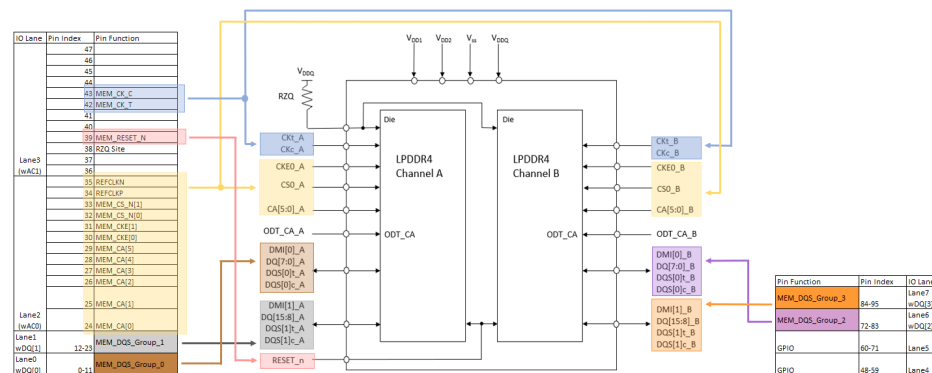


Figure 28. Single-Channel x32 LPDDR4, Single Rank



8.2.3.7. 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 204. 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 2ch x16	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 2ch x16

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.

For guidelines on designing your PCB, refer to the *EMIF PCB Routing Guidelines* section in the *PCB Design Guidelines (HSSI, EMIF, MIPI, True Differential, PDN) User Guide: Agilex 5 FPGAs and SoCs* document.

9. Agilex 5 FPGA EMIF IP - LPDDR5 Support

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

9.1. External Memory Interfaces (EMIF) 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.

Table 205. Group: High-level Configuration / Memory Device

Parameter Name	Description
Number of Channels	Specifies the number of channels that the interface should implement. For multi-channel devices, this should always match the number of channels on the device. Default value is 2 Legal values are: 1, 2, 4 (Identifier: MEM_NUM_CHANNELS)
Data DQ Width	Number of DQ pins per memory channel, used for data. Default value is 16 Legal values are: 16, 32 (Identifier: MEM_CHANNEL_DATA_DQ_WIDTH)
Die Density	Capacity of each memory die (in Gbits), per channel per die. For dual-die packages, this is the density of each die, not the density of the full package. Default value is 32 Legal values are: 2, 3, 4, 6, 8, 12, 16, 24, 32 (Identifier: MEM_DIE_DENSITY_GBITS)
CS Width	Specifies the total number of CS pins used by each channel. Default value is 1 Legal values are: 1, 2 (Identifier: MEM_CHANNEL_CS_WIDTH)
Auto-set Memory Operating Frequency - FSP0	If true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_FSP0_OPERATING_FREQ_MHZ_AUTOSSET_EN)
Memory Operating Frequency - FSP0	Specifies the FSP0 operating frequency of the memory interface in MHz. This is not the same as boot-frequency (boot frequency is not a user parameter). Legal values are: 800, 1066.667, 1375, 1600, 1866.667, 2133.333, 2400, 2750 (Identifier: MEM_FSP0_OPERATING_FREQ_MHZ)
continued...	

© Altera Corporation. Altera, the Altera logo, the 'a' logo, and other Altera marks are trademarks of Altera Corporation. Altera and Intel warrant performance of its FPGA and semiconductor products to current specifications in accordance with Altera's or Intel's standard warranty as applicable, but reserves the right to make changes to any products and services at any time without notice. Altera and Intel assume no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera or Intel. Altera and Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

Parameter Name	Description
Enable Frequency Set Point (FSP) 1	If true, users can enable and set values for a second frequency set point. Default value is false (Identifier: MEM_FSP1_EN)
Auto-set Memory Operating Frequency - FSP1	if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_FSP1_OPERATING_FREQ_MHZ_AUTOSET_EN)
Memory Operating Frequency - FSP1	Specifies the FSP1 operating frequency of the memory interface in MHz. Legal values are: 800, 1066.667, 1375, 1600, 1866.667, 2133.333, 2400, 2750 (Identifier: MEM_FSP1_OPERATING_FREQ_MHZ)
Enable Frequency Set Point (FSP) 2	If true, users can enable and set values for a third frequency set point. Default value is false (Identifier: MEM_FSP2_EN)
Auto-set Memory Operating Frequency - FSP2	if true, let IP select max frequency that this configuration can support for the current device speedgrade. If false, user can set custom value for operating frequency. Default value is true (Identifier: MEM_FSP2_OPERATING_FREQ_MHZ_AUTOSET_EN)
Memory Operating Frequency - FSP2	Specifies the FSP2 operating frequency of the memory interface in MHz. Legal values are: 800, 1066.667, 1375, 1600, 1866.667, 2133.333, 2400, 2750 (Identifier: MEM_FSP2_OPERATING_FREQ_MHZ)
Memory Operating Frequency	Specifies the frequency at which the memory interface will run. (Identifier: MEM_OPERATING_FREQ_MHZ)

Table 206. Group: High-level Configuration / PHY

Parameter Name	Description
Auto-set PLL Reference Clock Frequency	if true, let IP select max PLL refclk frequency that this configuration can support. If false, user can set custom value for PLL refclk frequency. Default value is true (Identifier: PHY_REFCLK_FREQ_MHZ_AUTOSET_EN)
Enable Advanced List of PLL Reference Clock Frequencies	If true, provide extended list of possible refclk values. Otherwise, prune possible list of refclk values to a more reasonable length. Default value is false (Identifier: PHY_REFCLK_ADVANCED_SELECT_EN)
Reference Clock Frequency	Specifies the reference clock frequency for the EMIF IOPLL. (Identifier: PHY_REFCLK_FREQ_MHZ)
AC Placement	Indicates location on the device where the interface will reside (specifically, the location of the AC lanes in terms I/O BANK and TOP vs BOT part of the I/O BANK). Legal ranges are derived from device floorplan. Default value is BOT Legal values are: BOT, TOP, FULL (Identifier: PHY_AC_PLACEMENT)
Auto-set Mainband Access Mode	if true, let IP select most likely usecase for the PHY_MAINBAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode.
continued...	

Parameter Name	Description
	Default value is true (Identifier: PHY_MAINBAND_ACCESS_MODE_AUTOSET_EN)
Mainband Access Mode	Specifies the path through which the EMIF QHIP mainband interface is exposed to the user. The mainband interface is the AXI4 interface to the memory controller. Legal values are: NOC, ASYNC, SYNC (Identifier: PHY_MAINBAND_ACCESS_MODE)
Auto-set Sideband Access Mode	if true, let IP select most likely usecase for the PHY_SIDEHAND_ACCESS_MODE; if false, let user set a custom value for sideband access mode. Default value is true (Identifier: PHY_SIDEHAND_ACCESS_MODE_AUTOSET_EN)
Sideband Access Mode	Specifies the path through which the EMIF QHIP sideband interface is exposed to the user. The sideband interface is the AXI4-Lite interface to the IOSSM. Legal values are: NOC, FABRIC (Identifier: PHY_SIDEHAND_ACCESS_MODE)
Pin Swizzle Map	Specifies the swizzle map for the data lanes and pins. (Identifier: PHY_SWIZZLE_MAP)
Use Debug Toolkit	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. Default value is false (Identifier: DEBUG_TOOLS_EN)
Instance ID	Instance ID of the EMIF IP. This is useful when using a discovery mechanism over the side-band interface, to identify which EMIF instance's mailbox is at which offset. If expecting to use a discovery mechanism in hardware, this parameter must be set uniquely for all EMIFs that share a sideband. Otherwise, this parameter can be ignored / kept at the default value. Default value is 0 Legal values are: from 0 to 6 (Identifier: INSTANCE_ID)

Table 207. Group: High-level Configuration / Controller

Parameter Name	Description
Use In-Line ECC	Specifies whether in-line ECC is enabled in the controller. Default value is false (Identifier: CTRL_ECC_INLINE_EN)
Use WR Link ECC	Specifies whether write link ECC is enabled in the controller. Default value is false (Identifier: CTRL_ECC_WR_LINK_EN)
Use RD Link ECC	Specifies whether read link ECC is enabled in the controller. Default value is false (Identifier: CTRL_ECC_RD_LINK_EN)
Use ECC Autocorrection	If ECC is enabled, specifies whether single-bit-errors (SBEs) should be corrected or just reported. Default value is true (Identifier: CTRL_ECC_AUTOCORRECT_EN)
<i>continued...</i>	

Parameter Name	Description
Use Data Masking	Specifies whether Data Masking is enabled by the controller. When ECC is enabled, RMWs will occur (to recompute / write ECC), regardless of whether this is enabled. Default value is false (Identifier: CTRL_DM_EN)
Use WDBI	Specifies whether write Data-bus-inversion is enabled by the controller. Default value is false (Identifier: CTRL_WR_DBI_EN)
Use RDBI	Specifies whether read Data-bus-inversion is enabled by the controller. Default value is false (Identifier: CTRL_RD_DBI_EN)

Table 208. Group: Advanced: Memory Timing / Overrides / JEDEC_TABLE

Parameter Name	Description
JEDEC Parameter	Name of JEDEC Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: MEM_FSP0_CL_CYC, MEM_FSP1_CL_CYC, MEM_FSP2_CL_CYC, MEM_FSP0_CWL_CYC, MEM_FSP1_CWL_CYC, MEM_FSP2_CWL_CYC, MEM_RDQS_POSTAMBLE_MODE, MEM_RD_PREAMBLE_CYC, MEM_RD_POSTAMBLE_CYC, MEM_WR_POSTAMBLE_CYC, MEM_MINNUMREFSREQ, MEM_TRCD_NS, MEM_TRPAB_NS, MEM_TRPPB_NS, MEM_TRAS_NS, MEM_TRAS_MAX_NS, MEM_TWR_NS, MEM_FSP0_TRRD_L_NS, MEM_FSP1_TRRD_L_NS, MEM_FSP2_TRRD_L_NS, MEM_FSP0_TRRD_S_NS, MEM_FSP1_TRRD_S_NS, MEM_FSP2_TRRD_S_NS, MEM_TFAW_NS, MEM_FSP0_TRBTP_NS, MEM_FSP1_TRBTP_NS, MEM_FSP2_TRBTP_NS, MEM_FSP0_TWTR_S_NS, MEM_FSP1_TWTR_S_NS, MEM_FSP2_TWTR_S_NS, MEM_FSP0_TWTR_L_NS, MEM_FSP1_TWTR_L_NS, MEM_FSP2_TWTR_L_NS, MEM_FSP0_TPPD_NS, MEM_FSP1_TPPD_NS, MEM_FSP2_TPPD_NS, MEM_TRC_NS, MEM_TZQLAT_NS, MEM_TPW_RESET_NS, MEM_TERQE_NS, MEM_TERQX_NS, MEM_TRDQE_OD_NS, MEM_TRDQX_OD_NS, MEM_TRDQSTFE_NS, MEM_TRDQSTFX_NS, MEM_TCCDMW_NS, MEM_TREFW_NS, MEM_TREFI_NS, MEM_FSP0_TRFCAB_NS, MEM_FSP1_TRFCAB_NS, MEM_FSP2_TRFCAB_NS, MEM_FSP0_TRFCPB_NS, MEM_FSP1_TRFCPB_NS, MEM_FSP2_TRFCPB_NS, MEM_FSP0_TPBR2PBR_NS, MEM_FSP1_TPBR2PBR_NS, MEM_FSP2_TPBR2PBR_NS, MEM_TPBR2ACT_NS, MEM_FSP0_TCKCSH_NS, MEM_FSP1_TCKCSH_NS, MEM_FSP2_TCKCSH_NS, MEM_FSP0_TCMDPD_NS, MEM_FSP1_TCMDPD_NS, MEM_FSP2_TCMDPD_NS, MEM_FSP0_TXP_NS, MEM_FSP1_TXP_NS, MEM_FSP2_TXP_NS, MEM_TCSH_NS, MEM_FSP0_TCSLCK_NS, MEM_FSP1_TCSLCK_NS, MEM_FSP2_TCSLCK_NS, MEM_FSP0_TCSPD_NS, MEM_FSP1_TCSPD_NS, MEM_FSP2_TCSPD_NS, MEM_FSP0_TMRWPD_NS, MEM_FSP1_TMRWPD_NS, MEM_FSP2_TMRWPD_NS, MEM_FSP0_TZQPD_NS, MEM_FSP1_TZQPD_NS, MEM_FSP2_TZQPD_NS, MEM_FSP0_TESPD_NS, MEM_FSP1_TESPD_NS, MEM_FSP2_TESPD_NS, MEM_TSR_NS, MEM_TXSR_NS, MEM_FSP0_TMRR_NS, MEM_FSP1_TMRR_NS, MEM_FSP2_TMRR_NS, MEM_FSP0_TMRW_NS, MEM_FSP1_TMRW_NS, MEM_FSP2_TMRW_NS, MEM_FSP0_TMRD_NS, MEM_FSP1_TMRD_NS, MEM_FSP2_TMRD_NS, MEM_TOSCO_NS, MEM_TDQSCK_NS (Identifier: JEDEC_OVERRIDE_TABLE_PARAM_NAME)

Table 209. Group: Advanced: Memory Timing / Values

Parameter Name	Description
[FSP0] Read Latency	[FSP0] Read Latency of the memory device in clock cycles.
<i>continued...</i>	

Parameter Name	Description
	(Identifier: MEM_FSP0_CL_CYC)
[FSP1] Read Latency	[FSP1] Read Latency of the memory device in clock cycles. (Identifier: MEM_FSP1_CL_CYC)
[FSP2] Read Latency	[FSP2] Read Latency of the memory device in clock cycles. (Identifier: MEM_FSP2_CL_CYC)
[FSP0] Write Latency	[FSP0] Write Latency in clock cycles. (Identifier: MEM_FSP0_CWL_CYC)
[FSP1] Write Latency	[FSP1] Write Latency in clock cycles. (Identifier: MEM_FSP1_CWL_CYC)
[FSP2] Write Latency	[FSP2] Write Latency in clock cycles. (Identifier: MEM_FSP2_CWL_CYC)
Read Postamble Mode	RDQS Postamble Mode. (Identifier: MEM_RDQS_POSTAMBLE_MODE)
Read Preamble Cycles	RDQS Preamble length (in cycles). (Identifier: MEM_RD_PREAMBLE_CYC)
Read Postamble Cycles	RDQS Postamble length (in cycles). (Identifier: MEM_RD_POSTAMBLE_CYC)
Write Postamble Cycles	WCK Postamble length (in cycles). (Identifier: MEM_WR_POSTAMBLE_CYC)
Min Number of Refs Req'd	Minimum Number of Refreshes Required. (Identifier: MEM_MINNUMREFSREQ)
tRCD	RAS-to-CAS Delay in nanoseconds. (Identifier: MEM_TRCD_NS)
tRPab	All-Bank Precharge Time in nanoseconds. (Identifier: MEM_TRPAB_NS)
tRPpb	Per-Bank Precharge Time in nanoseconds. (Identifier: MEM_TRPPB_NS)
tRAS	Row Active Time in nanoseconds. (Identifier: MEM_TRAS_NS)
tRAS_MAX	Specifies the maximum Activate-to-Precharge command period in nanoseconds. (Identifier: MEM_TRAS_MAX_NS)
tWR	Write Recovery Time in nanoseconds. (Identifier: MEM_TWR_NS)
[FSP0] tRRD_L	[FSP0] RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Long) in nanoseconds. (Identifier: MEM_FSP0_TRRD_L_NS)
[FSP1] tRRD_L	[FSP1] RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Long) in nanoseconds. (Identifier: MEM_FSP1_TRRD_L_NS)
[FSP2] tRRD_L	[FSP2] RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Long) in nanoseconds.
continued...	

Parameter Name	Description
	(Identifier: MEM_FSP2_TRRD_L_NS)
[FSP0] tRRD_S	[FSP0] RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Short) in nanoseconds. (Identifier: MEM_FSP0_TRRD_S_NS)
[FSP1] tRRD_S	[FSP1] RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Short) in nanoseconds. (Identifier: MEM_FSP1_TRRD_S_NS)
[FSP2] tRRD_S	[FSP2] RAS-to-RAS (Active Bank-A to Active Bank-B) Delay Time (Short) in nanoseconds. (Identifier: MEM_FSP2_TRRD_S_NS)
tFAW	Four-bank ACTIVE window time in nanoseconds. (Identifier: MEM_TFAW_NS)
[FSP0] tRBTP	[FSP0] Read Burst End to Precharge Command Delay in nanoseconds. (Identifier: MEM_FSP0_TRBTP_NS)
[FSP1] tRBTP	[FSP1] Read Burst End to Precharge Command Delay in nanoseconds. (Identifier: MEM_FSP1_TRBTP_NS)
[FSP2] tRBTP	[FSP2] Read Burst End to Precharge Command Delay in nanoseconds. (Identifier: MEM_FSP2_TRBTP_NS)
[FSP0] tWTR_S	[FSP0] Write-to-Read Delay (Short) in nanoseconds. (Identifier: MEM_FSP0_TWTR_S_NS)
[FSP1] tWTR_S	[FSP1] Write-to-Read Delay (Short) in nanoseconds. (Identifier: MEM_FSP1_TWTR_S_NS)
[FSP2] tWTR_S	[FSP2] Write-to-Read Delay (Short) in nanoseconds. (Identifier: MEM_FSP2_TWTR_S_NS)
[FSP0] tWTR_L	[FSP0] Write-to-Read Delay (Long) in nanoseconds. (Identifier: MEM_FSP0_TWTR_L_NS)
[FSP1] tWTR_L	[FSP1] Write-to-Read Delay (Long) in nanoseconds. (Identifier: MEM_FSP1_TWTR_L_NS)
[FSP2] tWTR_L	[FSP2] Write-to-Read Delay (Long) in nanoseconds. (Identifier: MEM_FSP2_TWTR_L_NS)
[FSP0] tPPD	[FSP0] Precharge-to-precharge delay in nanoseconds. (Identifier: MEM_FSP0_TPPD_NS)
[FSP1] tPPD	[FSP1] Precharge-to-precharge delay in nanoseconds. (Identifier: MEM_FSP1_TPPD_NS)
[FSP2] tPPD	[FSP2] Precharge-to-precharge delay in nanoseconds. (Identifier: MEM_FSP2_TPPD_NS)
tRC	Activate-to-Activate command period (same bank) in nanoseconds. (Identifier: MEM_TRC_NS)
tzQLAT	ZQCAL Latch Quiet Time in nanoseconds. (Identifier: MEM_TZQLAT_NS)
tpW_RESET	Min RESET_n low time for Reset Initialization with Stable Power Time in nanoseconds.
<i>continued...</i>	

Parameter Name	Description
	(Identifier: MEM_TPW_RESET_NS)
tERQE	Enhanced RDQS Toggle Mode Entry Time in nanoseconds. (Identifier: MEM_TERQE_NS)
tERQX	Enhanced RDQS Toggle Mode Exit Time in nanoseconds. (Identifier: MEM_TERQX_NS)
tRDQE_OD	ODT-disable from Enhanced RDQS Toggle Mode Entry Time in nanoseconds. (Identifier: MEM_TRDQE_OD_NS)
tRDQX_OD	ODT-enable from Enhanced RDQS Toggle Mode Exit Time in nanoseconds. (Identifier: MEM_TRDQX_OD_NS)
tRDQSTFE	Read/Write-based RDQS _t Training Mode Entry Time in nanoseconds. (Identifier: MEM_TRDQSTFE_NS)
tRDQSTFX	Read/Write-based RDQS _t Training Mode Exit Time in nanoseconds. (Identifier: MEM_TRDQSTFX_NS)
tCCDMW	CAS-to-CAS Delay for Masked Write in nanoseconds. (Identifier: MEM_TCCDMW_NS)
tREFW	Refresh Window in nanoseconds. (Identifier: MEM_TREFW_NS)
tREFI	Refresh Interval Time in nanoseconds. (Identifier: MEM_TREFI_NS)
[FSP0] tRFCab	[FSP0] All-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_FSP0_TRFCAB_NS)
[FSP1] tRFCab	[FSP1] All-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_FSP1_TRFCAB_NS)
[FSP2] tRFCab	[FSP2] All-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_FSP2_TRFCAB_NS)
[FSP0] tRFCpb	[FSP0] Per-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_FSP0_TRFCPB_NS)
[FSP1] tRFCpb	[FSP1] Per-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_FSP1_TRFCPB_NS)
[FSP2] tRFCpb	[FSP2] Per-Bank Refresh Cycle Time in nanoseconds. (Identifier: MEM_FSP2_TRFCPB_NS)
[FSP0] tpbR2pbR	[FSP0] Per-Bank Refresh to Per-Bank Refresh minimum interval time in nanoseconds. (Identifier: MEM_FSP0_TPBR2PBR_NS)
[FSP1] tpbR2pbR	[FSP1] Per-Bank Refresh to Per-Bank Refresh minimum interval time in nanoseconds. (Identifier: MEM_FSP1_TPBR2PBR_NS)
[FSP2] tpbR2pbR	[FSP2] Per-Bank Refresh to Per-Bank Refresh minimum interval time in nanoseconds. (Identifier: MEM_FSP2_TPBR2PBR_NS)
tpbR2ACT	Per-Bank Refresh to Activate minimum interval time in nanoseconds. (Identifier: MEM_TPBR2ACT_NS)
<i>continued...</i>	

Parameter Name	Description
[FSP0] tCKCSH	[FSP0] Valid Clock Requirement before CS goes High (Power-Down AC Timings) in nanoseconds. (Identifier: MEM_FSP0_TCKCSH_NS)
[FSP1] tCKCSH	[FSP1] Valid Clock Requirement before CS goes High (Power-Down AC Timings) in nanoseconds. (Identifier: MEM_FSP1_TCKCSH_NS)
[FSP2] tCKCSH	[FSP2] Valid Clock Requirement before CS goes High (Power-Down AC Timings) in nanoseconds. (Identifier: MEM_FSP2_TCKCSH_NS)
[FSP0] tCMDPD	[FSP0] Delay from valid command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP0_TCMDPD_NS)
[FSP1] tCMDPD	[FSP1] Delay from valid command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP1_TCMDPD_NS)
[FSP2] tCMDPD	[FSP2] Delay from valid command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP2_TCMDPD_NS)
[FSP0] tXP	[FSP0] Exit Power-Down to Next Valid Command Delay Time in nanoseconds. (Identifier: MEM_FSP0_TXP_NS)
[FSP1] tXP	[FSP1] Exit Power-Down to Next Valid Command Delay Time in nanoseconds. (Identifier: MEM_FSP1_TXP_NS)
[FSP2] tXP	[FSP2] Exit Power-Down to Next Valid Command Delay Time in nanoseconds. (Identifier: MEM_FSP2_TXP_NS)
tCSH	Minimum CS High Pulse Width at Power Down Exit in nanoseconds. (Identifier: MEM_TCSH_NS)
[FSP0] tCSLCK	[FSP0] Valid Clock Requirement after Power Down Entry in nanoseconds. (Identifier: MEM_FSP0_TCSLCK_NS)
[FSP1] tCSLCK	[FSP1] Valid Clock Requirement after Power Down Entry in nanoseconds. (Identifier: MEM_FSP1_TCSLCK_NS)
[FSP2] tCSLCK	[FSP2] Valid Clock Requirement after Power Down Entry in nanoseconds. (Identifier: MEM_FSP2_TCSLCK_NS)
[FSP0] tCSPD	[FSP0] Delay time from Power Down Entry to CS going High in nanoseconds. (Identifier: MEM_FSP0_TCSPD_NS)
[FSP1] tCSPD	[FSP1] Delay time from Power Down Entry to CS going High in nanoseconds. (Identifier: MEM_FSP1_TCSPD_NS)
[FSP2] tCSPD	[FSP2] Delay time from Power Down Entry to CS going High in nanoseconds. (Identifier: MEM_FSP2_TCSPD_NS)
[FSP0] tMRWPD	[FSP0] Delay from MRW Command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP0_TMRWPD_NS)
[FSP1] tMRWPD	[FSP1] Delay from MRW Command to Power Down Entry in nanoseconds.
continued...	

