Trion® DDR DRAM Block User Guide

UG-DDR-DRAM-v1.7
April 2022
www.efinixinc.com

Copyright © 2022. All rights reserved. Efinix, the Efinix logo, the Titanium logo, Quantum, Trion, and Efinity are trademarks of Efinix, Inc. All other trademarks and service marks are the property of their respective owners. All specifications subject to change without notice.

Contents

Introduction.. 3
  Features..3
  Trion FPGAs with DDR DRAM.. 4

Functional Description..4
  DDR PHY..5
  DDR Controller.. 5
  AXI Interface..6
    About the atype Signal..6
    Interface Mappings..7
    AXI Signals..8
    Clock Domain Synchronization FIFO Depths.. 9
    Narrow Transfers..10
  Arbiter.. 11
  Configuration Registers..11
  I^{2}C Interface.. 11
  Startup Sequencer.. 12

Address Mapping.. 13

Clocking..14

Reset.. 15
  Reset with the DDR Reset Controller Core.. 16

Performing Auto-Calibration.. 17

Manual Gate Delay Tuning..18

DDR Interface Designer Settings.. 20

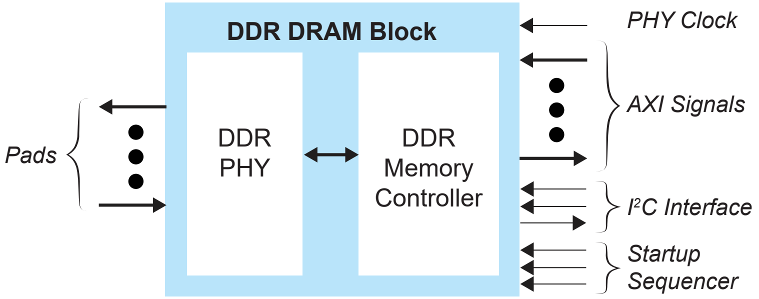
Supported DDR DRAM Modules..23

Revision History..24

Introduction

The Trion® DDR DRAM hardened memory controller and PHY provides a robust and complete solution to implement an external memory interface to a DRAM module. Because the block is hardened, it provides a power and area efficient solution.

Figure 1: DDR DRAM Block Diagram



Features

- Supports multiple protocols: LPDDR2, LPDDR3, DDR3, and DDR3L
- DDR PHY supports data rates up to 1066 Mbps per lane
- Has programmable termination with calibration to compensate for variations across power, voltage, and temperature (PVT)
- Read and write leveling and gate training to optimize performance
- User configurable data width support: x8, x16, or x32 depending on the FPGA and package
- User configurable timing parameters
- Supports up to 2 pseudo-AXI interfaces with multiple accesses
  - Data width up to 256 bits to allow a flexible system interconnect to the FPGA core
  - Includes a built-in arbiter to manage memory read and write requests from multiple sources
- Access to the controller's configuration registers through an I^{2}C interface
- Power management: automatically enters into active power down, pre-charge power down, and self-refresh mode

Table 1: Supported DDR PHY Frequency

Table shows the supported frequency for DDR PHY and not the actual bandwidth. You need to validate the actual system bandwidth in your system.

DDR DRAM Interface	**Supported Frequency (MHz)**	
	Minimum	**Maximum**
DDR3	300	533
DDR3L	300	533
LPDDR3	75	533
LPDDR2	75	533