Parameter Name	Description
	(Identifier: MEM_FSP1_TMRWPD_NS)
[FSP2] tMRWPD	[FSP2] Delay from MRW Command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP2_TMRWPD_NS)
[FSP0] tZQPD	[FSP0] Delay from ZQ Calibration Start/Latch Command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP0_TZQPD_NS)
[FSP1] tZQPD	[FSP1] Delay from ZQ Calibration Start/Latch Command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP1_TZQPD_NS)
[FSP2] tZQPD	[FSP2] Delay from ZQ Calibration Start/Latch Command to Power Down Entry in nanoseconds. (Identifier: MEM_FSP2_TZQPD_NS)
[FSP0] tESPD	[FSP0] Delay time from Self-Refresh Entry command to Power Down Entry command in nanoseconds. (Identifier: MEM_FSP0_TESPD_NS)
[FSP1] tESPD	[FSP1] Delay time from Self-Refresh Entry command to Power Down Entry command in nanoseconds. (Identifier: MEM_FSP1_TESPD_NS)
[FSP2] tESPD	[FSP2] Delay time from Self-Refresh Entry command to Power Down Entry command in nanoseconds. (Identifier: MEM_FSP2_TESPD_NS)
tSR	Minimum Self-Refresh Time (Entry to Exit) in nanoseconds. (Identifier: MEM_TSR_NS)
tXSR	Exit Self-Refresh time in nanoseconds. (Identifier: MEM_TXSR_NS)
[FSP0] tMRR	[FSP0] Mode Register Read Command Period Time in nanoseconds. (Identifier: MEM_FSP0_TMRR_NS)
[FSP1] tMRR	[FSP1] Mode Register Read Command Period Time in nanoseconds. (Identifier: MEM_FSP1_TMRR_NS)
[FSP2] tMRR	[FSP2] Mode Register Read Command Period Time in nanoseconds. (Identifier: MEM_FSP2_TMRR_NS)
[FSP0] tMRW	[FSP0] Mode Register Write Command Period Time in nanoseconds. (Identifier: MEM_FSP0_TMRW_NS)
[FSP1] tMRW	[FSP1] Mode Register Write Command Period Time in nanoseconds. (Identifier: MEM_FSP1_TMRW_NS)
[FSP2] tMRW	[FSP2] Mode Register Write Command Period Time in nanoseconds. (Identifier: MEM_FSP2_TMRW_NS)
[FSP0] tMRD	[FSP0] Mode Register Set Command Period Time in nanoseconds. (Identifier: MEM_FSP0_TMRD_NS)
[FSP1] tMRD	[FSP1] Mode Register Set Command Period Time in nanoseconds. (Identifier: MEM_FSP1_TMRD_NS)
[FSP2] tMRD	[FSP2] Mode Register Set Command Period Time in nanoseconds.
<i>continued...</i>	

Parameter Name	Description
	(Identifier: MEM_FSP2_TMRD_NS)
tOSCO	Delay time from Stop WCK2DQI/WCK2DQO Interval Oscillator Command to Mode Register Readout time in nanoseconds. (Identifier: MEM_TOSCO_NS)
tDQSCK	DQS output access time from CK in nanoseconds. (Identifier: MEM_TDQSCK_NS)

Table 210. Group: Advanced: Analog Overrides / Overrides / ANALOG_TABLE

Parameter Name	Description
Analog Parameter	Name of Analog Parameter to explicitly override; the values will be applied and appear in the list below. Default value is Legal values are: PHY_TERM_X_R_S_AC_OUTPUT_OHM, PHY_TERM_X_R_S_CK_OUTPUT_OHM, PHY_TERM_X_R_S_DQ_OUTPUT_OHM, PHY_TERM_X_DQ_SLEW_RATE, PHY_TERM_X_R_T_DQ_INPUT_OHM, PHY_TERM_X_DQ_VREF, PHY_TERM_X_R_T_REFCLK_INPUT_OHM, PHY_DFE_X_TAP_1, PHY_DFE_X_TAP_2, PHY_DFE_X_TAP_3, PHY_DFE_X_TAP_4, MEM_ODT_DQ_X_TGT_WR, MEM_ODT_DQ_X_NON_TGT, MEM_ODT_DQ_X_RON, MEM_ODT_DQ_X_WCK, MEM_VREF_DQ_X_VALUE, MEM_ODT_CA_X_CA_COMM, MEM_ODT_CA_X_CA_ENABLE, MEM_ODT_CA_X_CS_ENABLE, MEM_ODT_CA_X_CK_ENABLE, MEM_VREF_CA_X_CA_VALUE, MEM_DFE_X_TAP_1 (Identifier: ANALOG_PARAM_DERIVATION_PARAM_NAME)

Table 211. Group: Advanced: Analog Overrides / Values

Parameter Name	Description
AC Drive Strength	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. Legal values are: SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_AC_OUTPUT_OHM)
CK 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. Legal values are: SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_CK_OUTPUT_OHM)
DQ 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: PHY_TERM_X_DQ_IO_STD_TYPE)
FPGA DQ 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. Legal values are: SERIES_40_OHM_CAL (Identifier: PHY_TERM_X_R_S_DQ_OUTPUT_OHM)
DQ 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. <i>Perform board simulations to determine the slew rate that provides the best eye opening for the data bus signals.</i> Legal values are: SLOW, MEDIUM, FAST, FASTEST (Identifier: PHY_TERM_X_DQ_SLEW_RATE)
continued...	

Parameter Name	Description
DQ 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. Legal values are: RT_40_OHM_CAL, RT_50_OHM_CAL, RT_60_OHM_CAL (Identifier: PHY_TERM_X_R_T_DQ_INPUT_OHM)
DQ 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. Legal values are: from 0.0 to 100.0 (Identifier: PHY_TERM_X_DQ_VREF)
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. Legal values are: RT_OFF, RT_DIFF (Identifier: PHY_TERM_X_R_T_REFCLK_INPUT_OHM)
PHY DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_1)
PHY DFE Tap 2	This parameter allows you to select the amount of bias used on tap 2 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_2)
PHY DFE Tap 3	This parameter allows you to select the amount of bias used on tap 3 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_3)
PHY DFE Tap 4	This parameter allows you to select the amount of bias used on tap 4 of the FPGA DFE. (Identifier: PHY_DFE_X_TAP_4)
Target Write Termination	Specifies the target termination to be used during a write. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6 (Identifier: MEM_ODT_DQ_X_TGT_WR)
DQ Non-Target Termination	Specifies the termination to be used for the non-target rank in a multi-rank configuration. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6 (Identifier: MEM_ODT_DQ_X_NON_TGT)
Memory DQ Drive Strength	Specifies the termination to be used when driving read data from memory. The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: 6, 5, 4, 3, 2, 1 (Identifier: MEM_ODT_DQ_X_RON)
Data Clock Termination	Specifies the termination to be used for the data clock (WCK). The value of this parameter represents X, where: termination = $RZQ/X = (240\text{ Ohm})/X$. Legal values are: off, 1, 2, 3, 4, 5, 6 (Identifier: MEM_ODT_DQ_X_WCK)
VrefDQ Value	Specifies the initial VrefDQ value to be used. Legal values are: from 10.0 to 73.5 (Identifier: MEM_VREF_DQ_X_VALUE)
continued...	

Parameter Name	Description
CA Common Termination	Common termination value that can be applied to CA/CK. The value of this parameter represents X, where: termination = $RZQ/X = (240 \text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: off, 1, 2, 3, 4, 5, 6 (Identifier: MEM_ODT_CA_X_CA_COMM)
CA Termination Enable	Enable the common termination value on the CA bus. The value of this parameter represents X, where: termination = $RZQ/X = (240 \text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: false, true (Identifier: MEM_ODT_CA_X_CA_ENABLE)
CS Termination Enable	For LPDDR5, this enables the fixed-value 80 Ohm ($RZQ/3$) CS termination if it is supported by the memory. The value of this parameter represents X, where: termination = $RZQ/X = (240 \text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: false, true (Identifier: MEM_ODT_CA_X_CS_ENABLE)
CK Termination Enable	Enable the common termination value on the CK bus. The value of this parameter represents X, where: termination = $RZQ/X = (240 \text{ Ohm})/X$. "off" means this termination is disabled. Legal values are: false, true (Identifier: MEM_ODT_CA_X_CK_ENABLE)
VrefCA Value	Specifies the initial VrefCA value to be used. Legal values are: from 10.0 to 73.5 (Identifier: MEM_VREF_CA_X_CA_VALUE)
MEM DFE Tap 1	This parameter allows you to select the amount of bias used on tap 1 of the memory DFE. (Identifier: MEM_DFE_X_TAP_1)

Table 212. Group: Example Design / Fileset Types

Parameter 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. Default value is VERILOG Legal values are: VERILOG, VHDL (Identifier: EX_DESIGN_HDL_FORMAT)
Generate Synthesis Fileset	Generate Synthesis Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SYNTH)
Generate Simulation Fileset	Generate Simulation Example Design. Default value is true (Identifier: EX_DESIGN_GEN_SIM)

Table 213. Group: Example Design / User PLL

Parameter Name	Description
Auto-set User PLL Output Clock Frequency	if true, let IP select a reference clock frequency for the user PLL in the example design; if false, let user set a custom value for this parameter. Default value is true

continued...

Parameter Name	Description
	(Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ_AUTOSET_EN)
User PLL Output Clock Frequency	Frequency of the core clock in MHz. This clock drives the traffic generator and NoC initiator (If in NoC mode). Default value is 570 (Identifier: EX_DESIGN_USER_PLL_OUTPUT_FREQ_MHZ)
User PLL Reference Clock Frequency	PLL reference clock frequency in MHz for PLL supplying the core clock. Default value is 100 (Identifier: EX_DESIGN_USER_PLL_REFCLK_FREQ_MHZ)
NOC Reference Clock Frequency	Reference Clock Frequency for the NOC control IP. Default value is 100 Legal values are: 25, 100, 125 (Identifier: EX_DESIGN_NOC_PLL_REFCLK_FREQ_MHZ)

Table 214. Group: Example Design / Traffic Generator

Parameter Name	Description
Traffic Generator Remote Access	Specifies whether the Traffic Generator control and status registers are accessible via JTAG, exported to the fabric, or just disabled. Default value is JTAG Legal values are: EXPORT, JTAG (Identifier: EX_DESIGN_TG_CSR_ACCESS_MODE)
Traffic Generator Program	Specifies the traffic pattern to be run. Default value is MEDIUM Legal values are: SHORT, MEDIUM, LONG, INFINITE (Identifier: EX_DESIGN_TG_PROGRAM)

Table 215. Group: Example Design / Performance Monitor

Parameter Name	Description
Enable Performance Monitor for Channel 0	If true, example design will include a Performance Monitor instance connected to Channel 0. Default value is false (Identifier: EX_DESIGN_PMON_CH0_EN)
Enable Performance Monitor for Channel 1	If true, example design will include a Performance Monitor instance connected to Channel 1. Default value is false (Identifier: EX_DESIGN_PMON_CH1_EN)
Enable Performance Monitor for Channel 2	If true, example design will include a Performance Monitor instance connected to Channel 2. Default value is false (Identifier: EX_DESIGN_PMON_CH2_EN)
Enable Performance Monitor for Channel 3	If true, example design will include a Performance Monitor instance connected to Channel 3. Default value is false (Identifier: EX_DESIGN_PMON_CH3_EN)

9.2. 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.

9.2.1. 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.

9.2.1.1. Estimating Pin Requirements

You should use the Quartus Prime software for final pin fitting.

However, you can estimate whether you have enough pins for your memory interface by performing the following steps:

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

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.

9.2.1.2. 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 216. Pin Options for LPDDR5 x32 and x16

Pins	1CH x32	2CH x16	
Data	32-bit DQ[15:0]_A DQ[15:0]_B	DQ[15:0]_A	DQ[15:0]_B
Data mask	DM[1:0]_A DM[1:0]_B	DM[1:0]_A	DM[1:0]_B
Read data strobe	RDQS[1:0]_t_A RDQS[1:0]_c_A RDQS[1:0]_t_B RDQS[1:0]_t_B	RDQS[1:0]_t_A RDQS[1:0]_c_A	RDQS[1:0]_t_B RDQS[1:0]_c_B
Write clock	WCK[1:0]_t_A WCK[1:0]_c_A WCK[1:0]_t_B WCK[1:0]_c_B	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.) Note: The LPDDR5 EMIF IP design would generate one reset pin to the memory module for both single channel x16/x32 or dual channel x16 designs. For early board bring up and easy debugging on dual channel x16 designs, you can test each channel independently as single channel design. You can design the board with two reset pins connected through a resistor jumper to choose the reset from the primary channel or secondary channel. The example for dual channel x16 LPDDR5 with two reset pins is shown in the figure below. For mature designs, you can design the board with one reset pin connected out from the primary channel.	

1CHX32 PINMAP (FRANK)

The diagram illustrates the pin connections for the 1CHX32 processor core, organized into four levels: Lvl5, Lvl4, BL3, and BL2. The pin list on the left shows the pin numbers and their functions. The block diagram on the right shows the internal components of the processor, including the Die, L2 Cache, and L3 Cache. The detailed pin connection table on the right lists the pin names and their corresponding functions.

Pin	Function
18	DIFFERENTIAL "B" SIDE REFERENCE CLOCK INPUT SIGNAL
19	DIFFERENTIAL "B" SIDE REFERENCE CLOCK INPUT SIGNAL
3	MENA_CAS[5]
8	MENA_CAS[5]
1	MENA_VCK[11:0]
1	MENA_VCK[11:0]
5	MENA_VCK[11:0]
4	MENA_VCK[11:0]
3	MENA_CAS[5]
2	MENA_CAS[5]
1	MENA_CAS[5]
1	MENA_CAS[5]

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*.

Table 217. Maximum Number of LPDDR5 Interfaces

Device	Package	1ch x32	2ch x16	4ch x16
A5EA005B/ A5EA007B	B15A ¹	—	—	—
A5EA005B / A5EA007B	B23B	1	1	—
A5EG005B / A5EG007B	B18A	—	—	—
A5EC013B ES / A5ED013B ES	B23A	1	1	—
A5EC008B / A5EC013A / A5EC013B / A5ED008B / A5ED013A / A5ED013B	B23A	1	1	—
A5EC008B / A5EC013A / A5EC013B / A5ED008B / A5ED013A / A5ED013B	B32A	2	2	—
A5EC008B / A5EC013B / A5ED008B / A5ED013B	M16A	2	2	—
A5EA008B / A5EA013B A5EB008B / A5EB013B A5EE008B / A5EE013B	B23B	2	2	—
A5EC028A / A5EC028B / A5ED028A / A5ED028B	B23A	1	1	—
A5EC028A / A5EC028B / A5ED028A / A5ED028B	B32A	2	2	—
A5EC028B / A5ED028B	M16A	2	2	—
A5EA028B / A5EB028B / A5EE028B	B23B	2	2	—
A5EC065B ES / A5ED065B ES	B23A	1	1	—
A5EC043A / A5EC043B / A5EC052A / A5EC052B / A5EC065A / A5EC065B / A5ED043A / A5ED043B / A5ED052A / A5ED052B / A5ED065A / A5ED065B	B23A	1	1	—
A5EC065B ES / A5ED065B ES	B32A	4	4	2
A5EC043A / A5EC043B / A5EC052A / A5EC052B / A5EC065A / A5EC065B / A5ED043A / A5ED043B / A5ED052A / A5ED052B / A5ED065A / A5ED065B	B32A	4	4	2
A5DC064A ES / A5DD064A ES	B32B	4	4	2
A5DC051A / A5DC064A / A5DD051A / A5DD064A	B32B	4	4	2

¹ The B15A package can support one instance of 1ch x16 LPDDR5 interface.

4ch x16 interface requires two adjacent IO96B banks located on the same edge of the device.

9.2.2. Agilex 5 FPGA EMIF IP Resources

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

9.2.2.1. 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](#).

9.2.2.2. 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](#).

9.2.3. 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).

9.2.3.1. 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, RLDRAMx, and QDRx 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 can be used as general-purpose I/O pins.
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. 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.
9. 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.
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 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.
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 – Lanes 4, 5, 6, and 7 are all effectively occupied and are not usable by the external memory interface.
13. Two memory interfaces cannot share an I/O 48 sub-bank.

9.2.3.2. 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.

9.2.3.3. 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.

9.2.3.4. 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.

9.2.4. Pin Placements for Agilex 5 FPGA LPDDR5 EMIF IP

9.2.4.1. 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 219. Address and Command Pin Placement

Address/Command Lane	Index Within Byte Lane	LPDDR5
AC1	11	
	10	
	9	
	8	CS_N[1]
	7	CK_C[0]
	6	CK_T[0]
	5	CS_N[0]
	4	CA[6]
	3	RESET_N
	2	RZQ Site
	1	
	0	
AC0	11	Differential "N-Side" reference clock input site
	10	Differential "P-Side" reference clock input site
	9	CA[5]
	8	CA[4]
	7	WCK_C[1]
	6	WCK_T[1]
	5	WCK_C[0]
	4	WCK_T[0]
	3	CA[3]
	2	CA[2]
	1	CA[1]
	0	CA[0]

9.2.4.2. 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 220. Component

Controler	Data Width Usage	BL7 [P95: P84]	BL6 [P83: P72]	BL5 [P71: P60]	BL4 [P59: P48]	BL3 [P47: P36]	BL2 [P35: P24]	BL1 [P23: P12]	BL0 [P11: P0]		BL7 [P95: P84]	BL6 [P83: P72]	BL5 [P71: P60]	BL4 [P59: P48]	BL3 [P47: P36]	BL2 [P35: P24]	BL1 [P23: P12]	BL0 [P11: P0]
Primary	LPDDR5 x16	GPIO	GPIO	GPIO	GPIO	AC1 P	AC0 P	DQ[1] P	DQ[0] P									
Primary & 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	LPDDR5 4ch x16	DQ[1] S	DQ[0] S	AC1 S	AC0 S	AC1 P	AC0 P	DQ[1] P	DQ[0] P									DQ[0] P

Note: • 1 ESO silicon supports LPDDR5x16 only on bottom sub-bank (BL0-BL3).
• P Primary controller.
• S Secondary controller.

Figure 31. LPDDR5 2chx16 , Single Rank

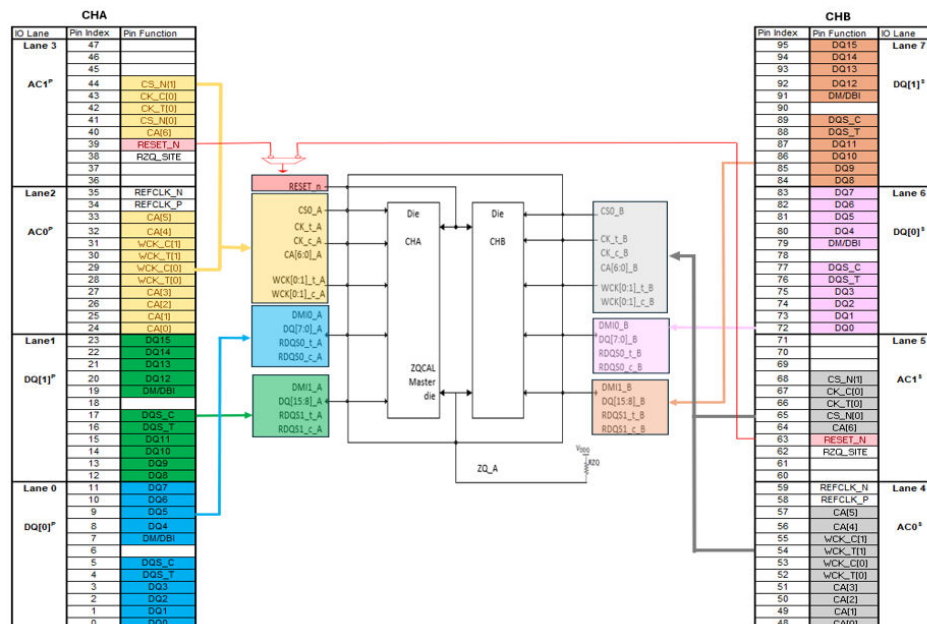
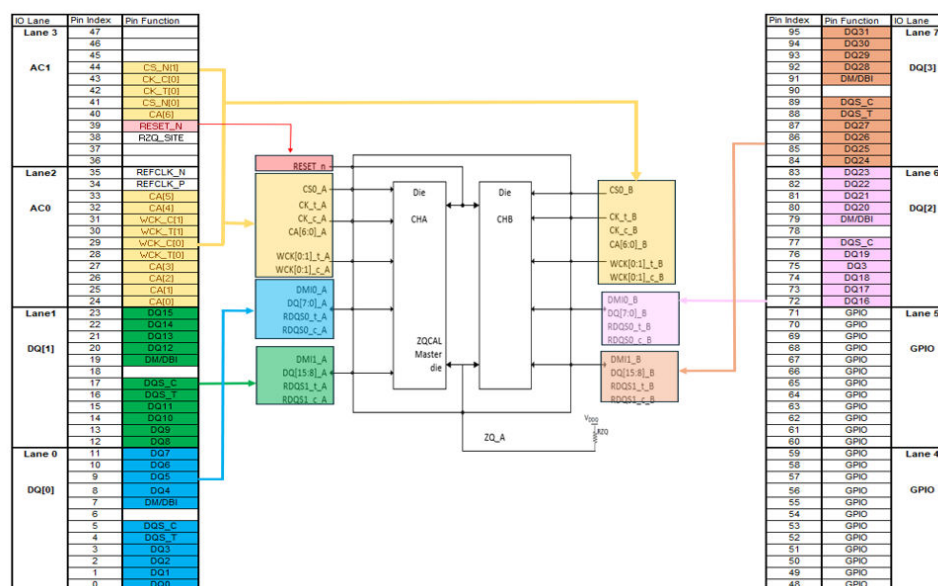


Figure 32. LPDDR5 1chx32 , Single Rank



Refer to the *Pin Options for LPDDR5 1ch x32* figure in the *LPDDR5 Component Options* topic for information on the T-Line routing requirement for address and command pins.

9.2.4.3. 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 221. 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 can be 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.

For guidelines on designing your PCB, refer to the *EMIF PCB Routing Guidelines* section in the *PCB Design Guidelines (HSSI, EMIF, MIPI, True Differential, PDN) User Guide: Agilex 5 FPGAs and SoCs* document.



10. 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.

10.1. 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.

10.1.1. 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.

10.1.1.1. 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.

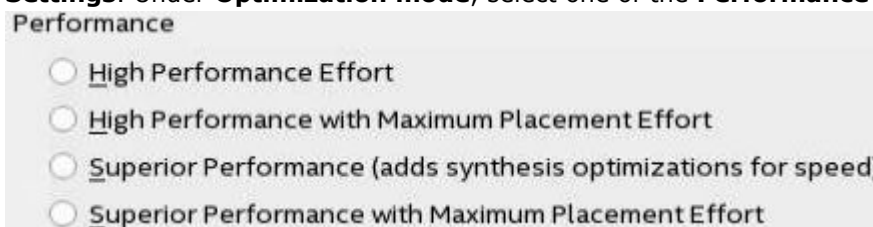
Core timing analysis excludes user logic timing to or from EMIF blocks. The EMIF IP provides a constrained clock (for example: ddr4_usr_clk) with which to clock customer logic; pll_afi_clk serves this purpose.

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`.

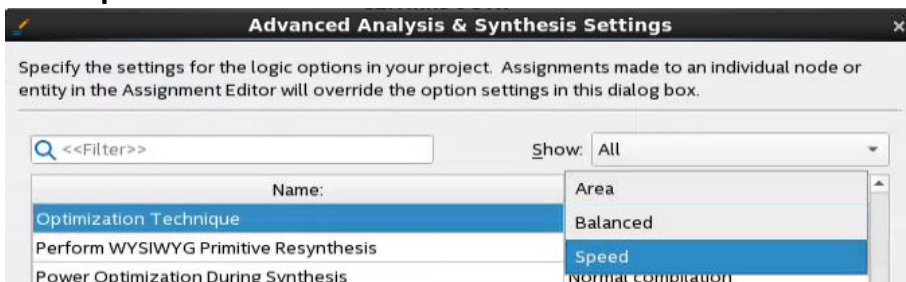
10.2. Optimizing Timing

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

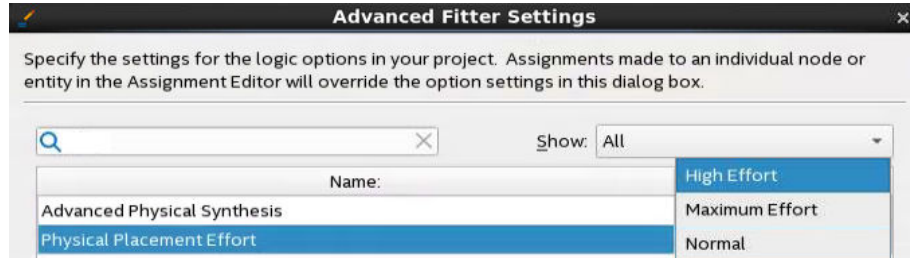
1. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings**. Under **Optimization mode**, select one of the **Performance** options.



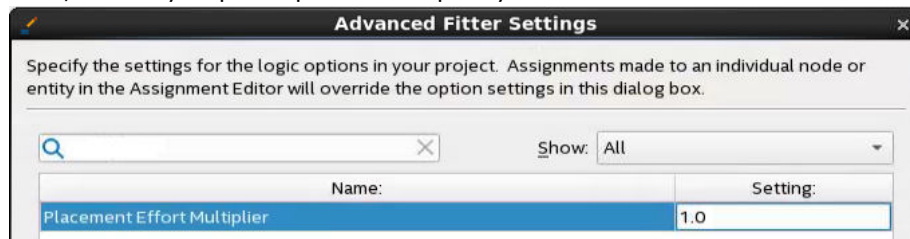
2. On the **Assignments** menu, click **Settings**. In the **Category** list, click **Compiler Settings** > **Advanced Settings (Synthesis)**. For **Optimization Technique**, select **Speed**.



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.





11. 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.

11.1. Optimizing Efficiency for Secondary Controller

You can encounter lower than expected efficiency for sequential access patterns on the secondary memory controller in the following EMIF configurations:

- 2ch x16 LPDDR4 / LPDDR5 / DDR5
- 4ch x16 LPDDR4 / LPDDR5
- 1ch x16 of LPDDR4 / LPDDR5 / DDR5 on the top sub-bank

Asynchronous Clocking Mode (Fabric Direct - User Clock Asynchronous to PHY)

You can improve the efficiency for sequential access pattern on the secondary controller by increasing the AXI clock frequency. Recommendation is to set the AXI clock frequency to 1/4 of memory clock frequency. You can further improve the efficiency of sequential write access pattern on the secondary controller by increasing the burst length as shown in the Table 1. The data shown in the table is obtained by using a 2ch x16 LPDDR5 design, running at 1066.667MHz.

Table 222. Controller Efficiency with Different AXI Clock Frequency and Burst Length

AXI Clock Frequency (MHz)	Pattern	Burst Length	Controller Efficiency (%)	
			Primary	Secondary
133.333 (Equivalent to Sync Mode)	Sequential write	2	~90	~40
	Sequential read		~90	~50
266.667	Sequential write	2	~90	~75
	Sequential read		~90	~90
	Sequential write	4	~90	~80
	Sequential read		~90	~90
	Sequential write	8	~90	~90
	Sequential read		~90	~90
	Sequential write	16	~90	~90
	Sequential read		~90	~90

Random access patterns or mixed traffic patterns can mitigate the differences in the efficiency for the primary controller and secondary controller.

Table 223. Controller Efficiency with Different AXI Clock Frequency for Random Traffic Pattern

AXI Clock Frequency (MHz)	Pattern	Burst Length	Controller Efficiency (%)	
			Primary	Secondary
133.333	Random write	2	~40	~39
	Random read		~38	~38
266.667	Random write	2	~41	~41
	Random read		~38	~38

With increased AXI clock frequency, the efficiency number reported by the PMON IP represents the utilization of the AXI bus instead of controller efficiency. The following equation represents the efficiency of the memory controller:

$$\begin{aligned} \text{Controller efficiency} &= \text{AXI transactions accepted by AXI bus} / \text{Memory Controller Bandwidth} * 100 \% \\ &= (\text{PMON efficiency} * \text{AXI Clock Freq} * 256\text{-bit}) / (16\text{-bit} * \text{Mem Clock Freq}) * 100\% \end{aligned}$$

Synchronous Clocking Mode (Fabric Direct – User Clock Synchronous to PHY)

In LPDDR4, LPDDR5 and DDR5 memory interface, when using x16 2-channel configuration for long burst sequential access during write or read operation, performance is limited to 50% per read/write sub-channel for the secondary controller.

If your application requires high controller efficiency for sequential access pattern, it is recommended to use asynchronous clocking mode. Random access patterns or mixed traffic patterns can mitigate the differences in the efficiency for the primary controller and secondary controller.

Note that in synchronous clocking mode, the efficiency for the primary controller used for implementing a x32 DDR5/LPDDR4/LPDDR5 memory interface has reduced performance. This is because the bandwidth for AXI bus is only half of the bandwidth for the memory controller.

AXI Bus Bandwidth = AXI Clock Freq x 256-bit = (Memory Clock Freq / 8) * 256 = 32 * Memory Clock Frequency

Memory Controller Bandwidth = Memory Clock Freq * 2 * 32-bit = 64 * Memory Clock Frequency = 2 * AXI Bus Bandwidth

11.2. 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.

11.3. 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.

11.4. 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.

11.5. Improving Controller Efficiency

You can use the following methods to improve the efficiency of your controller.

- Frequency of Operation
- Series of Reads or Writes
- AXI to Memory Mapping

The following sections discuss these methods in detail.

11.5.1. 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.

11.5.2. 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.

11.5.3. AXI to Memory Mapping

You can select which region of the memory (bank, bank group, row, column, and chip) to access by selecting the corresponding AXI address.

The default configuration is as follows:

Table 224. DDR4 4-Bank-Group Case

<MSB								LSB>
Chip Select	Chip ID	Row	Bank	Bank Group [1]	Column [N:3]	Bank Group [0]	Column [2:0]	Datapath

Table 225. DDR4 2-Bank-Group Case

<MSB							LSB>
Chip Select	Chip ID	Row	Bank	Column [N:3]	Bank Group [0]	Column [2:0]	Datapath

Table 226. DDR5 4-Bank-Group Case

<MSB							LSB>
Chip Select	Chip ID	Row	Bank	Column [N:4]	Bank Group [1:0]	Column [3:0]	Datapath

Table 227. DDR5 8-Bank-Group Case

<MSB								LSB>
Chip Select	Chip ID	Row	Bank	Bank Group [2]	Column [N:4]	Bank Group [1:0]	Column [3:0]	Datapath

Table 228. LPDDR5 x16 Case

<MSB								LSB>
Chip Select	Chip ID	Row	Bank	Bank Group [1]	Column [N:4]	Bank Group [0]	Column [3:0]	Datapath

Table 229. LPDDR5 x32

<MSB								LSB>
Chip Select	Chip ID	Row	Bank	Bank Group [1]	Column [N:4]	Bank Group [0]	Column [3:0]	Datapath

Datapath: For a x32 or wider interface, two bits are allocated to datapath. For a x16 interface, one bit is allocated; this bit should always be set to zero.

Row/Bank/Bank Group/Column: These are allocated a number of address bits based on the requirements of the connected memory type.

Chip Select/Chip ID: These are each one bit wide and are omitted when not using multi-rank or 3DS configurations.

Example: AXI Address mapping for a DDR4 1Gb x8 device, 2 components per rank for a total DQ Width of x16

You can obtain addressing tables from your memory device datasheet. For this example, the following information applies:

Parameter	1 Gb x 8
Bank Group	BG[1:0]
Bank Address	BA[1:0]
Rows	A[15:0]
Columns	A[9:0]

The following AXI Address to Memory Mapping table can be constructed.

Memory Address Bits	AXI Address Bits
Datapath [0]	[0] - Always 0, 1 bit for x16 interfaces
Column [2:0]	[3:1]
BG [0]	[4]
Column [9:3]	[11:5]
BG [1]	[12]
BA [1:0]	[14:13]
Row [15:0]	[30:15]

Using this information, the AXI Address to write to Column 1016 of the Row 0, Bank group 0, Bank 0 is: 0x00000FE0.



12. Agilex 5 FPGA EMIF IP – Debugging

This chapter discusses issues and strategies for debugging your external memory interface IP.

12.1. Interface Configuration Performance Issues

There are many interface combinations and configurations possible in an Altera design, therefore it is impractical for Altera to explicitly state the achievable f_{MAX} for every combination.

Altera 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 Altera 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). Altera performance figures are typically stated for the default (single rank, unbuffered DIMM) case. Altera provides additional expected performance data where possible, but the f_{MAX} is not achievable in all configurations. Altera 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)

12.1.1. 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.

12.2. 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 Altera 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.

12.2.1. Altera IP Memory Model

Altera 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 Altera-generated memory model may be suitable to evaluate some limited functional issues, but it does not provide comprehensive functional simulation.

12.2.2. 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: Altera 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.

12.2.3. 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 Altera 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 Altera 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 Altera 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.

12.3. 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.

12.3.1. Evaluating FPGA Timing Issues

Usually, you should not encounter timing issues with Altera-provided IP unless your design exceeds Altera'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.

12.3.2. 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 Altera 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.

12.4. 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**

12.5. Debugging with the External Memory Interface Debug Toolkit

The External Memory Interface Debug Toolkit for Agilex 5 FPGAs provides access to data collected by the sequencer during memory calibration, as well as analysis tools to evaluate the stability of the calibrated interface and assess hardware conditions.

The debug toolkit provides the following types of reports:

- Interface and memory configuration, such as external memory protocol and interface width.
- Calibration results, including calibration status (pass or fail), calibration failure stage (if applicable), delay settings and margins, and VREF settings and margins.

The available task and analysis capabilities include the following:

- Ability to request recalibration of the memory interface.
- Ability to run the test engine in the design example.
- Ability to view the delay setting on any pin in the selected interface and update it if necessary.
- Ability to run VREF margining on the interface.
- Ability to run driver margining on the interface.

Note: Because the HPS EMIF controller does not support the External Memory Interface Debug Toolkit, verify that the HPS memory interface is operational using the non-HPS memory controller first. Create a design that instantiates the FPGA memory controller, use parameters that will be used for the HPS memory interface, and route it to the same I/O that the HPS EMIF uses.

12.5.1. Prerequisites for Using the EMIF Debug Toolkit

You must complete certain prerequisites — including generating a design example — before you can use the EMIF Debug Toolkit.

Complete the following steps to enable the EMIF Debug Toolkit:

1. You must configure your design to use the EMIF Debug Toolkit, as described in the following topics.
2. You must compile your design.
3. You must program the target device with the resulting SRAM Object File (.sof).

After completing the above steps, you are ready to run the EMIF Debug toolkit.

Related Information

[Launching the EMIF Debug Toolkit](#) on page 248

12.5.2. Configuring a Design to Use the Debug Toolkit

Perform the following tasks to configure your design for use with the toolkit.

12.5.2.1. Generating a Design Example with the Debug Toolkit

To enable the Debug Toolkit in the design example, follow these steps:

1. Navigate to the **PHY** section of the **High-level Configuration** tab.
2. Select the **Use Debug Toolkit** checkbox.

High-level Configuration | **Advanced: Memory Timing** | Advanced: Analog Overrides | Example Design

Memory Device

PHY

☐ Auto-set PLL Reference Clock Frequency

☐ Enable Advanced List of PLL Reference Clock Frequencies

Reference Clock Frequency: 100.0

AC Placement: AC Bottom Sub-bank (lanes 0-3)

Alert_n AC-Lane Index: AC2

☐ Force Using 4 AC Lanes

☐ Auto-set Mainband Access Mode

Mainband Access Mode: Fabric Direct - User Clock Asynchronous to PHY

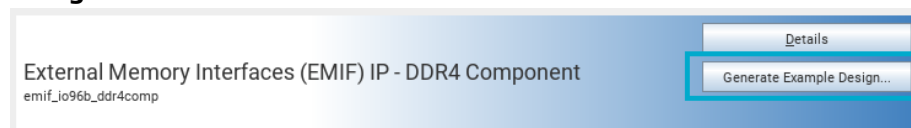
Sideband Access Mode: Fabric Direct

Pin Swizzle Map: BYTE_SWIZZLE_CH0=0,X,X,X,1,2,3,X; PIN_SWIZZLE_CH0_DQS0=3,1,7,5,0,4

☒ **Use Debug Toolkit**

Instance ID: 0

- After you have fully parameterized the interface, click **Generate Example Design**.



The generated design example has the debug toolkit enabled and all the necessary components wired up, as required for a single interface.

Related Information

[External Memory Interfaces \(EMIF\) IP Design Example User Guide: Agilex 5 FPGAs and SoCs](#)

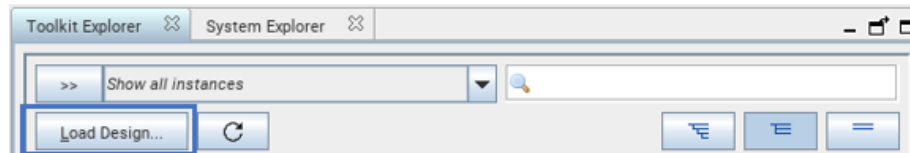
Design example and quick-start information for the Agilex 5 EMIF IP.

12.5.3. Launching the EMIF Debug Toolkit

Before launching the EMIF Debug Toolkit, ensure that you have configured your device with a programming file that has the EMIF Debug Toolkit enabled.

To launch the debug toolkit, follow these steps:

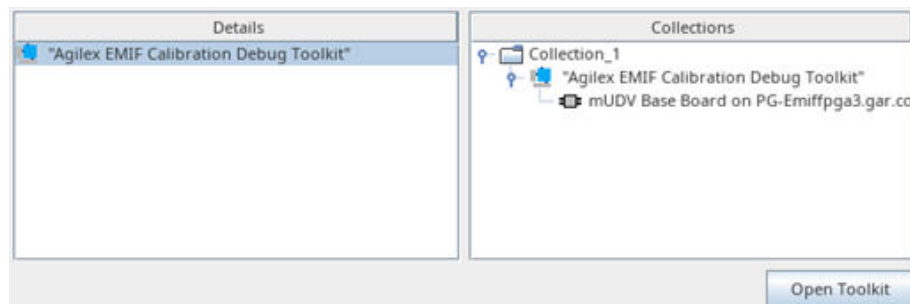
- In the Quartus Prime software, open the System Console by clicking **Tools > System Debugging Tools > System Console**.
- In the System Console, load the SRAM Object File (.sof) with which you have programmed the board. (Refer to the [Prerequisites for Using the EMIF Debug Toolkit](#) topic.)



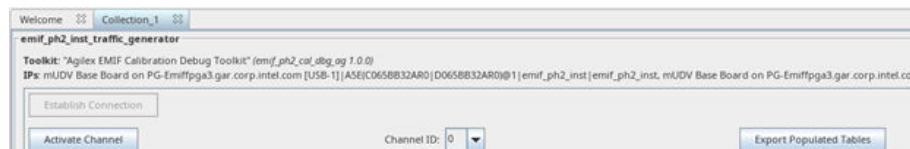
3. Select the correct instance that you want to debug.



4. Select **Agilex 7 EMIF Calibration Debug Toolkit** from the **Details** section. Click **Open Toolkit** to open the main view of the toolkit.