Trion FPGAs with DDR DRAM

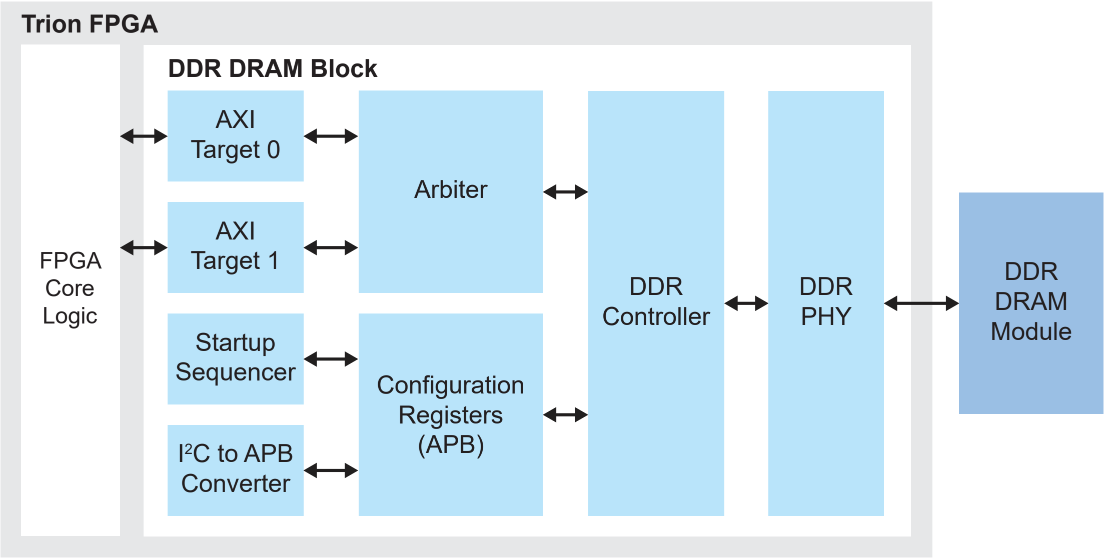
Table 2: DDR DRAM Widths Supported for FPGAs and Packages

FPGA	Package	DDR DRAM Support	DQ Widths
T20, T35	BGA324	1 block	x8 or x16
	BGA400	1 block	x8 or x16
T55, T85, T120	BGA324	1 block	x16
	BGA484	1 block	x16 or x32
	BGA576	1 block	x16 or x32

Functional Description

The DDR DRAM block includes the elements shown in the following figure:

Figure 2: DDR DRAM Block Biagram


Trion FPGA
DDR DRAM Block
FPGA Core Logic
AXI Target 0
AXI Target 1
Startup Sequencer
I²C to APB Converter
Arbiter
Configuration Registers (APB)
DDR Controller
DDR PHY
DDR DRAM Module

DDR PHY

The DDR PHY is a physical layer that interfaces with the external DDR DRAM module. It is fully compliant with the DDR3, LPDDR3, and LPDDR2 electrical specifications. DDR PHY has a built-in data training circuit to enable in system calibration, which helps optimize the system timing for high performance.

The PHY connects to the DRAM module using the signals in the following table.

Table 3: DDR DRAM Pads

Signal	Direction	Description
DDR_A[15:0]	Output	Address signals to the memories.
DDR_BA[2:0]	Output	Bank signals to/from the memories.
DDR_CAS_N	Output	Active-low column address strobe signal to the memories.
DDR_CKE	Output	Active-high clock enable signals to the memories.
DDR_CK	Output	Active-high clock signals to/from the memories. The clock to the memories and to the memory controller must be the same clock frequency and phase.
DDR_CK_N	Output	Active-low clock signals to/from the memories.The clock to the memories and to the memory controller must be the same clock frequency and phase.
DDR_CS_N	Output	Active-low chip select signals to the memories.
DDR_DQ[*n*:0]	Bidirectional	Data bus to/from the memories. For writes, the pad drives these signals. For reads, the memory drives these signals. These signals are connected to the DQ inputs on the memories. *n* is 7, 15, or 31 depending on the FPGA and DQ width.
DDR_DM[*n*]	Output	Active-high data-mask signals to the memories. *n* is 1, 1:0, or 3:0 depending on the FPGA and DQ width.
DDR_DQS_N[*n*:0]	Bidirectional	Differential data strobes to/from the memories. For writes, the pad drives these signals. For reads, the memory drives these signals. These signals are connected to the DQS inputs on the memories. *n* is 1, 1:0, or 3:0 depending on the FPGA and DQ width.
DDR_DQS[*n*:0]	Bidirectional	
DDR_ODT	Output	ODT signal to the memories.
DDR_RAS_N	Output	Active-low row address strobe signal to the memories.
DDR_RST_N	Output	Active-low reset signals to the memories.
DDR_WE_N	Output	Active-low write enable strobe signal to the memories.
DDR_VREF	Bidirectional	Reference voltage.
DDR_ZQ	Bidirectional	ZQ calibration pin.

DDR Controller

The DDR controller block handles all of the protocols and hand-shaking between the external memory interface and the PHY. It reduces latency and minimizes core logic consumption for external memory interface.

AXI Interface

The DDR DRAM block has 2 pseudo-AXI interfaces (target 0 and target 1) that provide an easy, efficient way to access the external memory device. The interface is half-duplex, which minimizes the number of signals required to interface with the core.

To reduce the number of bits needed for separate write and read address buses, the pseudo-AXI protocol uses a signal (`atype`) to indicate a read or write. This signal allows the DDR DRAM block to combine the read and write address buses and signals. All other AXI signals and operations comply with the AMBA AXI 1.0 specification.

The pseudo-AXI interface supports:

- A clock frequency that is asynchronous to the controller clock frequency; a FIFO handles the synchronization between the pseudo-AXI clock domain and the DDR controller clock domain
- Burst sizes of 1 – 32 bytes
- Burst lengths of 1 – 256 for data transfer
- Read and write strobes (data masking)
- Incrementing and wrapping burst types
- Narrow bus transfers

About the atype Signal

The `atype` signal indicates whether to perfom a read or write. When `avalid` is asserted:

- `atype` = 1 means a write operation
- `atype` = 0 means a read operation

With this `atype` signal, the addressing operation becomes half duplex. The memory interface is also half duplex, so the bandwidth is not affected.

Figure 3: Read Operation Example with atype

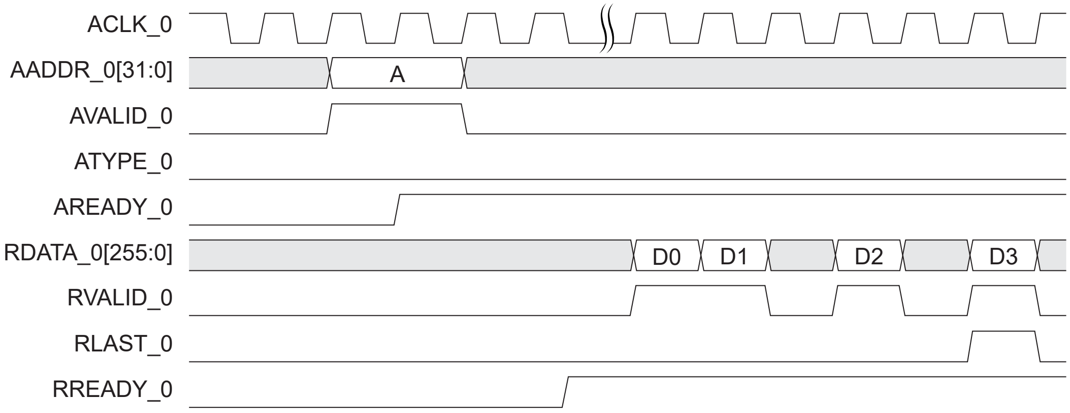
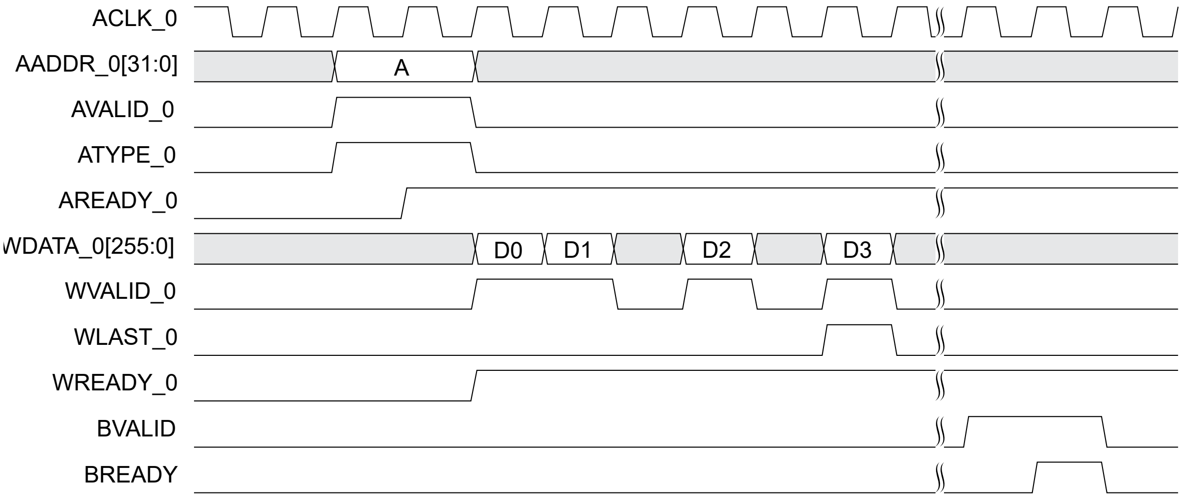