5. Click **Establish Connection** to initialize the toolkit.
6. Click **Activate Channel** to allow the toolkit to read the parameters and status for the selected interface, and to perform analysis tasks.



7. Successful activation of the Debug Toolkit is indicated by a message in the Messages tab.



Related Information

Prerequisites for Using the EMIF Debug Toolkit on page 247

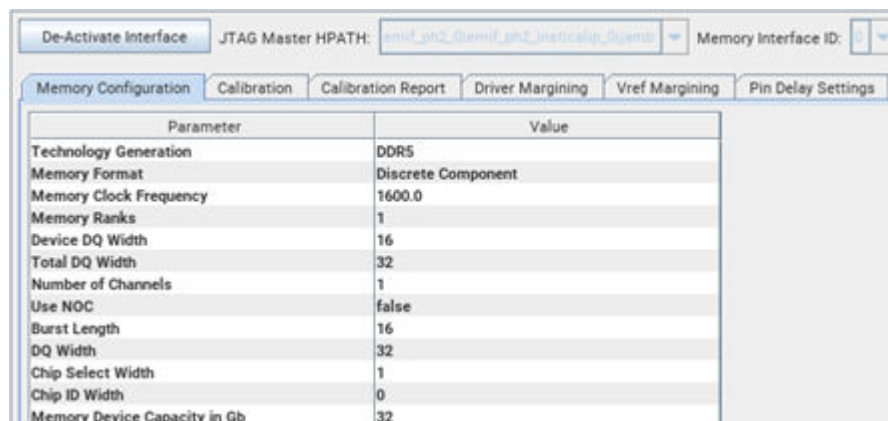
12.5.4. Using the EMIF Debug Toolkit

The main view of the EMIF Debug Toolkit contains the **Memory Configuration**, **Calibration**, **Calibration Report**, **Driver Margining**, **VREF Margining**, and **Pin Delay Settings** tabs.

Memory Configuration Tab

The **Memory Configuration** tab shows the IP settings, which you defined when you parameterized the EMIF IP.

Figure 36. Memory Configuration Tab



Parameter	Value
Technology Generation	DDR5
Memory Format	Discrete Component
Memory Clock Frequency	1600.0
Memory Ranks	1
Device DQ Width	16
Total DQ Width	32
Number of Channels	1
Use NOC	false
Burst Length	16
DQ Width	32
Chip Select Width	1
Chip ID Width	0
Memory Device Capacity in Gb	32

Calibration Tab

The **Calibration** tab allows you to run re-calibration and the test engine.

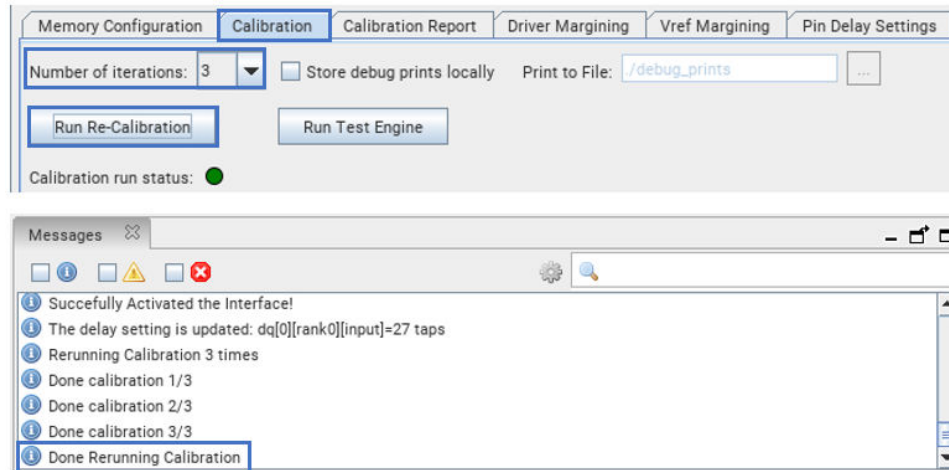
12.5.4.1. Running Re-Calibration

The **Calibration** tab lets you specify a number of iterations by which to rerun calibration.

To rerun calibration, follow these steps:

1. Select the desired number of iterations from the **Number of Iterations** pull-down menu.
2. Click **Run Re-Calibration** to repeat calibration the specified number of times. The system reports **Done Rerunning Calibration** in the **Messages** window upon completion.

A green dot in the **Calibration run status** indicator signifies that the calibration passed for all the iterations, while a red dot indicates that the calibration failed in at least one of the iterations.



Note: For multichannel interfaces, recalibrating an interface resets all channels associated with that EMIF.

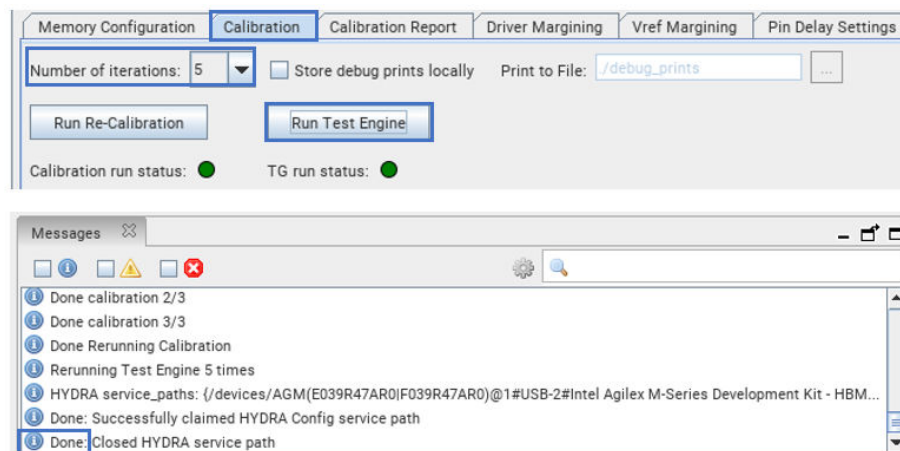
12.5.4.2. Running the Test Engine

You can use the debug toolkit to run the Test Engine.

To run the Test Engine, follow these steps:

1. Select the desired number of iterations that you want to rerun the test engine from the **Number of Iterations** pulldown menu.
2. Click **Run Test Engine** to run the traffic generator a specified number of times. A **Done** status on the **Messages** window indicates a complete run.

A green dot in the **Calibration run status** indicator signifies a pass for all iterations of the test engine, while a red dot indicates a failure in at least one iteration.



12.6. 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_a1`)
- Single write and read (with `AxLEN=axlen_b2`)
- Sequential address³ block of 512 writes and 512 reads (with `AxLEN=axlen_a1`)
- Sequential address³ block of 512 writes and 512 reads (with `AxLEN=axlen_b2`)
- Random address⁴ block of 512 writes and 512 reads (with `AxLEN=axlen_a1`)

- 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.

12.7. Guidelines for Developing HDL for Traffic Generator

If you are not getting the expected response on the AXI bus when using your own traffic generator to test your EMIF IP on hardware, ensure that your traffic generator meets the following guidelines.

1. The traffic generator issues transactions only after calibration has completed successfully. You can check the calibration status by using the AXI-Lite interface. In the EMIF example design, the `cal_done_rst_n` port on the `ed_synth_axil_driver_0` corresponds to the calibration status. A value of `cal_done_rst_n=1` indicates that the calibration has completed and passed. Your traffic generator can begin to issue AXI-compliant transactions only after `cal_done_rst_n=1`.

Figure 37. cal_done_rst_n in ed_synth_axil_driver_0

```

149 ed_synth_axil_driver_0 axil_driver_0 (
150     .axil_driver_clk      (user_pll_outclk1_clk),
151     .axil_driver_rst_n    (~rst_controller_reset_out_reset),
152     .axil_driver_awaddr   (axil_driver_0_axil_driver_axi4_lite_awaddr),
153     .axil_driver_awvalid  (axil_driver_0_axil_driver_axi4_lite_awvalid),
154     .axil_driver_awready  (axil_driver_0_axil_driver_axi4_lite_awready),
155     .axil_driver_wdata    (axil_driver_0_axil_driver_axi4_lite_wdata),
156     .axil_driver_wstrb    (axil_driver_0_axil_driver_axi4_lite_wstrb),
157     .axil_driver_wvalid   (axil_driver_0_axil_driver_axi4_lite_wvalid),
158     .axil_driver_wready   (axil_driver_0_axil_driver_axi4_lite_wready),
159     .axil_driver_bresp    (axil_driver_0_axil_driver_axi4_lite_bresp),
160     .axil_driver_bvalid   (axil_driver_0_axil_driver_axi4_lite_bvalid),
161     .axil_driver_bready   (axil_driver_0_axil_driver_axi4_lite_bready),
162     .axil_driver_araddr   (axil_driver_0_axil_driver_axi4_lite_araddr),
163     .axil_driver_arvalid  (axil_driver_0_axil_driver_axi4_lite_arvalid),
164     .axil_driver_arready  (axil_driver_0_axil_driver_axi4_lite_arready),
165     .axil_driver_rdata    (axil_driver_0_axil_driver_axi4_lite_rdata),
166     .axil_driver_rresp    (axil_driver_0_axil_driver_axi4_lite_rresp),
167     .axil_driver_rvalid   (axil_driver_0_axil_driver_axi4_lite_rvalid),
168     .axil_driver_rready   (axil_driver_0_axil_driver_axi4_lite_rready),
169     .axil_driver_awprot   (axil_driver_0_axil_driver_axi4_lite_awprot),
170     .axil_driver_arprot   (axil_driver_0_axil_driver_axi4_lite_arprot),
171     .cal_done_rst_n       (axil_driver_0_cal_done_rst_n_reset)
172 );

```

12.8. Guidelines for Traffic Generator Status Check

The Test Engine IP allows you to observe traffic generator status through the Signal Tap Logic Analyzer and allows you to export status interface so that you can observe the signals through top level design ports or access through user logic.

12.8.1. Status Check Using the Signal Tap Logic Analyzer

When you generate an Agilex 5 FPGA EMIF IP design example, you can specify the traffic pattern to run. Select the required setting, as illustrated in the figure below.

Figure 38. Selecting Traffic Generator Pattern

Traffic Generator	
Traffic Generator Remote Access:	Remote through JTAG
Traffic Generator Program:	Infinite Traffic Pattern
Performance Monitor	Short Traffic Pattern Medium-length Traffic Pattern Long Traffic Pattern Infinite Traffic Pattern

Refer to the *External Memory Interfaces (EMIF) IP Design Example User Guide: Agilex 5 FPGAs and SoCs* for more information.

Observing Status with the Signal Tap Logic Analyzer

Follow these steps to observe status using the Signal Tap Logic Analyzer:

1. Generate a Signal Tap Logic Analyzer file, which includes traffic generator status signals.

Figure 39.

Type	Alias	Tap	Node	Data Enable	Trigger Enable	Trigger Conditions
			Name	639	639	1 ✓ Basic OR
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_araddr[26.0]	✓	✓	XXXXXXXXh (OR)
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_arvalid	✓	✓	
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_awaddr[26.0]	✓	✓	XXXXXXXXh (OR)
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_awvalid	✓	✓	
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_rdata[31.0]	✓	✓	XXXXXXXXh (OR)
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_wdata[31.0]	✓	✓	XXXXXXXXh (OR)
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_rdata[255.0]	✓	✓	EEEEEEEEEEEEEEEEEEEE
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_wdata[255.0]	✓	✓	EEEEEEEEEEEEEEEEEEEE
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_rready	✓	✓	
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_wready	✓	✓	
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_rvalid	✓	✓	
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_wvalid	✓	✓	
*		Pre-Syn	axil_driver_0_cal_done_rst_n_reset	✓	✓	
*		Pre-Syn	traffic_generator hydra_inst global_csr status_done	✓	✓	
*		Pre-Syn	traffic_generator hydra_inst global_csr status_error	✓	✓	

Note: The `status_done` and `status_error` ports on the `traffic_generator|hydra_inst|global_csr` instance are the traffic generator status signals.

2. Save and enable the Signal Tap file in the project, and recompile the design.
3. Configure the device with your `.sof` file and observe the signals status through signal tap file.
4. Signal values of `status_done=1` and `status_error=0` indicate that the traffic test completed with no traffic errors.

Figure 40.

Type	Alias	Tap	Name	-256	-128	0	128	256	384
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_araddr[26.0]						
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_arvalid						
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_awaddr[26.0]						
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_awvalid						
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_rdata[31.0]						
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4lite_wdata[31.0]						
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_rdata[255.0]						
g		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_wdata[255.0]						
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_rready						
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_wready						
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_rvalid						
*		Pre-Syn	emif_io96b_ddr4comp_0[s0_axi4_wvalid						
*		Pre-Syn	axil_driver_0_cal_done_rst_n_reset						
*		Pre-Syn	traffic_generator hydra_inst global_csr status_done						
*		Pre-Syn	traffic_generator hydra_inst global_csr status_error						

Note: For an infinite traffic pattern program, you will not see `status_done=1`, because the traffic continues running. Set the trigger conditions at the rising edge of the `status_error` signal and let the traffic run. Any failure in traffic is detected if the `status_error` signal goes high.

12.8.2. Exporting the Status Interface to the Top-Level Design

Do not perform this procedure if you plan to observe signal activity through the Signal Tap Logic Analyzer only.

1. Navigate to the **IP Components** tab in the **Project Navigator**, and click `ed_synth.qsys` to open the design in the Platform Designer.

Project Navigator					
Entity	IP Component	Version	Supported Device Families	IP File	
ed_synth	<Platform Designer>		Agilx 5	ed_synth.qsys	
ed_synth_axil_driver_0	<Platform Designer>		Agilx 5	ip/ed_synth/ed_synth_axil_driver_0.ip	
ed_synth_emif_ph2_0	External Memory Interfaces (EMIF) IP	6.1.0	Agilx 5	ip/ed_synth/ed_synth_emif_ph2_0.ip	
ed_synth_perf_monch0_0	Performance Monitor Intel FPGA IP	1.0.0	Agilx 5	ip/ed_synth/ed_synth_perf_monch0.ip	
ed_synth_reset_handler	<Platform Designer>		Agilx 5	ip/ed_synth/ed_synth_reset_handler.ip	
ed_synth_rrip	Reset Release Intel FPGA IP	19.4.7	Agilx 5	ip/ed_synth/ed_synth_rrip.ip	
ed_synth_traffic_generator	Test Engine Intel FPGA IP	1.0.0	Agilx 5	ip/ed_synth/ed_synth_traffic_generator.ip	
ed_synth_user_pll	IOPLL Intel FPGA IP	19.3.1	Agilx 5	ip/ed_synth/ed_synth_user_pll.ip	

- | Name | Description | Export | Clock | Base |
|----------------------|--------------------------------------|--------------------------|-----------------|-------------|
| emul_axi2_0 | External Memory Interfaces (EMIP) IP | | | |
| ref_clk_0 | Clock Input | ref_clk_0 | exported | |
| core_mem_0 | Reset Input | Double-click to export | | |
| user_async_clk_0 | Clock Input | Double-click to export | user_pll_out... | |
| user_pll_out... | Reset Output | Double-click to export | user_async... | |
| axi_mem4 | AXI4 Subordinate | Double-click to export | | 0x0 |
| mem_0 | Conduit | emul_pdt2_0_mem_0 | | |
| oct_0 | Conduit | emul_pdt2_0_oct_0 | | |
| axi_async_clk | Clock Input | Double-click to export | user_pll_out... | |
| axi_mem2_0 | Reset Input | Double-click to export | | |
| axi_mem | AXI4-Lite Subordinate | Double-click to export | [axi_async_clk] | 0x0000_0000 |
| axi_driver_0 | EMIP AXI-Lite Driver IP | | | |
| axi_driver_clk | Clock Input | Double-click to export | user_pll_out... | |
| axi_driver_status | Reset Input | Double-click to export | axi_driver... | |
| axi_driver_axi4_lite | AXI4-Lite Manager | Double-click to export | axi_driver... | |
| axi_done_hdl_n | Reset Output | Double-click to export | | |
| traffic_generator | Test Engine Intel FPGA IP | | | |
| remote_mif_clk | Clock Input | Double-click to export | user_pll_out... | |
| remote_mif_reset | Reset Input | Double-click to export | | |
| tag_reset | Reset Output | Double-click to export | | |
| status | Conduit | traffic_generator_status | | |
| driver0_axi4 | AXI4 Manager | Double-click to export | [driver0_clk] | |
| driver0_clk | Clock Input | Double-click to export | user_pll_out... | |

-  Sync System Infos  Auto  Validate System Integrity  Generate HDL...

- 
- [Send Feedback](#)

Figure 44.



12.9. 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.

12.9.1. 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.

12.9.2. 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.

12.9.3. 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.

12.9.4. 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.

12.9.5. 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

12.9.6. 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.

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

Designs are often tested using previous generations of Altera 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 Altera 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 Altera 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 Altera.

In all of the above scenarios, you must determine if the issue still exists in the latest version of the Altera software. Bug fixes and enhancements are added to the Altera 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.

12.9.8. Try A Different PCB

If you are using the same Altera 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.

12.9.9. 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.

12.9.10. Debugging Checklist

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

Table 230. Checklist

Check	Item
<input type="checkbox"/>	Try a different fit.
<input type="checkbox"/>	Check IP parameters at the operating frequency (t_{MRD} , t_{WTR} for example).
<input type="checkbox"/>	Ensure you have constrained your design with proper timing deration and have closed timing.
<input type="checkbox"/>	Simulate the design. If it fails in simulation, it will fail in hardware.
<input type="checkbox"/>	Analyze timing.
<input type="checkbox"/>	Place and assign R_{ZQ} (OCT).
<input type="checkbox"/>	Measure the power distribution network (PDN).
<input type="checkbox"/>	Measure signal integrity.
<input type="checkbox"/>	Measure setup and hold timing.
<input type="checkbox"/>	Measure FPGA voltages.
<input type="checkbox"/>	Vary voltages.
<input type="checkbox"/>	Heat and cool the PCB.
<input type="checkbox"/>	Operate at a lower or higher frequency.
<input type="checkbox"/>	Check board timing and trace Information.
<i>continued...</i>	

Check	Item
<input type="checkbox"/>	Check True Differential Signaling and clock sources, I/O voltages and termination.
<input type="checkbox"/>	Check PLL clock source, specification, and jitter.
<input type="checkbox"/>	Retarget to a smaller interface width or a single bank.

12.10. 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

12.10.1. 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.

12.10.1.1. 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.

12.10.1.2. 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.

12.10.1.3. 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 Altera FPGA IP matches that actually achieved on the PCB, otherwise Quartus Prime timing analysis of the interface is accurate.

12.10.1.4. 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.

12.10.1.5. 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.

12.10.1.6. 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.

12.10.1.7. 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.

12.10.1.8. 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.

12.10.1.9. 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.

12.10.2. 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.

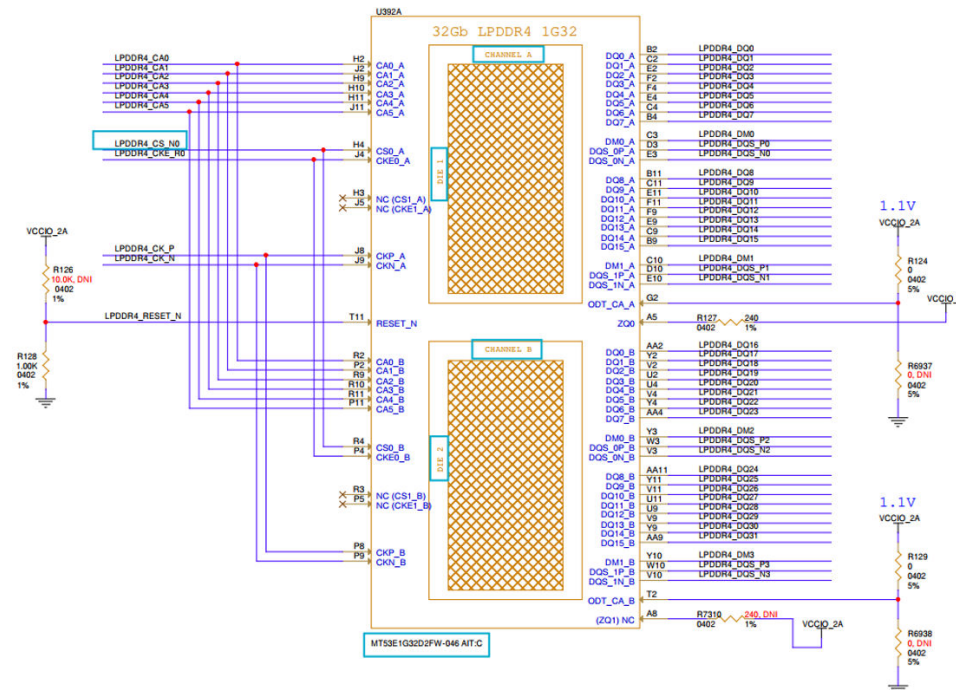
12.10.2.1. Verifying High-Level Configuration

Memory Device

Verify that the correct memory *Number of Channels*, *Data DQ Width*, *Die Density*, and *CS width* are selected for the memory configuration used. Inaccurate information can result in calibration failure or traffic failure.

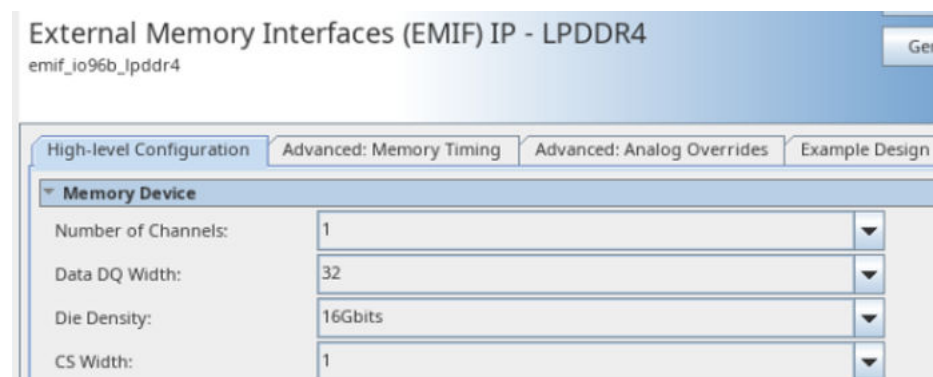
The example below shows memory parameterization based on an Agilex 5 FPGA E-Series 065B Premium Development Kit targeting an MT53E1G32D2FW LPDDR4 EMIF configuration:

Figure 45. Example of MT53E1G32D2FW LPDDR4 EMIF Configuration



- Number of Channels: 1. (A/C pins are shared between CH_A and CH_B)
- Data DQ Width: 32. (DQ pins per memory channel. Die 1 and Die 2 implemented as single x32 interface.)
- Die Density: 16 Gbits. (Select capacity of each memory die (in Gbits), not total memory density per interface.)
- CS width: 1. (Above topology is single-rank per channel.)

Figure 46. Example EMIF IP High Level Configuration



Pin Swizzle Map

The Quartus Prime software uses fixed pin placement for each supported memory standard, as described in the [Pin Placement](#) topic in the *Architecture* chapter. Do not make changes using a .qsf assignment or the pin planner if you need to swap the DQ pins within a DQS group or the DQS group within a channel/interface. Fitter errors may occur if you enter the incorrect pin location for the EMIF interface. The following is an example of a fitter error for LPDDR4 protocol when an incorrect pin assignment is specified:

```
Error (175001): The Fitter cannot place 1 BYTE, which is within Generic
Component ed_synth_emif_io96b_lpddr4_0.
```

The Pin Swizzle Map allows you to specify the ordering of the physical connections of each DQ pin on the memory component to the FPGA. Incorrect information entered in the Pin Swizzle Map may cause calibration errors.

Refer to the [External Memory Interfaces \(EMIF\) IP Design Example User Guide: Agilex 5 FPGAs and SoCs](#) for details on configuring DQ pin swizzling.

Figure 47. Example of a Pin Swizzle Map Entry

Pin Swizzle Map for Component interfaces

For component interfaces, you must match the net name in the schematic to the index of default pin placement, to determine the pin swizzling specification.

Consider the example of DQS Group 0 for Channel 1 on the LPDDR4 Component interface on the Agilex 5 FPGA E-Series 065B Premium Development Kit targeting MT53E1G32D2FW LPDDR4 memory, as shown in the table below:

Table 231. Example of Pin Swizzle Targeting a Component-type Memory Interface

Lane	Pin	Pin Index	Default Pin Placement	Net Name in Schematic	Swizzling Index
BL6	CA59	83	MEM_0_MEM_DQ [23]	MEM_1_MEM_DQ [11]	11
	BW59	82	MEM_0_MEM_DQ [22]	MEM_1_MEM_DQ [10]	10
	BR59	81	MEM_0_MEM_DQ [21]	MEM_1_MEM_DQ [8]	8
	BU59	80	MEM_0_MEM_DQ [20]	MEM_1_MEM_DQ [9]	9
continued...					

Lane	Pin	Pin Index	Default Pin Placement	Net Name in Schematic	Swizzling Index
	GR62	79			
	BU62	78	MEM_0_MEM_DMI [2]	MEM_0_MEM_DMI [1]	
	CA69	77	MEM_0_MEM_DQS_C [2]	MEM_0_MEM_DQS_C [1]	
	BW69	76	MEM_0_MEM_DQS_T [2]	MEM_0_MEM_DQS_T [1]	
	BU71	75	MEM_0_MEM_DQ [19]	MEM_1_MEM_DQ [12]	12
	BR71	74	MEM_0_MEM_DQ [18]	MEM_1_MEM_DQ [14]	14
	BU69	73	MEM_0_MEM_DQ [17]	MEM_1_MEM_DQ [13]	13
	BR69	72	MEM_0_MEM_DQ [16]	MEM_1_MEM_DQ [15]	15

For this specific group, the correct pin swizzling specification is:

```
PIN_SWIZZLE_CH0_DQS1 = 15,13,14,12,9,8,10,11;
```

The following example shows pin swizzle map parameters based on an Agilex 5 FPGA E-Series 065B Premium Development Kit targeting LPDDR4 EMIF. You must add all parameters to the pin swizzle map in a single line, each separated by semi-colons as shown:

```
BYTE_SWIZZLE_CH0=3,2,X,X,X,X,1,0;  
PIN_SWIZZLE_CH0_DQS1=15,13,14,12,9,8,10,11;  
PIN_SWIZZLE_CH0_DQS0=3,2,1,0,5,4,7,6;  
PIN_SWIZZLE_CH0_DQS3=31,30,28,29,25,24,26,27;  
PIN_SWIZZLE_CH0_DQS2= 16,18,17,19,23,20,22,21;
```

12.10.2.2. Verifying Memory Timing Parameters

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.

12.10.2.3. Verifying 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.

12.10.2.4. Debugging Intermittent Issues

Intermittent failures can be challenging to isolate. In such cases, it is advisable to return to general debugging to better understand the symptoms. Use the [Debugging Checklist](#) as a starting point to narrow down the issue.

13. Agilex 5 FPGA EMIF IP - Mailbox Support

This chapter discusses mailbox support.

13.1. Agilex 5 Mailbox Structure and Register Definitions

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

You can access the mailbox using the AXI-lite interface or through JTAG Avalon Master Bridge (JAMB) when you enable the debug toolkit option in the EMIF example design.

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.

The base address for the mailbox is 0x0500_0000.

Table 232. Mailbox Structure

Register Name	Byte Offset (Decimal)	Width (bits)	Access	Description
CMD_PARAM_6	1056	32	RW	[Input] This register specifies the seventh parameter (if applicable) for the requested command.
CMD_PARAM_5	1060	32	RW	[Input] This register specifies the sixth parameter (if applicable) for the requested command.
CMD_PARAM_4	1064	32	RW	[Input] This register specifies the fifth parameter (if applicable) for the requested command.
CMD_PARAM_3	1068	32	RW	[Input] This register specifies the fourth parameter (if applicable) for the requested command.
CMD_PARAM_2	1072	32	RW	[Input] This register specifies the third parameter (if applicable) for the requested command.
CMD_PARAM_1	1076	32	RW	[Input] This register specifies the second parameter (if applicable) for the requested command.
CMD_PARAM_0	1080	32	RW	[Input] This register specifies the first parameter (if applicable) for the requested command.
continued...				

Register Name	Byte Offset (Decimal)	Width (bits)	Access	Description
CMD_REQ	1084	32	RW	<i>[Input]</i> This register specifies the command to be performed and the target IP type and identifier.
CMD_RESPONSE_DATA_2	1104	32	RO	<i>[Output]</i> 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	<i>[Output]</i> 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	<i>[Output]</i> 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	<i>[Output]</i> 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 233. 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.	RO	0x0
continued...				

Bit	Name	Description	Access	Reset
		'0' One or more EMIF instances in the IO96 have either failed to calibrate or have not completed calibration yet.		

Table 234. CMD_RESPONSE_STATUS Register

Bit	Name	Description	Access	Reset
[31:16]	CMD_RESPONSE_DATA_SHORT	For commands that return up to 16-bit value, this field contains the requested response data.	RO	0x0
[15:8]	Reserved			
[7:5]	STATUS_CMD_RESPONSE_ERROR	Indicates which errors occurred while processing the command, which are specific to the command being executed. '000' - No errors. Other error codes will be outlined alongside the commands to which they are associated.	RO	0x0
[4:1]	STATUS_GENERAL_ERROR	Reports general errors that occurred while a command was being executed: '0000' - no error. '0001' - Indicates that an illegal command was received. '0010' - indicates that the interface the command is targeting doesn't exist. '0011' - indicates that the conditions for executing the command are not met. '0100' - indicates an unexpected command for 2-stage commands like BIST_MEM_INIT_START . '0110' - indicates that the specified interface is busy.	RO	0x0
[0]	STATUS_COMMAND_RESPONSE_READY	Tells whether the command response is ready. '1' - Indicates that the command response is ready.	RW	0x0

13.1.1. Mailbox Supported Commands

Read-Only Registers

The following read-only registers provide direct access to retrieve information. Note that these registers are not write-protected; you should not attempt to write to any of these registers. Additionally, you should avoid reading from registers that are not used in the target design, as doing so might result in unpredictable or invalid data being returned.

Table 235. Read-Only Registers

Register Name	Byte Offset (Hexadecimal)	Description
MAILBOX_HEADER	0x000	[Output] Mailbox Header This register specifies information about the mailbox protocol. [Fields] [2:0] MB_SPEC_VER: This field specifies the version of mailbox specification being used by the IOSSM firmware.
continued...		

Register Name	Byte Offset (Hexadecimal)	Description
		Current version number is 1.
MEM_INTF_INFO_0	0x200	[Output] Memory Interface Information This register specifies the memory interface IP type and instance ID for the interface 0/1 in the IO96B. It will only return the primary interface of each channel. [Fields] [31:29] IP_TYPE: Indicates the type of IP: <ul style="list-style-type: none"> 0x0 – Not used. 0x1 – Primary MC of Primary IO96B. 0x2 – Secondary MC of Primary IO96B. 0x3 – Primary MC of Secondary IO96B. 0x4 – Secondary MC of Secondary IO96B. [28:24] INSTANCE_ID: IP identifier.
MEM_INTF_INFO_1	0x280	
MEM_TECHNOLOGY_INTF0	0x210	[Output] Memory Technology This register provides the memory technology type for the memory interface 0/1. [Fields] [2:0] TECH: Reports the memory technology type: <ul style="list-style-type: none"> 0x0 = DDR4 0x1 = DDR5 0x2 = DDR5_RDIMM 0x3 = LPDDR4 0x4 = LPDDR5
MEM_TECHNOLOGY_INTF1	0x290	
MEMCLK_FREQ_INTF0	0x220	[Output] Memory Clock Frequency - This register reports the memory clock frequency in kilohertz (KHz) for the memory interface 0/1. For protocols with multiple FSPs, it reports the current memory clock frequency in kilohertz (KHz). [Fields] [31:0] FREQ_KHZ_FSP_CUR: Current memory clock frequency in KHz.
MEMCLK_FREQ_INTF1	0x2A0	
MEMCLK_FREQ_FSP0_INTF0	0x224	[Output] Memory Clock Frequency - Frequency Set Point 0 This register reports the memory clock frequency in kilohertz (KHz) for Frequency Set Point 0 of the memory interface 0/1. Only valid for LPDDR5 protocol. [Fields] [31:0] FREQ_KHZ_FSP0: Memory clock frequency in KHz for Frequency Set Point 0.
MEMCLK_FREQ_FSP0_INTF1	0x2A4	
MEMCLK_FREQ_FSP1_INTF0	0x228	[Output] Memory Clock Frequency - Frequency Set Point 1 This register reports the memory clock frequency in kilohertz (KHz) for Frequency Set Point 1 of the memory interface 0/1. Only valid for LPDDR5 protocol. [Fields] [31:0] FREQ_KHZ_FSP1: Memory clock frequency in KHz for Frequency Set Point 1.
MEMCLK_FREQ_FSP1_INTF1	0x2A8	
MEMCLK_FREQ_FSP2_INTF0	0x22C	MEMCLK_FREQ_INTF0_FSP2 [Output] Memory Clock Frequency - Frequency Set Point 2 This register reports the memory clock frequency in kilohertz (KHz) for Frequency Set Point 2 of the memory interface 0/1. Only valid for LPDDR5 protocol. [Fields] [31:0] FREQ_KHZ_FSP2: Memory clock frequency in KHz for Frequency Set Point 2.
MEMCLK_FREQ_FSP2_INTF1	0x2AC	
MEM_WIDTH_INFO_INTF0	0x230	[Output] Memory Width Information
MEM_WIDTH_INFO_INTF1	0x2B0	

continued...

Register Name	Byte Offset (Hexadecimal)	Description
		This register provides the memory width information for the memory interface 0/1. [Fields] [23:16] C_WIDTH: Channel width. [15:8] CS_WIDTH: Chip select width. [7:0] DQ_WIDTH: Data width
MEM_TOTAL_CAPACITY_INTF0	0x234	[Output] Total Memory Capacity
MEM_TOTAL_CAPACITY_INTF1	0x2B4	This register reports the total memory capacity per channel in gigabits (Gb) for the memory interface 0/1. [Fields] [7:0] CAPACITY_GBITS: Total memory device capacity per channel in Gb. Total memory capacity is calculated as: $CAPACITY = (DQ_WIDTH / DEVICE_WIDTH) * NUM_RANKS * C_WIDTH * DEVICE_DENSITY$
ECC_ENABLE_INTF0	0x240	[Output] ECC Enable Status
ECC_ENABLE_INTF1	0x2C0	This register provides detailed information regarding the ECC enable status and type for the memory interface 0/1. [Fields] [18:18] RD_LINK_ECC_ENABLED: Indicates the Read Link-ECC enable status. <ul style="list-style-type: none"> 1'b1 – Read Link-ECC is enabled. 1'b0 – Read Link-ECC is disabled. [17:17] WR_LINK_ECC_ENABLED: Indicates the Write Link-ECC enable status. <ul style="list-style-type: none"> 1'b1 – Write Link-ECC is enabled. 1'b0 – Write Link-ECC is disabled. [16:16] LINK_ECC_SUPPORTED: Indicates if Link-ECC is supported by the interface. <ul style="list-style-type: none"> 1'b1 – Link-ECC is supported. 1'b0 – Link-ECC is not supported. [8:8] ECC_TYPE: Specifies the ECC operational mode. <ul style="list-style-type: none"> 1'b0 – Out-of-Band ECC. 1'b1 – In-line ECC. [1:0] ECC_ENABLE_TYPE: Reports the current ECC error reporting and correcting status. <ul style="list-style-type: none"> 1'b00 – ECC is disabled. Data will be written and returned without ECC verification. 1'b01 – ECC is enabled without error detection or correction. 1'b10 – ECC is enabled with error detection; errors are reported but not corrected. 1'b11 – ECC is enabled with error detection and correction; single-bit errors are automatically corrected.
ECC_SCRUB_STATUS_INTF0	0x244	[Output] ECC Scrub Status
ECC_SCRUB_STATUS_INTF1	0x2C4	This register indicates the ECC scrub operation status for the memory interface 0/1. [Fields] [1:1] ECC_SCRUB_IN_PROGRESS: Reports whether an ECC scrub operation is currently in progress. <ul style="list-style-type: none"> 1'b0 – Not actively performing a scrubbing operation. 1'b1 – The Controller is in the process of performing a scrubbing operation. [0:0] ECC_SCRUB_STATUS: The ECC scrub operation initiated using ECC_SCRUB_MODE_0/1_START mailbox command has completed. <ul style="list-style-type: none"> 1'b0 – Scrubbing operation has not completed. 1'b1 – Scrubbing operation has completed.
continued...		

Register Name	Byte Offset (Hexadecimal)	Description
LP_MODE_INTF0	0x250	<p>[Output] Low Power Mode Status</p> <p>This register provides the current low power state and its validity for the memory interface 0/1.</p> <p>[Fields]</p> <p>[5:0] LP_STATE: Current Interface Low Power State.</p> <ul style="list-style-type: none"> 1'b000000 – Idle. 1'b000001 – Active Power Down. 1'b000010 – Active Power Down with Memory Clock Gating. 1'b000011 – Pre-Charge Power Down. 1'b000100 – Pre-Charge Power Down with Memory Clock Gating. 1'b000101 – Self-Refresh Short. 1'b000110 – Self-Refresh Short with Memory Clock Gating. 1'b001000 – Self-Refresh Long. 1'b001001 – Self-Refresh Long with Memory Clock Gating. 1'b001010 – Self-Refresh Long with Memory Clock and Controller Clock Gating. 1'b001011 – Self-Refresh Power Down Short. 1'b001100 – Self-Refresh Power Down Short with Memory Clock Gating. 1'b001101 – Self-Refresh Power Down. 1'b001110 – Self-Refresh Power Down Long with Memory Clock Gating. 1'b001111 – Self-Refresh Power Down Long with Memory and Controller Clock Gating. <p>[6:6] LP_STATE_VALID: Indicates whether the data for the low power state is valid.</p> <ul style="list-style-type: none"> 1'b1 – Valid. The low power state information is reliable. 1'b0 – Invalid. The interface is currently transitioning into or out of a low power state.
LP_MODE_INTF1	0x2D0	
MEM_INIT_STATUS_INTF0	0x260	<p>[Output] BIST Memory Initialization Status</p> <p>This register holds the status of the BIST memory initialization operation for the memory interface specified by the instance ID. The value of this register is only valid if the mailbox command BIST_MEM_INIT_START was issued previously.</p> <p>[Fields]</p> <p>[0:0] MEM_INIT_STATUS: Indicates the status of the BIST memory content initialization operation.</p> <ul style="list-style-type: none"> 1'b0 – Memory content initialization operation still in progress if previously initiated. 1'b1 – Memory content initialization operation has been completed.
MEM_INIT_STATUS_INTF1	0x2E0	
BIST_STATUS_INTF0	0x264	<p>[Output] BIST Results Status</p> <p>This register contains BIST results for the previously initiated BIST operation for memory interface 0/1</p> <p>[Fields]</p> <p>[31:16] BIST_FAIL_RESULT_OFFSET: Holds the offset pointer of the BIST failure results. The data at the pointer location will be as below:</p> <ul style="list-style-type: none"> OFFSET [0] to OFFSET [8]: [287:0]BIST_EXP_DATA - Holds the expected read data for BIST data check failure. OFFSET [9] to OFFSET [10]: [37:0]BIST_FAIL_ADDR - Holds the actual failing address for BIST data check failure. OFFSET [11] to OFFSET [20]: [287:0] BIST_FAIL_DATA - Holds the actual failing data for BIST data check failure. <p>[15:8] BIST_FAIL_RESULT_SIZE: Holds the size of the BIST failure results. The value of this will be 640 bits.</p>
BIST_STATUS_INTF1	0x2E4	

continued...

Register Name	Byte Offset (Hexadecimal)	Description
		<p>[1:1] BIST_RESULT: Holds the result of the BIST operation. For this BIST mode, the test will end at the first failure, or completely check the specified data range if no failures were found. This value is valid when BIST_STATUS indicates that the BIST operation has completed.</p> <ul style="list-style-type: none"> 1'b0 = Data check failed. 1'b1 = Data check passed. <p>[0:0] BIST_STATUS: Indicates the status of the BIST operation.</p> <ul style="list-style-type: none"> 1'b0 – BIST operation still in progress if previously initiated. 1'b1 – BIST operation has been completed.
ECC_ERR_STATUS	0x300	<p>[Output] ECC Error Status</p> <p>This register contains the status of the ECC error buffer.</p> <p>[Fields]</p> <p>[31:16] ECC_ERR_OVERFLOW: This field indicates that buffer overflow has occurred due to a new ECC interrupt occurring after the ECC error buffer is at full capacity of 16 entries. Each new interrupt type that cannot be enqueued into the buffer are logically-OR'd to the existing contents of the overflow status field. The ECC interrupt types indicated by the different positional bits in the overflow field are as below:</p> <ul style="list-style-type: none"> [0] – Single-bit error. [1] – Multiple single-bit errors. [2] – Double-bit error. [3] – Multiple double-bit errors. [8] – Single-bit error during ECC scrubbing. [9] – Write link ECC single-bit error (LPDDR5 only). [10] – Write link ECC double-bit error (LPDDR5 only). [11] – Read link ECC single-bit error (LPDDR5 only). [12] – Read link ECC double-bit error (LPDDR5 only). [13] – RMW read link ECC double-bit error (LPDDR5 only). <p>[15:0] ECC_ERR_COUNTER: This field indicates the number of entries in ECC error buffer. The buffer has a maximum capacity of 16 entries after which overflow occurs. The mailbox command ECC_CLEAR_ERR_BUFFER clears the ECC error buffer and resets the ECC_ERR_COUNTER to 0.</p>
ECC_ERR_DATA_START	0x310	<p>[Output] ECC Error Data Start</p> <p>This register is the start of the ECC error data buffer. The buffer has a maximum capacity of 16 ECC errors. Each ECC error data entry has a size of 64 bits arranged in two 32-bit registers (Register 1 and Register 2). The mailbox command ECC_CLEAR_ERR_BUFFER clears the ECC error buffer and resets the ECC_ERR_COUNTER to 0.</p> <p>[Fields]</p> <p>R1[24:22] IP_TYPE: This field specifies the IP type of the interface that produced the ECC interrupt. IP_TYPE of 0 indicates that the buffer entry is not populated with valid data.</p> <p>R1[21:17] INSTANCE_ID: This field specifies the instance ID of the interface that produced the ECC interrupt.</p> <p>R1[16:10] ECC_ERR_SOURCE_ID: This field specifies the source ID associated with the 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.</p> <p>R1[9:6] ECC_ERR_TYPE: This field specifies the ECC interrupt type of the ECC event.</p> <ul style="list-style-type: none"> 1'b0000 – Single-bit error. 1'b0001 – Multiple single-bit errors. 1'b0010 – Double-bit error.

continued...