Figure 4: Write Operation Example with atype


ACLK_0
AADDR_0[31:0]
A
AVALID_0
ATYPE_0
AREADY_0
WDATA_0[255:0]
D0
D1
D2
D3
WVALID_0
WLAST_0
WREADY_0
BVALID
BREADY

Interface Mappings

Table 4: Data Width and Burst Mapping by FPGA

FPGA	DQ Width	AXI Target 0	AXI Target 1
T55, T85, T120	x32	256-bit data width with 1 access for a DRAM chip x32, 8 bursts	128-bit data width with 2 accesses for a DRAM chip x32, 8 bursts
	x16	256-bit data width with 1 access for a DRAM chip x16, 16 bursts	128-bit data width with 1 access for a DRAM chip x16, 8 bursts
T20, T35	x16	128-bit data width with 1 access for a DRAM chip x16, 8 bursts	128-bit data width with 1 access for a DRAM chip x16, 8 bursts

Table 5: AXI Data to DDR Module DQ Mapping Part 1

AXI	[255:240]	[239:224]	[223:208]	[207:192]	[191:176]	[175:160]	[159:144]	[143:128]
Burst	8	8	7	7	6	6	5	5
DQ	[31:16]	[15:0]	[31:16]	[15:0]	[31:16]	[15:0]	[31:16]	[15:0]

Table 6: AXI Data to DDR Module DQ Mapping Part 2

AXI	[127:112]	[111:96]	[95:80]	[79:64]	[63:48]	[47:32]	[31:16]	[15:0]
Burst	4	4	3	3	2	2	1	1
DQ	[31:16]	[15:0]	[31:16]	[15:0]	[31:16]	[15:0]	[31:16]	[15:0]

AXI Signals

Table 7: AXI Gobal Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
ACLK_0, ACLK_1	Input	N/A	AXI clock inputs.

Table 8: AXI Shared Read/Write Signals (Interface to FPGA Fabric)

Signal x is 0 or 1	Direction	Clock Domain	Description
AADDR_*x*[31:0]	Input	ACLK_x	Address. ATYPE defines whether it is a read or write address. It gives the address of the first transfer in a burst transaction.
ABURST_*x*[1:0]	Input	ACLK_x	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated.
AID_*x*[7:0]	Input	ACLK_x	Address ID. This signal identifies the group of address signals. Depends on ATYPE, the ID can be for a read or write address group.
ALEN_*x*[7:0]	Input	ACLK_x	Burst length. This signal indicates the number of transfers in a burst.
ALOCK_*x*[1:0]	Input	ACLK_x	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
AREADY_*x*	Output	ACLK_x	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
ASIZE_*x*[2:0]	Input	ACLK_x	Burst size. This signal indicates the size of each transfer in the burst.
ATYPE_*x*	Input	ACLK_x	This signal distinguishes whether is it is a read or write operation. 0 = read and 1 = write.
AVALID_*x*	Input	ACLK_x	Address valid. This signal indicates that the channel is signaling valid address and control information.