Register Name	Byte Offset (Hexadecimal)	Description
		<ul style="list-style-type: none"> 1'b0011 – Multiple double-bit errors. 1'b1000 – Single-bit error during ECC scrubbing. 1'b1001 – Write link ECC single-bit error (LPDDR5 only). 1'b1010 – Write link ECC double-bit error (LPDDR5 only). 1'b1011 – Read link ECC single-bit error (LPDDR5 only). 1'b1100 – Read link ECC double-bit error (LPDDR5 only). 1'b1101 – Read link ECC double-bit error, caused by read-modify-write operation (LPDDR5 only). <p>R1[5:0] ECC_ERR_ADDR_UPPER: This field specifies the upper 6 bits of the address of the read data that caused the ECC event. The Controller will pad this parameter with zeros for any address bits not used by the controller.</p> <p>R2[31:0] ECC_ERR_ADDR_LOWER: This field specifies the lower 32 bits of the address of the read data that caused the ECC event. The Controller will pad this parameter with zeros for any address bits not used by the controller.</p> <p><i>Note:</i> Refer to ECC Error Handling for details on ECC Error Buffer Byte Offset.</p>
STATUS	0x400	<p>[Output] Status Register</p> <p>“At a Glance” status register. This field is automatically updated by the Calibration IOSSM and no explicit operation is required to trigger an update.</p> <ul style="list-style-type: none"> 1'b001 – Calibration was successful. 1'b010 – Calibration failed. 1'b100 – Calibration ongoing. 1'b000 – Interface unused.
STATUS_CAL_INTF0	0x404	[Output] Memory Calibration Status
STATUS_CAL_INTF1	0x408	<p>This register provides the calibration status for memory interface 0/1.</p> <p>[Fields]</p> <p>[2:0] STATUS: Calibration Status of memory interface</p> <ul style="list-style-type: none"> 1'b001 – Calibration was successful. 1'b010 – Calibration failed. 1'b100 – Calibration ongoing. 1'b000 – Interface unused.

Mailbox Supported Commands

The following registers require handshaking procedures with the mailbox for successful retrieval of information. Refer to the instructions described in *Sending a Mailbox Command* for more information on handshaking procedures.

Table 236. Supported Commands

CMD_TYPE Enum	CMD_OPCODE Enum	Value
CMD_TRIG_CONTROLLER_OP (value=0x4)	ECC_ENABLE_SET	0x0101
	ECC_INTERRUPT_MASK	0x0105
	ECC_WRITEBACK_ENABLE	0x0106
	ECC_INJECT_ERROR	0x0109
	ECC_CLEAR_ERR_BUFFER	0x0110
	ECC_SCRUB_MODE_0_START	0x0202
continued...		

CMD_TYPE Enum	CMD_OPCODE Enum	Value
	ECC_SCRUB_MODE_1_START	0x0203
	BIST_STANDARD_MODE_START	0x0301
	BIST_MEM_INIT_START	0x0303
	BIST_SET_DATA_PATTERN_UPPER	0x0305
	BIST_SET_DATA_PATTERN_LOWER	0x0306
	CHANGE_FSP_LP5	0x0c01
	LP_MODE_ENTER	0x0d01
	LP_MODE_EXIT	0x0d02
	LP_MODE_AUTO	0x0d04
CMD_TRIG_MEM_CAL_OP (value=0x5)	TRIG_MEM_CAL	0x000a

13.1.2. Mailbox Command Definitions

Table 238. CMD_REQ Definition

Bit	Name	Description	Access	Reset
[31:29]	CMD_TARGET_IP_TYPE	Indicates the type of IP, as follows: <ul style="list-style-type: none"> 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 239. CMD_TYPE Definition

CMD_TYPE	Value	Description
CMD_NOP	0x00	No operation command.
CMD_TRIG_CONTROLLER_OP	0x04	Triggering memory controller-related operations.
CMD_TRIG_MEM_CAL_OP	0x05	Triggering calibration events.

Table 240. CMD_TARGET_IP_TYPE Definition

Multi-channel/ Lockstep Configurations	CMD_TARGET_IP_TYPE			
	1 – Primary MC, Primary IO96B	2 – Secondary MC, Primary IO96B	3 – Primary MC, Secondary IO96B	4 – Secondary MC, Secondary IO96B
LPDDR4/5 2CHx16	CH1	CH2		
LPDDR4/5 4CHx16	CH1	CH2	CH3	CH4
DDR5 2CHx16	CH1	CH2		
DDR5 2CHx32	CH1		CH2	
DDR5 x40 lockstep	CH1	*		
DDR4 x40 lockstep	CH1	*		
DDR4 x64, x72 lockstep	CH1	*	*	*

Note: * These controllers are used but have no (or limited) mailbox features due to limited lockstep capabilities.

Table 241. Command Definitions

CMD_REQ	Description
<p><i>CMD_REQ</i> [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> <i>CMD_REQ</i> [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> <i>CMD_REQ</i> [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP <i>CMD_REQ</i> [15:0]: CMD_OPCODE = ECC_ENABLE_SET</p>	<p>Command to enable different ECC modes for the memory interface specified using the instance ID. ECC mode can be changed only if the design was generated with ECC enabled.</p> <p>[Inputs] <i>CMD_PARAM_0</i> [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.</p> <p>[Outputs] [Errors] <i>CMD_RESPONSE_STATUS</i> - STATUS_CMD_RESPONSE_ERROR: 000 - No errors 001 - ECC was not enabled during design generation. [Outputs] N/A</p>
<p><i>CMD_REQ</i> [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> <i>CMD_REQ</i> [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> <i>CMD_REQ</i> [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP <i>CMD_REQ</i> [15:0]: CMD_OPCODE = ECC_INTERRUPT_MASK</p>	<p>Command to set mask for ECC interrupts for the memory interface specified using the instance ID, in order to disable specific ECC interrupts.</p> <p>[Inputs] <i>CMD_PARAM_0</i> [13:0]: 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.</p>

continued...

CMD_REQ	Description
	<p>Bit [13] = A RMW Read Link ECC double-bit error has been detected</p> <p>Bit [12] = A Read Link ECC double-bit error has been detected.</p> <p>Bit [11] = A Read Link ECC single-bit error has been detected.</p> <p>Bit [10] = A Write Link ECC double-bit error has been detected by the periodic MRR to MR43.</p> <p>Bit [9] = A Write Link ECC single-bit error has been detected by the periodic MRR to MR43.</p> <p>Bit [8] = An ECC correctable error has been detected in a scrubbing read operation</p> <p>Bit [7] = The triggered scrub operation has completed.</p> <p>Bit [6] = One or more ECC writeback commands could not be executed.</p> <p>Bit [3] = Another un-correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged.</p> <p>Bit [2] = An un-correctable ECC event has been detected on a read operation.</p> <p>Bit [1] = Another correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged.</p> <p>Bit [0] = A correctable ECC event has been detected on a read operation</p> <p>[Outputs] N/A</p>
<p>CMD_REQ column is:</p> <p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></p> <p>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></p> <p>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</p> <p>CMD_REQ [15:0]: CMD_OPCODE = ECC_WRITEBACK_ENABLE</p>	<p>Command to set automatic writing of corrected errors on read operation for the memory interface specified using the instance ID</p> <p>CMD_PARAM_0 [0:0]: ECC_WRITEBACK_EN</p> <p>Enables automatic writing of corrected data on single bit correctable errors on read operations. This parameter is only relevant if ECC is enabled with detection and correction (ECC_ENABLE = 'b11). Note: No writebacks will be issued during BIST. 'b0 = Disable. 'b1 = Enable</p> <p>[Outputs] N/A</p> <p>[Command-Specific Errors]</p> <p>'b000 – No errors</p> <p>'b001 – ECC detection and correction not enabled</p>
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></p> <p>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></p> <p>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</p> <p>CMD_REQ [15:0]: CMD_OPCODE = ECC_SCRUB_MODE_0_START</p>	<p>Command to start ECC scrub in mode 0 where scrub is performed at regular intervals for the memory interface specified using the instance ID. Automatic writing of corrected data should be enabled using the ECC_WRITEBACK_EN command before initiating scrub operations.</p> <p>[Inputs]</p> <p>CMD_PARAM_0 [15:0]: ECC_SCRUB_INTERVAL</p> <p>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.</p> <p>CMD_PARAM_1 [11:0]: ECC_SCRUB_LEN</p> <p>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. The burst lengths for different protocols are:</p> <p>DDR4 - 8</p>

continued...

CMD_REQ	Description
	<p>DDR5/LPDDR4/LPDDR5 - 16</p> <p>CMD_PARAM_2 [0:0]: ECC_SCRUB_FULL_MEM Defines whether to perform ECC scrub on full memory or on the specified address range.</p> <p>'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</p> <p>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.</p> <p>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.</p> <p>[Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: ECC_SCRUB_INITIATED 'b1 – ECC scrub initiated successfully 'b0 – ECC scrub initiation failed</p> <p>[Command-Specific Errors] 'b000 – No errors 'b001 – ECC not enabled</p>
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP CMD_REQ [15:0]: CMD_OPCODE = ECC_SCRUB_MODE_1_START</p>	<p>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. Automatic writing of corrected data should be enabled using the ECC_WRITEBACK_EN command before initiating scrub operations.</p> <p>[Inputs] CMD_PARAM_0 [15:0]: ECC_SCRUB_IDLE_CNT Defines the number of controller clock cycles that the scrubbing engine waits in the Controller's idle state before starting scrubbing operations. The Controller is considered idle when the command queue is empty. When this condition is detected, an internal counter loads with the value programmed in this parameter and count down on each controller clock. When the counter expires, either the scrubbing operation begins or the next address is tested. The controller clock is based on the Controller's operating frequency. Clearing this parameter to 0x0000 disables idle operation.</p> <p>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. The burst lengths for different protocols are: DDR4 - 8 DDR5/LPDDR4/LPDDR5 - 16</p> <p>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</p>

continued...

CMD_REQ	Description
	<p>'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.</p> <p>[Outputs] CMD_RESPONSE_DATA_SHORT [0:0]: ECC_SCRUB_INITIATED 'b1 – ECC scrub initiated successfully 'b0 – ECC scrub initiation failed</p> <p>[Command-Specific Errors] 'b000 – No errors 'b001 – ECC not enabled</p>
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID> CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP CMD_REQ [15:0]: CMD_OPCODE = ECC_INJECT_ERROR</p>	<p>Command to force a specific check code to be written into memory interface specified using the instance ID for diagnostic purposes or for flagging a particular memory address as erroneous for future accesses. The procedure is as follows:</p> <ol style="list-style-type: none"> 1. Set the ECC_ENABLE parameter to enable detection ('b1x). 2. Ensure that no writes to the controller are pending. 3. Send ECC_INJECT_ERROR command through mailbox setting the ECC_XOR_CHECK_BITS input parameter. Use the syndromes to program the ECC_XOR_CHECK_BITS parameter. Each byte of the ECC_XOR_CHECK_BITS parameter controls the ECC event forcing for a separate user-word space. For example, setting a value of 0xF4 as ECC_XOR_CHECK_BITS will result in the check bits to be updated and written to the memory such that it flags a single-bit correctable error on bit [0] of the user-word on subsequent access of the same address. 4. Execute a write command for an aligned user word. The controller will XOR the ECC_XOR_CHECK_BITS parameter with the generated checksum bits from the word written to the memory. The next read command to the same address will force the ECC error. 5. Depending on the programming of the ECC_XOR_CHECK_BITS parameter, a single bit, double bit or multi-bit ECC error will occur. For single bit and double bit errors, the appropriate bit in the ECC_INTERRUPT_STATUS parameter will be set to 'b1 and the ECC error signature parameters will be filled with the relevant information. <p>[Inputs] CMD_PARAM_0 [31:0]: ECC_XOR_CHECK_BITS The check bits generated by the next write operation will be XOR'ed with this parameter. The result will be written into memory as the new check value. ECC_XOR_CHECK_BITS [31:24] maps to user word [255:192]</p>

continued...

CMD_REQ	Description
	<p>ECC_XOR_CHECK_BITS [23:16] maps to user word [191:128] ECC_XOR_CHECK_BITS [15:8] maps to user word [127:64] ECC_XOR_CHECK_BITS [7:0] maps to user word [63:0] [Outputs] N/A [Command-Specific Errors] 'b000 – No errors 'b001 – ECC detection not enabled</p>
<p>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_TRIG_CONTROLLER_OP CMD_REQ [15:0]: CMD_OPCODE = ECC_CLEAR_ERR_BUFFER</p>	<p>Command to clear the ECC error buffer and reset the ECC error counter and overflow registers. [Inputs] N/A [Outputs] N/A</p>
<p>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</p>	<p>Command to initiate Original MOV11 3N BIST algorithm for data checking for the memory interface specified using the instance ID. Use BIST_STATUS_INTF0/ BIST_STATUS_INTF1 registers to get the status of the operation. BIST operations are supported only for designs with DBI disabled. [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 2^{addr_space} that the BIST logic will check. As an example, if the addr_space parameter was programmed to 0x1c, then the BIST logic would check 2²⁸ bytes = 256 MBytes. Only used if BIST_FULL_MEM is 'b0. A BIST test must cover a minimum of 2 bursts. Therefore, the user must program this parameter to a value such that the start address and end address of the BIST test will encompass a minimum of 2 bursts. The burst length for different protocols are: DDR4 - 8 DDR5/LPDDR4/LPDDR5 - 16 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.</p>
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE> CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></p>	<p>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. Use MEM_INIT_STATUS_INTF0/ MEM_INIT_STATUS_INTF1</p>

continued...

CMD_REQ	Description
<p>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP CMD_REQ [15:0]: CMD_OPCODE = BIST_MEM_INIT_START</p>	<p>registers to get the status of the operation. If in-line ECC is enabled and you want to complete memory initialization on the entire memory, then before running the test, you should disable in-line ECC.</p> <p>[Inputs]</p> <p>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 $2^{\text{addr_space}}$ that the BIST logic will check. As an example, if the addr_space parameter was programmed to 0x1c, then the BIST logic would check 2^{28} bytes = 256 MBytes. Only used if BIST_FULL_MEM is 'b0.</p> <p>Note: A BIST test must cover a minimum of 2 bursts. Therefore, the user must program this parameter to a value such that the start address and end address of the BIST test will encompass a minimum of 2 bursts. The burst length for different protocols are: DDR4 - 8 DDR5/LPDDR4/LPDDR5 - 16</p> <p>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</p> <p>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.</p> <p>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.</p> <p>[Outputs]</p> <p>CMD_RESPONSE_DATA_SHORT [0:0]: BIST_INITIATED 'b1 - BIST memory initialization initiated successfully 'b0 - BIST memory initialization initiation failed</p> <p>[Command-Specific Errors]</p> <p>'b00 - No errors 'b01 - A previous command's saved state not restored. 'b10 - In-line ECC must be disabled before full memory initialization. Either disable in-line ECC or specify a valid address range to be initialized.</p>
<p>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</p>	<p>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.</p>

continued...

CMD_REQ	Description
	<p>The active portion of the core word depends on the DQ width per channel and the burst length. For example:</p> <ul style="list-style-type: none"> - DDR5/LPDDR4/LPDDR5 (BL16) will use BIST_DATA_PATTERN [255:0] for x32 and BIST_DATA_PATTERN [127:0] for x16. - DDR4 (BL8) will use BIST_DATA_PATTERN [127:0] for x32 and BIST_DATA_PATTERN [63:0] for x16. <p>If the axi4_rdata bus width is greater than the width of the active BIST_DATA_PATTERN, you will get repeated BIST_DATA_PATTERN when reading from the initialized memory locations. For example:</p> <ul style="list-style-type: none"> - DDR4 x32: axi4_rdata[255:0] = { 2{BIST_DATA_PATTERN[127:0]} } - DDR4 x16: axi4_rdata[255:0] = { 4{BIST_DATA_PATTERN[63:0]} } <p>[Outputs] N/A</p> <p>[Command-Specific Errors]</p> <p>'b00 – No errors</p> <p>'b01 – Cannot set upper data pattern for slim interface.</p>
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></p> <p>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></p> <p>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</p> <p>CMD_REQ [15:0]: CMD_OPCODE = BIST_SET_DATA_PATTERN_LOWER</p>	<p>CMD_PARAM_4 [31:0]: BIST_DATA_PATTERN [159:128]</p> <p>CMD_PARAM_3 [31:0]: BIST_DATA_PATTERN [127:96]</p> <p>CMD_PARAM_2 [31:0]: BIST_DATA_PATTERN [95:64]</p> <p>CMD_PARAM_1 [31:0]: BIST_DATA_PATTERN [63:32]</p> <p>CMD_PARAM_0 [31:0]: BIST_DATA_PATTERN [31:0]</p> <p>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.</p> <p>The active portion of the core word depends on the DQ width per channel and the burst length. For example:</p> <ul style="list-style-type: none"> - DDR5/LPDDR4/LPDDR5 (BL16) will use BIST_DATA_PATTERN [255:0] for x32 and BIST_DATA_PATTERN [127:0] for x16. - DDR4 (BL8) will use BIST_DATA_PATTERN [127:0] for x32 and BIST_DATA_PATTERN [63:0] for x16. <p>If the axi4_rdata bus width is greater than the width of the active BIST_DATA_PATTERN, you will get repeated BIST_DATA_PATTERN when reading from the initialized memory locations. For example:</p> <ul style="list-style-type: none"> - DDR4 x32: axi4_rdata[255:0] = { 2{BIST_DATA_PATTERN[127:0]} } - DDR4 x16: axi4_rdata[255:0] = { 4{BIST_DATA_PATTERN[63:0]} }
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></p> <p>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></p> <p>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</p> <p>CMD_REQ [15:0]: CMD_OPCODE = CHANGE_FSP_LP5</p>	<p>Command to change the FSP when using the LPDDR5 protocol.</p> <p>[Inputs]</p> <p>CMD_PARAM_0 [1:0]:</p> <p>TARGET_FSP</p> <p>'b00 – FSP 0</p> <p>'b01 – FSP 1</p> <p>'b10 – FSP 2</p> <p>[Outputs]</p> <p>CMD_RESPONSE_DATA_SHORT [0:0]:</p> <p>FSP_CHANGE_STATUS</p> <p>'b1 – The FSP was changed successfully.</p> <p>'b0 – The FSP was not changed successfully.</p> <p>[Error Codes]</p> <p>'b000 – No errors.</p>

continued...

CMD_REQ	Description
	<p>'b001 – Current Protocol is not LPDDR5. 'b010 – The current FSP is equal to the target FSP – No change is needed. 'b011 – The FSP could not be changed.</p>
<p><i>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></i> <i>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></i> <i>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</i> <i>CMD_REQ [15:0]: CMD_OPCODE = LP_MODE_ENTER</i></p>	<p>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.</p> <p>[Inputs] CMD_PARAM_0[3:0]: LP_STATE Specified the low power state the interface should enter: '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) '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)</p> <p>[Outputs] N/A</p> <p>[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. 'b011 – Disable auto LP mode using the LP_MODE_AUTO command and confirm the idle state using the LP_MODE_STATUS command before using the LP_MODE_ENTER command.</p>
<p><i>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></i> <i>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></i> <i>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</i> <i>CMD_REQ [15:0]: CMD_OPCODE = LP_MODE_EXIT</i></p>	<p>Command to exit any low power state.</p> <p>[Inputs] N/A [Outputs] N/A [Error Codes] 'b000 – No errors</p>
<p><i>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></i> <i>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></i> <i>CMD_REQ [23:16]: CMD_TYPE = CMD_TRIG_CONTROLLER_OP</i> <i>CMD_REQ [15:0]: CMD_OPCODE = LP_MODE_AUTO</i></p>	<p>Command to cause the Interface to enter certain automatic low power states. Supported low power modes for this command are listed below as valid options. Do not use LP_MODE_EXIT opcode to exit from the automatic low power states. Note that other interface operations, including recalibration and mode register reads and writes, sending read/write request to the controller can cause automatic exits from some low-power states.</p> <p>[Inputs] CMD_PARAM_0 [3:0]: Specified the low power state the interface should enter. 'b0000 – Disable automatic low power mode. '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).</p>

continued...

CMD_REQ	Description
	<p>'b0101 – Self-Refresh Short (DDR4/DDR5 only).</p> <p>'b0110 – Self-Refresh Short with Memory Clock Gating (DDR4/DDR5 only).</p> <p>'b1011 – Self-Refresh Power Down Short (LPDDR4/LPDDR5 only).</p> <p>'b1100 – Self-Refresh Power Down Short with Memory Clock Gating (LPDDR4/LPDDR5 only).</p> <p>CMD_PARAM_1 [11:0]: Defines the number of idle controller clocks that can elapse before the controller will automatically issue an entry into the low power state specified</p> <p>[Outputs]</p> <p>[Error Codes]</p> <p>'b000 – No errors</p> <p>'b001 – The selected low power state is invalid for the current protocol.</p> <p>'b010 – The selected low power state is invalid/does not exist.</p> <p>'b011 – The idle clock count should not be zero.</p>
<p>CMD_REQ [31:29]: CMD_TARGET_IP_TYPE = <TARGET_IP_TYPE></p> <p>CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID = <TARGET_IP_INSTANCE_ID></p> <p>CMD_REQ [23:16]: CMD_TYPE =</p> <p>CMD_TRIG_MEM_CAL_OP</p> <p>CMD_REQ [15:0]: CMD_OPCODE = TRIG_MEM_CAL</p>	<p>Command to trigger a memory calibration for the memory interface specified using the instance ID. You should only recalibrate the memory interface for debug purposes. (HPS EMIF on Agilex 5 devices does not support recalibration.)</p> <p>For multi-channel interfaces, recalibration can be initiated by sending the TRIG_MEM_CAL command as below:</p> <ul style="list-style-type: none"> 1 channel – 1 request to the interface 2 channel LPDDR4/LPDDR5 – 1 request to one of the interfaces 2 channel DDR5 – 2 separate requests to the interfaces in each IO96B 4 channel LPDDR4/LPDDR5 – 2 requests to one of the interfaces in each IO96B <p>[Inputs]</p> <p>[Outputs]</p> <p>CMD_RESPONSE_DATA_SHORT [0]: MEM_CAL_INITIATED</p> <p>'b1 – Memory calibration triggered successfully</p> <p>'b0 – Memory calibration triggering failed</p> <p>[Errors]</p> <p>CMD_RESPONSE_STATUS - STATUS_CMD_RESPONSE_ERROR:</p> <p>000 - No errors</p> <p>001 - Triggering memory calibration is not supported for this memory interface.</p>

13.1.3. ECC Syndrome Codes

Table 242. ECC Syndrome Codes

Syndrome	Incorrect Bit	Syndrome	Incorrect Bit	Syndrome	Incorrect Bit
0x00	No Error	0x4f	Data [46]	0xa4	Data [22]
0x01	Check [0]	0x52	Data [45]	0xa7	Data [21]
0x02	Check [1]	0x54	Data [44]	0xa8	Data [20]
continued...					

Syndrome	Incorrect Bit	Syndrome	Incorrect Bit	Syndrome	Incorrect Bit
0x04	Check [2]	0x57	Data [43]	0xab	Data [19]
0x08	Check [3]	0x58	Data [42]	0xad	Data [18]
0x0b	Data [63]	0x5b	Data [41]	0xb0	Data [17]
0x0e	Data [62]	0x5e	Data [40]	0xb5	Data [16]
0x10	Check [4]	0x62	Data [39]	0xcb	Data [15]
0x13	Data [61]	0x64	Data [38]	0xce	Data [14]
0x15	Data [60]	0x67	Data [37]	0xd3	Data [13]
0x16	Data [59]	0x68	Data [36]	0xd5	Data [12]
0x19	Data [58]	0x6b	Data [35]	0xd6	Data [11]
0x1a	Data [57]	0x6d	Data [34]	0xd9	Data [10]
0x1c	Data [56]	0x70	Data [33]	0xda	Data [9]
0x20	Check [5]	0x75	Data [32]	0xdc	Data [8]
0x23	Data [55]	0x80	Check [7]	0xe3	Data [7]
0x25	Data [54]	0x8a	Data [31]	0xe5	Data [6]
0x26	Data [53]	0x8f	Data [30]	0xe6	Data [5]
0x29	Data [52]	0x92	Data [29]	0xe9	Data [4]
0x2a	Data [51]	0x94	Data [28]	0xea	Data [3]
0x2c	Data [50]	0x97	Data [27]	0xec	Data [2]
0x31	Data [49]	0x98	Data [26]	0xf1	Data [1]
0x34	Data [48]	0x9b	Data [25]	0xf4	Data [0]
0x40	Check [6]	0x9d	Data [24]		
0x4a	Data [47]	0xa2	Data [23]		
Any other Syndrome			Multi-bit error		

13.1.4. ECC Error Handling

The user RAM includes an integrated ECC error buffer, that improves ECC error management capabilities with the mailbox. The ECC error buffer is administered by the IOSSM, which routinely inspects the memory controller's interrupt status to detect ECC events.

Upon identifying an ECC event, the IOSSM captures and stores relevant details from the event, including the following:

- The type of ECC error that triggered the event.
- The memory address involved in the error.
- The Source ID linked to the error-causing transaction.

Table 243. ECC Error Buffer Byte Offset

Register Name	Byte Offset (Hexadecimal)	Description
ECC Error Status	0x0300	ECC_ERROR_COUNTER, ECC_ERR_OVERFLOW
ECC Error Buffer Entry 0 – (ECC_ERR_DATA_START)	0x0310	ENTRY 0 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0314	ENTRY 0 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 1	0x0318	ENTRY 1 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x031C	ENTRY 1 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 2	0x0320	ENTRY 2 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0324	ENTRY 2 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 3	0x0328	ENTRY 3 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x032C	ENTRY 3 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 4	0x0330	ENTRY 4 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0334	ENTRY 4 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 5	0x0338	ENTRY 5 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x033C	ENTRY 5 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 6	0x0340	ENTRY 6 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0344	ENTRY 6 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 7	0x0348	ENTRY 7 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x034C	ENTRY 7 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 8	0x0350	ENTRY 8 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0354	ENTRY 8 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 9	0x0358	ENTRY 9 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x035C	ENTRY 9 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 10	0x0360	ENTRY 10 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0364	ENTRY 10 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 11	0x0368	ENTRY 11 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x036C	ENTRY 11 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 12	0x0370	ENTRY 12 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0374	ENTRY 12 R2: ECC_ERR_ADDR_LOWER
<i>continued...</i>		

Register Name	Byte Offset (Hexadecimal)	Description
ECC Error Buffer Entry 13	0x0378	ENTRY 13 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x037C	ENTRY 13 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 14	0x0380	ENTRY 14 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x0384	ENTRY 14 R2: ECC_ERR_ADDR_LOWER
ECC Error Buffer Entry 15	0x0388	ENTRY 15 R1: IP_TYPE, INSTANCE_ID, ECC_ERR_SOURCE_ID, ECC_ERR_TYPE, ECC_ERR_ADDR_UPPER
	0x038C	ENTRY 15 R2: ECC_ERR_ADDR_LOWER

When the IOSSM firmware identifies an ECC event and the buffer has not yet reached its maximum capacity (as indicated by the ECC_ERR_COUNTER not exceeding the maximum capacity of 16 entries), it then populates the next buffer entry with pertinent information in the format outlined in the table below.

Table 244. ECC Error Buffer Structure

32-Bit Register	Bit Range	Register Field	Description
Register 1 (R1)	5:0	ECC_ERR_ADDR_UPPER	Upper 6 bits of the ECC Error Address
	9:6	ECC_ERR_TYPE	Type of ECC Error
	16:10	ECC_ERR_SOURCE_ID	Source Transaction AXI ID
	21:17	INSTANCE_ID	EMIF Instance ID
	22:24	IP_TYPE	EMIF IP Type
	31:25	—	Reserved
Register 2 (R2)	31:0	ECC_ERR_ADDR_LOWER	Lower 32 bits of the ECC Error Address

ECC Error Buffer Overflow Handling

The ECC error buffer is considered full when the ECC_ERR_COUNTER reaches the buffer's maximum capacity of 16 entries. If the buffer becomes full, the IOSSM does not write new data to the ECC error buffer for any subsequent interrupts. Instead, the ECC_ERR_OVERFLOW bits corresponding to each new interrupt type that cannot be buffered are logically OR'ed with the existing overflow status field's contents.

Table 245. ECC Error Types and Overflow

ECC Error Type	ECC_ERR_TYPE	ECC_ERR_OVERFLOW
Single-bit error	'b0000	'b0000_0000_0000_0001
Multiple single-bit errors	'b0001	'b0000_0000_0000_0010
Double-bit error	'b0010	'b0000_0000_0000_0100
Multiple double-bit errors	'b0011	'b0000_0000_0000_1000
Single-bit error during ECC scrubbing	'b1000	'b0000_0000_1000_0000
Write link ECC single-bit error (LPDDR5 only)	'b1001	'b0000_0001_0000_0000
Write link ECC double-bit error (LPDDR5 only)	'b1010	'b0000_0010_0000_0000
continued...		

ECC Error Type	ECC_ERR_TYPE	ECC_ERR_OVERFLOW
Read link ECC single-bit error (LPDDR5 only)	'b1011	'b0000_0100_0000_0000
Read link ECC double-bit error (LPDDR5 only)	'b1100	'b0000_1000_0000_0000
Read link ECC double-bit error, caused by read-modify-write operation (LPDDR5 only)	'b1101	'b0001_0000_0000_0000

Clearing ECC Error Buffer

To clear the ECC error buffer and reset the ECC_ERR_COUNTER and the ECC_ERR_OVERFLOW to zero, you should issue the ECC_CLEAR_ERR_BUFFER mailbox command. This command ensures that the buffer is empty and allows for new entries.

13.2. Accessing Read-Only Registers

To access read-only registers, you must follow these steps:

1. Read the desired register information from base address (0x0500_0000) + Byte Offset.
2. Decode the retrieved read data by referring to the description on each register.

The following examples illustrate read-only register access.

13.2.1. Example 1: Reading IP_TYPE and IP_INSTANCE_ID of All the Interfaces in the IO96B Through Read-Only Registers

This example provides instructions for reading the IP_TYPE and IP_INSTANCE_ID of all the interfaces in the IO96B using the mailbox.

The values shown below are for illustrative purposes and are obtained from an EMIF example design using DDR4 x32 + ECC implemented on the Agilex 5 FPGA E-Series 065B Premium Development Kit. This configuration uses the Primary MC of the Primary IO96B.

Base address=0x500_0000

Address for each read-only register = Base address + offset of each register

Register Name	Byte Offset (Hexadecimal)	Address (Hexadecimal)
MEM_INTF_INFO_0	0x200	0x5000200
MEM_INTF_INFO_1	0x280	0x5000280

1. Read from MEM_INTF_INFO_0 (address = 0x5000200)
 The expected read_data=0x2000_0000.
 IP_TYPE for interface 0 = MEM_INTF_INFO_0[31:29] = 0x1, indicating interface 0 is using Primary MC of Primary IO96B.
 INSTANCE_ID for interface 0 = MEM_INTF_INFO_0 [28:24] = 0x0.
2. Read from MEM_INTF_INFO_1 (address = 0x5000280)
 The expected read_data=0x0000_0000.

IP_TYPE for interface 1 = MEM_INTF_INFO_1 [31:29] = 0x0, indicating interface 1 does not exist.

INSTANCE_ID for interface 1 = MEM_INTF_INFO_1 [28:24] = 0x0; not a valid data as interface 1 does not exist.

13.2.2. Example 2: Reading the Memory Clock Frequency for an Interface

This example illustrates the reading of the memory clock frequency for an interface in the IO96B from read-only registers.

The values in this example are for illustrative purposes and are obtained from an EMIF example design with DDR4 x32 + ECC running at 800MHz on the Agilex 5 FPGA E-Series 065B Development Kit - Premium. This configuration uses the Primary MC of the Primary IO96B.

Base address=0x500_0000

Address for each read-only register = Base address + offset of each register

Register Name	Byte Offset (Hexadecimal)	Address (Hexadecimal)
MEMCLK_FREQ_FSP_CUR_INTF0	0x220	0x5000220

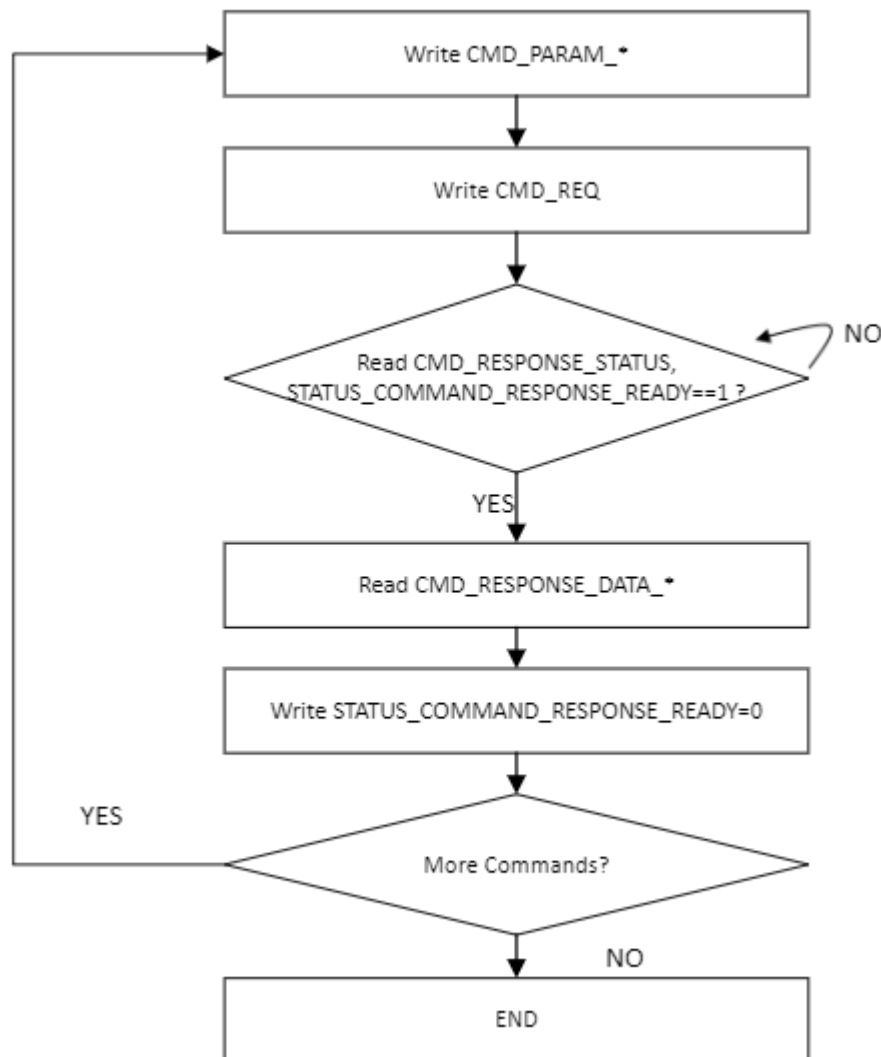
The expected read_data=0x000c_3500

13.3. Sending a Mailbox Command

Follow these steps to send a mailbox command.

1. Wait for the CMD_REQ to be cleared to ensure the previous command has been dispatched.
2. Write to the CMD_PARAM_* register. If the command does not require any CMD_PARAM – for example, TRIG_MEM_CAL – you can skip this step.
3. Write to the CMD_REQ with the desired command (CMD_TARGET_IP_TYPE, CMD_TARGET_INSTANCE_ID, CMD_TYPE and CMD_OPCODE)
4. Wait until STATUS_CMD_RESPONSE_READY (BIT 0 of CMD_RESPONSE_STATUS) equals 1 .
5. Read the CMD_RESPONSE_DATA_* and/or CMD_RESPONSE_DATA_SHORT (as required for the CMD_REQ).
6. Clear the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) . Perform a Read-Modify-Write operation:
 - a. Read from CMD_RESPONSE_STATUS register.
 - b. Write_Data = Data in (a) & 0xffff_fffe (Change only bit 0).
 - c. Write to Address = 0x500_045c Data= Write_Data from (b).

Figure 48. Sending a Mailbox Command



13.3.1. Example 3: Sending Recalibration Request and Reading Calibration Status Using the Mailbox

This example provides instructions for sending recalibration request and reading the calibration status using the mailbox.

The values in this example are for illustrative purposes and are obtained from an EMIF example design using DDR5 2CHx40 RDIMM, implemented on the Agilex 7 FPGA Development Kit –HBM2e Edition.

In a 2CHx40 DDR5 RDIMM design, CH1 and CH2 use the Primary MC in Primary IO96B and Secondary IO96B, respectively; hence the CMD_TARGET_IP_TYPE is 0x1 for CH1 , and 0x3 for CH2. The CMD_TARGET_IP_INSTANCE defaults to 0 in the example design generated by the Quartus Prime software.

The table below illustrates the construction of the data to be written to the CMD_REQ register for this request:

Channel	CMD_REQ [31:29]: CMD_TARGET_IP_TYPE	CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID	CMD_REQ [23:16]: CMD_TYPE	CMD_REQ [15:0]: CMD_OPCODE	CMD_REQ [31:0]:
CH1	0x1	0x0	0x05	0xa	0x2005_000a
CH2	0x3	0x0	0x05	0xa	0x6005_000a

Recalibrate CH1 using the JAMB/AXI-Lite interface for CH1

1. It is not required to write to the CMD_PARAM_* register as this OPCODE does not require any cmd_param.
2. Write to the CMD_REQ(address=0x500_043c) with write_data=0x2005_000a .
3. Read from CMD_RESPONSE_STATUS(address=0x500_045c) until you get the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) equals 1.
4. Clear the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) Perform a Read-Modify-Write operation:
 - a. Read from CMD_RESPONSE_STATUS register.
 - b. Write_Data = Data in (a) & 0xffff_fffe (Change only bit 0).
 - c. Write to address = 0x500_045c data= write_data from (b).

Recalibrate CH2 using the JAMB/AXI-Lite interface for CH2

Repeat the same steps as listed above. For Step 2, the write_data=0x6005_000a, because the CMD_TARGET_IP_TYPE for CH2 is 0x3 in this design.

Read the CH1 Calibration Status using the JAMB/AXI-Lite interface for CH1

1. Read from read-only register STATUS(address=0x500_0400) till you get either bit_1/bit_0 is asserted.
 - bit_0 = 1 means one or more EMIF instances have calibrated successfully.
 - bit_1 = 1 means one or more EMIF instances have failed to calibrate successfully.
 - bit_2 = 1 means one or more EMIF instances calibration still ongoing.
 - bit_1 and bit_2 and bit_3 = 0 means no EMIF interface used in the design.

Read the CH2 Calibration Status using the JAMB/AXI-Lite interface for CH2

Repeat the same steps as for CH1, above.

13.3.1.1. Example 4: Sending ECC_INJECT_ERROR Using the Mailbox and Reading ECC Error Status, ECC Error Buffer Registers

This example provides instructions for sending ECC_INJECT_ERROR using mailbox command.

The values in this example are for illustrative purposes and are obtained from an EMIF example design using LPDDR4 1ch x32 component, implement on the Agilex 5 FPGA E-Series 065B Premium Development Kit.

For a 1ch x32 LPDDR4 component design, the interface uses the Primary MC of Primary IO96B, hence the CMD_TARGET_IP_TYPE is 0x1. The CMD_TARGET_IP_INSTANCE defaults to 0 in the design example generated by the Quartus Prime software. The design example is generated with In-Line ECC enabled.

The following table illustrates the construction of the data to be written to the CMD_REQ register for this request:

Table 246. Construction of Data to be Written to the CMD_REQ Register

CMD_REQ [31:29]: CMD_TARGET_IP_TYPE	CMD_REQ [28:24]: CMD_TARGET_IP_INSTANCE_ID	CMD_REQ [23:16]: CMD_TYPE	CMD_REQ [15:0]: CMD_OPCODE	CMD_REQ [31:0]:
0x1	0x0	0x04	0x109	0x20040109

Flagging error on specific memory address using JAMB/AXI-Lite

Prerequisites:

- Ensure the design generated with ECC enabled. If you modify the ECC enable status via a mailbox command, verify ECC is enabled by reading the ECC_ENABLE_INTF0/1 read-only status register.
- Ensure the design is generated with a non-infinite Traffic Generator Program so that write traffic to a known address can be compiled and executed in user mode. Follow these steps for compiling traffic program to perform single write and read on a specific memory address:
 1. Open the *traffic_patterns.py* file from the <project directory>/qii/hydra_sw folder and navigate to a function called *tut1_block_rw*.
 2. Specify the address at "addr" parameter as shown in example below and save the file.

```
def tut1_block_rw(self):
    return [
        # Write to a specific address with specific data
        *self.gencmd("wr", addr=0x2000, data=0x0123456789ABCDEF),
        # Wait for all write responses to return
        wait_writes(),
        # Read from the same address and check if we get back the same data
        *self.gencmd("rd", addr=0x2000, data=0x0123456789ABCDEF),
        # Wait for all read responses to return
        wait_reads(),
        # Idle for 100 cycles
        sleep(100)
    ]
```

(Comment or remove all other commands within this function, except for the ones performing single write to a specific address and reading from the same address.)

3. Execute the command to compile the traffic program:

```
quartus_py main.py --ipdir=<project directory>/qii/ --prog=tut1_block_rw
```

4. The <project directory>/qii/hydra_sw/bin folder is updated from the command execution above. Copy the generated /bin folder to <project directory>/qii. The traffic program is now ready to perform a single write to a known address and read from the same address.

Execution:

1. Write to CMD_PARAM_0(address=0x 0x5000438) register the syndrome to program the ECC_XOR_CHECK_BITS parameter(write_data=0xf4).
 Note: 0xf4 will update the check bits and write to memory as single bit correctable error on bit [0]. Refer to Table 283 for ECC syndrome codes.
2. Write the CMD_REQ(address=0x500043C) with write_data=0x20040109.
3. Read from CMD_RESPONSE_STATUS(address=0x500_045c) until you get the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) equals 1.

4. Clear the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) Perform a Read-Modify-Write operation:
 - a. Read from CMD_RESPONSE_STATUS register.
 - b. Write_Data = Data in (a) & 0xffff_ffe (Change only bit 0).
 - c. Write to address = 0x500_045c data= write_data from (b).
5. Run traffic generator.

```
system-console --script=<path to testengine_library.tcl> --sof=<path to sof_file> --update=1 --n-loops=1
```

You should observe that the traffic has failed.

6. Read ECC_ERR_COUNTER from ECC_ERR_STATUS(address= 0x5000300) read-only register (bit 0 to bit 15). The expected read_data=0x0000_0001; indicates that 1 error encountered.
7. Read ECC Error Buffer Entry 0(R1 address= 0x5000310 , R2 address= 0x5000314).

The expected read_data from R1= 0x00400000

```
IP_TYPE [22:24]          : 1 ==> Primary MC of Primary I096B
INSTANCE_ID [21:17]      : 0
ECC_ERR_SOURCE_ID [16:10] : 0
ECC_ERR_TYPE [9:6]       : 0 ==> Single-bit error
ECC_ERR_ADDR_UPPER [5:0] : 0 ==> 0x0
```

The expected read_data from R2= 0x00002000.

```
ECC_ERR_ADDR_LOWER[31:0]: 8192 ==> 0x2000
```

8. Repeat the above steps and observe the ECC_ERR_COUNTER increase accordingly. Read the respective ECC Error Buffer Entry (0 to 15) based on the number of ECC events.

When the ECC error buffer reaches its full capacity of 16 entries, proceed to the next steps to check ECC_ERR_OVERFLOW status.

9. Read ECC_ERR_OVERFLOW from ECC_ERR_STATUS (address= 0x5000300) read-only register (bit 16 to bit 31).

The expected read_data=0x0001; indicates that single-bit error overflow occurred.

Clear the ECC error buffer, reset the ECC error counter and reset overflow register

It is not required to write to the CMD_PARAM_* register as this OPCODE does not require any cmd_param.

The following table illustrates the construction of the data to be written to the CMD_REQ register for this request using the same design as the previous example targeting an Agilex 5 FPGA E-Series 065B Premium Development Kit:

Table 247. Construction of Data to be Written to the CMD_REQ Register

CMD_REQ [31:29]: CMD_TARGET_IP_TY PE	CMD_REQ [28:24]: CMD_TARGET_IP_IN STANCE_ID	CMD_REQ [23:16]: CMD_TYPE	CMD_REQ [15:0]: CMD_OPCODE	CMD_REQ [31:0]:
0x1	0x0	0x04	0x110	0x20040110

1. Write the CMD_REQ(address=0x500043C) with write_data=0x20040110.
2. Read from CMD_RESPONSE_STATUS(address=0x500_045c) until you get the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) equals 1.
3. Clear the STATUS_COMMAND_RESPONSE_READY (Bit 0 of CMD_RESPONSE_STATUS register) Perform a Read-Modify-Write operation:
 - a. Read from CMD_RESPONSE_STATUS register.
 - b. Write_Data = Data in (a) & 0xffff_fffe (Change only bit 0).
 - c. Write to address = 0x500_045c data= write_data from (b).
4. Reading ECC_ERR_STATUS (address= 0x5000300) and ECC Error Buffer Entry (0 to 15) should return 0, indicates that the previous data cleared/reset successfully.

13.4. IOSSM Mailbox Access Script

This section discusses mailbox scripts, which are available to perform all mailbox operations, including reading status registers through read-only registers.

13.4.1. Mailbox Script Overview

The mailbox script provides complete implementation of command execution as described in previous chapters. This script uses TCL code that performs read and write operations to the respective registers.

The table below lists the three TCL files that are included in the download:

Table 248. TCL Files Included in Mailbox Script Download

File Name	Description
mb_base_parameter.tcl	Parameterization of addresses for each register.
mb_iossm_command.tcl	Functions to perform mailbox access flow and reading read-only register.
mb_user_input.tcl	User input to perform desired operation.

Call the respective function with the required argument or arguments from mb_user_input.tcl to perform any mailbox access operation.

Do not modify mb_base_parameter.tcl or mb_iossm_command.tcl.

You can download the mailbox access script at this location: [Agilex 5 IOSSM Mailbox Access Script](#).

13.4.1.1. mb_iossm_command.tcl

This topic describes the functions available for accessing a mailbox or read-only registers.

The following table lists the mailbox access script functions. (For detailed information on the corresponding arguments for these functions, refer to the subsequent *User Input Arguments Description* table.)

Table 249. Mailbox Access Script Functions

Registers	Functions	Argumentss
MAILBOX_HEADER MEM_INTF_INFO_0 MEM_INTF_INFO_1 MEM_TECHNOLOGY_INTF0 MEM_TECHNOLOGY_INTF1 MEMCLK_FREQ_INTF0 MEMCLK_FREQ_INTF1 MEMCLK_FREQ_FSP0_INTF0 MEMCLK_FREQ_FSP0_INTF1 MEMCLK_FREQ_FSP1_INTF0 MEMCLK_FREQ_FSP1_INTF1 MEMCLK_FREQ_FSP2_INTF0 MEMCLK_FREQ_FSP2_INTF1 MEM_WIDTH_INFO_INTF0 MEM_WIDTH_INFO_INTF1 MEM_TOTAL_CAPACITY_INTF0 MEM_TOTAL_CAPACITY_INTF1	intf_info	<calibration interface>
STATUS STATUS_CAL_INTF0 STATUS_CAL_INTF1	cal_status	<calibration interface>
ECC_ENABLE_INTF0 ECC_ENABLE_INTF1 ECC_SCRUB_STATUS_INTF0 ECC_SCRUB_STATUS_INTF1	ecc_status	<calibration interface>
MEM_INIT_STATUS_INTF0 MEM_INIT_STATUS_INTF1 BIST_STATUS_INTF0 BIST_STATUS_INTF1	bist_status	<calibration interface>
LP_MODE_INTF0 LP_MODE_INTF1	lp_mode_status	<calibration interface>
ECC_ENABLE_SET	mb_ecc_enable_set	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <ECC_ENABLE>
ECC_INTERRUPT_MASK	mb_ecc_interrupt_mask	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <ECC_INTERRUPT_MASK>
ECC_WRITEBACK_ENABLE	mb_ecc_writeback_enable	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <ECC_WRITEBACK_EN>
ECC_INJECT_ERROR	mb_ecc_inject_error	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <ECC_XOR_CHECK_BITS>
ECC_CLEAR_ERR_BUFFER	mb_ecc_clear_buffer	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID>
ECC_SCRUB_MODE_0_START	mb_ecc_scrub_mode_0_start	<calibration interface>
continued...		

Registers	Functions	Argumentss
		<CMD_TARGET_IP_TYPE> <ECC_SCRUB_INTERVAL> <ECC_SCRUB_LEN> <ECC_SCRUB_FULL_MEM> <ECC_SCRUB_START_ADDR [31:0]> <ECC_SCRUB_START_ADDR [36:32]> <ECC_SCRUB_END_ADDR [31:0]> <ECC_SCRUB_END_ADDR [36:32]>
ECC_SCRUB_MODE_1_START	mb_ecc_scrub_mode_1_start	<calibration interface> <CMD_TARGET_IP_TYPE> <ECC_SCRUB_IDLE_CNT> <ECC_SCRUB_LEN> <ECC_SCRUB_FULL_MEM> <ECC_SCRUB_START_ADDR [31:0]> <ECC_SCRUB_START_ADDR [36:32]> <ECC_SCRUB_END_ADDR [31:0]> <ECC_SCRUB_END_ADDR [36:32]>
BIST_STANDARD_MODE_START	mb_run_bist_std_mode_start	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <BIST_ADDR_SPACE [5:0]>
BIST_MEM_INIT_START	mb_bist_mem_init_start	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <BIST_DATA_PATTERN> <BIST_START_ADDR [37:32]> <BIST_START_ADDR [31:0]> <BIST_FULL_MEM>
BIST_SET_DATA_PATTERN_UPPER	mb_bist_set_data_pattern_upper	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <BIST_DATA_PATTERN [287:256]> <BIST_DATA_PATTERN [255:224]> <BIST_DATA_PATTERN [223:192]> <BIST_DATA_PATTERN [191:160]>
BIST_SET_DATA_PATTERN_LOWER	mb_bist_set_data_pattern_lower	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <BIST_DATA_PATTERN [159:128]> <BIST_DATA_PATTERN [127:96]> <BIST_DATA_PATTERN [95:64]> <BIST_DATA_PATTERN [63:32]> <BIST_DATA_PATTERN [31:0]>
CHANGE_FSP_LP5	mb_change_fsp_lp5	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <TARGET_FSP>
LP_MODE_ENTER	mb_lp_mode_enter	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <LP_STATE>
LP_MODE_EXIT	mb_lp_mode_exit	<calibration interface>
continued...		

Registers	Functions	Argumentss
		<CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID>
LP_MODE_AUTO	mb_lp_mode_auto	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID> <LP_STATE_AUTO> <LP_IDLE_CNT>
TRIG_MEM_CAL	mb_reset_cal	<calibration interface> <CMD_TARGET_IP_TYPE> <CMD_TARGET_IP_INSTANCE_ID>

Table 250. User Input Arguments Description

Arguments	Valid Input	Description
<calibration interface>	calip_0	Calibration interface instance 0
	calip_1	Calibration interface instance 1
<CMD_TARGET_IP_TYPE>	0x1, 0x2, 0x3, 0x4	Refer to the CMD_REQ Definition table.
<CMD_TARGET_IP_INSTANCE_ID>	Unique ID defined in EMIF IP	Refer to the CMD_REQ Definition table.
<ECC_ENABLE>	0x0, 0x1, 0x2, 0x3	Refer to the ECC_ENABLE_SET description in the Command Definitions table.
<ECC_INTERRUPT_MASK>	0x0 – 0xfff	Refer to the ECC_INTERRUPT_MASK description in the Command Definitions table.
<ECC_WRITEBACK_EN>	0x0, 0x1	Refer to the ECC_WRITEBACK_EN description in the Command Definitions table.
<ECC_XOR_CHECK_BITS>	Refer table 283	Refer to the ECC_INJECT_ERROR description in the Command Definitions table.
<ECC_SCRUB_INTERVAL>	User defined	Refer to the ECC_SCRUB_MODE_0_START description in the Command Definitions table.
<ECC_SCRUB_LEN>	0x8, 0x10	
<ECC_SCRUB_FULL_MEM>	0x0, 0x1	
<ECC_SCRUB_START_ADDR [31:0]> <ECC_SCRUB_START_ADDR [36:32]>	User defined	
<ECC_SCRUB_END_ADDR [31:0]> <ECC_SCRUB_END_ADDR [36:32]>	User defined	
<ECC_SCRUB_IDLE_CNT>	User defined	Refer to the ECC_SCRUB_MODE_1_START description in the Command Definitions table.
<BIST_ADDR_SPACE [5:0]>	User defined	Refer to the BIST_STANDARD_MODE_START description in the Command Definitions table.
<BIST_DATA_PATTERN>	0x0, 0x1	Refer to the BIST_MEM_INIT_START description in the Command Definitions table.

continued...

Arguments	Valid Input	Description
<BIST_START_ADDR [37:32]> <BIST_START_ADDR [31:0]>	User defined	Refer to the BIST_MEM_INIT_START description in the Command Definitions table.
<BIST_FULL_MEM>	0x0, 0x1	Refer to the BIST_MEM_INIT_START description in the Command Definitions table.
<BIST_DATA_PATTERN [287:256]> <BIST_DATA_PATTERN [255:224]> <BIST_DATA_PATTERN [223:192]> <BIST_DATA_PATTERN [191:160]>	User defined	Refer to the BIST_SET_DATA_PATTERN_UPPER description in the Command Definitions table.
<BIST_DATA_PATTERN [159:128]> <BIST_DATA_PATTERN [127:96]> <BIST_DATA_PATTERN [95:64]> <BIST_DATA_PATTERN [63:32]> <BIST_DATA_PATTERN [31:0]>	User defined	Refer to the BIST_SET_DATA_PATTERN_LOWER description in the Command Definitions table.
<TARGET_FSP>	0x0, 0x1, 0x2	Refer to the CHANGE_FSP_LP5 description in the Command Definitions table.
<LP_STATE>	0x8, 0x9, 0xa, 0xd, 0xe, 0xf	Refer to the LP_MODE_ENTER description in the Command Definitions table.
<LP_STATE_AUTO>	0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0xb, 0xc	Refer to the LP_MODE_AUTO description in the Command Definitions table.
<LP_IDLE_CNT>	User defined	

13.4.1.2. mb_user_input.tcl

This topic provides guidance for entering input using the `mb_user_input.tcl` script.

Follow these steps to perform function calls:

1. Identify the registers you want to access. Refer to the [Read-Only Registers](#) and [Command Definitions](#) tables for details.
2. Refer to the tables in the *mb_iossm_command.tcl* topic for the required arguments.
3. Call the respective functions with the necessary arguments, following the format below:

```
<function name> <lists of arguments separated by spaces>
```

Example 1: Accessing MEM_INTF_INFO_0 read-only register.

Function call: `intf_info calip_0`

Example 2: Accessing ECC_INJECT_ERROR mailbox command register.

Function call: `mb_ecc_inject_error calip_0 0x1 0x0 0xf4`

Example 3: Accessing ECC_ENABLE_SET mailbox command register.

Function call: `mb_ecc_enable_set calip_0 0x1 0x0 0x2`

13.4.2. Running the Mailbox Script

Follow these steps to run the mailbox script:

1. Configure the device with the `.sof` design file. (Ensure that the *Use Debug Toolkit* feature is enabled in the design.)
2. Download and extract all Mailbox Access Script files into the same directory.
3. Refer to the *mb_user_input.tcl* topic and modify the function calls as needed.
4. Save the file.
5. To run the script, use the following command:

```
system-console --script=mb_user_input.tcl  
--sof=<sof file directory>/<sof file name>.sof
```

(Ensure that you have configured the device with the same `.sof` design file that you specify in this command.)

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

Document Version	Quartus Prime Version	IP Version	Changes
2025.03.31	25.1	3.0.0	<ul style="list-style-type: none"> In the <i>Architecture</i> chapter: <ul style="list-style-type: none"> Added a section to the <i>Lockstep Configuration</i> topic. Reorganized and added content to the <i>EMIF IP for Hard Processor Subsystem (HPS)</i> section. In the <i>DDR4</i> chapter, modified the table contents in the <i>Maximum Number of Interfaces</i> topic. In the <i>DDR5</i> chapter, modified the table contents in the <i>Maximum Number of Interfaces</i> topic. In the <i>LPDDR4</i> chapter, modified the table contents in the <i>Maximum Number of Interfaces</i> topic. In the <i>LPDDR5</i> chapter, modified the table contents in the <i>Maximum Number of Interfaces</i> topic. In the <i>Controller Optimization</i> chapter, added the <i>Optimizing Efficiency for Secondary Controller</i> topic. In the <i>Debugging</i> chapter: <ul style="list-style-type: none"> Added a note to the <i>Debugging with the External Memory Interface Debug Toolkit</i> topic. Changed <i>LVDS</i> to <i>True Differential Signaling</i> in the <i>Debugging Checklist</i> topic. In the <i>Mailbox Support</i> chapter, added the <i>IOSSM Mailbox Access Script</i> section.
2025.01.13	24.3.1		<ul style="list-style-type: none"> In the <i>Architecture</i> chapter, added content to the <i>Agilex 5 EMIF IP for Hard Processor Subsystem (HPS)</i> topic. In the <i>DDR4</i> chapter: <ul style="list-style-type: none"> Added a detailed table to the <i>Maximum Number of Interfaces</i> topic. Modified the <i>Address and Command Pin Placement for DDR4</i> topic. Added figures to the <i>DDR4 Data Width Mapping</i> topic, and changed some terminology in the existing tables.. Added the <i>Clamshell Topology</i> topic. In the <i>DDR5</i> chapter: <ul style="list-style-type: none"> Added a detailed table to the <i>Maximum Number of Interfaces</i> topic. Modified the first two rows of the table in the <i>Address and Command Pin Placement for DDR5</i> topic. Added figures to the <i>DDR5 Data Width Mapping</i> topic.

continued...

Document Version	Quartus Prime Version	IP Version	Changes
			<ul style="list-style-type: none"> In the LPDDR4 chapter: <ul style="list-style-type: none"> Added a detailed table to the <i>Maximum Number of Interfaces</i> topic. In the LPDDR5 chapter: <ul style="list-style-type: none"> Added a detailed table to the <i>Maximum Number of Interfaces</i> topic. Added figures to the <i>LPDDR5 Data Width Mapping</i> topic. In the <i>Controller Optimization</i> chapter, added one bullet point to the <i>Improving Controller Efficiency</i> topic, and added the <i>AXI to Memory Mapping</i> topic. Added content to the <i>Hardware and Calibration Issues</i> section of the <i>Debugging</i> chapter. In the <i>Mailbox</i> chapter, made numerous changes to existing topics, added the <i>ECC Error Handling</i> topic, and revised the <i>Example 1</i> and <i>Example 2</i> topics.
2024.11.18	24.3	1.0.0	<ul style="list-style-type: none"> In the <i>Introduction</i> chapter, modified the last bullet point in the <i>Agilex 5 EMIF IP Protocol and Feature Support</i> topic. In the <i>About the External Memory Interfaces IP</i> chapter, updated the table of IPs and associated version numbers. In the <i>Product Architecture</i> chapter, modified the <i>Lockstep Configuration</i> topic. Moved the mailbox content from the <i>Architecture</i> chapter to a new chapter entitled <i>Agilex 5 5 FPGA EMIF IP - Mailbox Support</i>, later in the document. Recast the <i>End-User Signals</i> chapter to provide signals information for DDR4 Component, DDR4 DIMM, DDR5 Component, DDR5 DIMM, LPDDR4, and LPDDR5 interfaces. Recast the <i>DDR4 Support</i>, <i>DDR5 Support</i>, <i>LPDDR4 Support</i>, and <i>LPDDR5 Support</i> chapters with updated parameter information.

continued...

Document Version	Quartus Prime Version	IP Version	Changes
			<ul style="list-style-type: none"> Removed the <i>Layout Design Guidelines</i> sections from the DDR4, LPDDR4, and LPDDR5 chapters. This information now resides in the <i>Agilex 5 Printed Circuit Board (PCB) Design Guidelines (HSSI, EMIF, MIPI, PDN) User Guide</i>. Updated the mailbox content in the new <i>Agilex 5 FPGA EMIF IP - Mailbox Support</i> chapter. Updated several screenshots throughout.
2024.07.08	24.2	6.2.0	<ul style="list-style-type: none"> In the <i>Introduction</i> chapter: <ul style="list-style-type: none"> Added DDR5 to the list of supported protocols in the <i>Agilex 5 EMIF IP Protocol and Feature Support</i> topic. Added links to DDR5 content to the <i>Agilex 5 EMIF IP Design Checklist</i> topic. In the <i>Architecture</i> chapter: <ul style="list-style-type: none"> Added DDR5 to the <i>Agilex 5 EMIF Architecture: Protocol and Maximum Interface Width Support</i> and <i>Architecture: Introduction</i> topics. Added the <i>Lockstep Configuration</i> topic. Added <i>ECC_WRITEBACK_ENABLE</i> and <i>ECC_INJECT_ERROR</i> to the <i>Mailbox Supported Commands</i> and <i>Mailbox Command Definitions</i> topics. Added content to the <i>HPS EMIF Mapping</i> table in the <i>Agilex 5 EMIF IP for Hard Processor Subsystem (HPS)</i> topic. In the <i>End-User Signals</i> chapter: <ul style="list-style-type: none"> Added the <i>Interfaces for DDR5</i> section. Updated the DDR4, LPDDR4, and LPDDR5 sections as necessary. In the <i>DDR4</i> chapter: <ul style="list-style-type: none"> Updated the <i>Parameter Descriptions</i> section. Added tables to the <i>DDR4 Data Width Mapping</i> topic. Added the <i>x4 DIMM Implementation</i> topic. Modified the content of the <i>DDR4 Byte Lane Swapping</i> topic. Removed a reference from the <i>DDR4 Interface x4 Data Lane</i> topic. Added the <i>DDR5</i> chapter. In the <i>LPDDR4</i> chapter: <ul style="list-style-type: none"> Updated the <i>Parameter Descriptions</i> section. Made minor terminology changes in the <i>LPDDR4 Component Options</i> topic. Made changes to the steps in the <i>General Guidelines</i> topic. Renamed the <i>Pin Placements for Agilex 5 FPGA EMIF IP for LPDDR4</i> topic to <i>LPDDR4 Data Width Mapping</i>. In the <i>LPDDR5</i> chapter: <ul style="list-style-type: none"> Updated the <i>Parameter Descriptions</i> section. Removed the former step 8 from the <i>General Guidelines - LPDDR5</i> topic. In the <i>Debugging</i> chapter, added the <i>Debugging with the External Memory Interface Debug Toolkit</i> section.
2024.04.01	24.1	6.1.0	Initial release.