Table 9: AXI Read Data Channel Signals (Interface to FPGA Fabric)

Signal x is 0 or 1	Direction	Clock Domain	Description
RDATA_*x*[127:0]	Output	ACLK_x	(T20, T35): Read data.
RDATA_0[255:0]	Output	ACLK_0	(T55, T85, T120): AXI target 0 read data.
RDATA_1[127:0]	Output	ACLK_1	(T55, T85, T120): AXI target 1 read data.
RID_*x*[7:0]	Output	ACLK_x	Read ID tag. This signal is the identification tag for the read data group of signals generated by the slave.
RLAST_*x*	Output	ACLK_x	Read last. This signal indicates the last transfer in a read burst.
RREADY_*x*	Input	ACLK_x	Read ready. This signal indicates that the master can accept the read data and response information.
RRESP_*x*[1:0]	Output	ACLK_x	Read response. This signal indicates the status of the read transfer.
RVALID_*x*	Output	ACLK_x	Read valid. This signal indicates that the channel is signaling the required read data.

Table 10: AXI Write Data Channel Signals (Interface to FPGA Fabric)

Signal x is 0 or 1	Direction	Clock Domain	Description
WDATA_*x*[127:0]	Input	ACLK_x	(T20, T35): Write data.
WDATA_0[255:0]	Input	ACLK_0	(T55, T85, T120): AXI target 0 write data.
WDATA_1[127:0]	Input	ACLK_1	(T55, T85, T120): AXI target 1 write data.
WID_*x*[7:0]	Input	ACLK_x	Write ID tag. This signal is the ID tag of the write data transfer.
WLAST_*x*	Input	ACLK_x	Write last. This signal indicates the last transfer in a write burst.
WREADY_*x*	Output	ACLK_x	Write ready. This signal indicates that the slave can accept the write data.
WSTRB_*x*[15:0]	Input	ACLK_x	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus.
WSTRB_0[31:0] WSTRB_1[15:0]	Input	ACLK_x	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus.
WVALID_*x*	Input	ACLK_x	Write valid. This signal indicates that valid write data and strobes are available.

Table 11: AXI Write Response Channel Signals (Interface to FPGA Fabric)

Signal x is 0 or 1	Direction	Clock Domain	Description
BID_*x*[7:0]	Output	ACLK_x	Response ID tag. This signal is the ID tag of the write response.
BREADY_*x*	Input	ACLK_x	Response ready. This signal indicates that the master can accept a write response.
BVALID_*x*	Output	ACLK_x	Write response valid. This signal indicates that the channel is signaling a valid write response.

Clock Domain Synchronization FIFO Depths

Table 12: Clock Domain Synchronization FIFO Depths

FIFO	AMBA Naming[1]	Depth
AXI address channel FIFO	AXI A	4
AXI write channel FIFO	AXI W	4
AXI read channel FIFO	AXI R	4
AXI respond channel FIFO	AXI B	4
DDR controller command FIFO	-	8
DDR controller read FIFO	-	16
DDR controller write FIFO	-	16

---

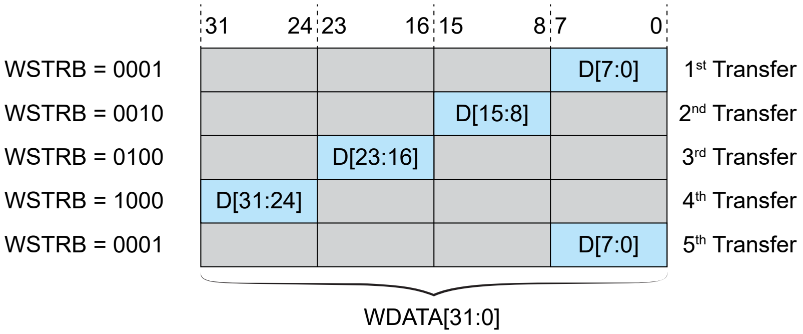
(1) The AXI interface is half duplex.

Narrow Transfers

AXI DDR implementation supports narrow transfers. It is important to set the `WSTRB_x[n:0]` signals correctly when performing narrow transfers. There is one write strobe for each eight bits of the write data bus. A master must ensure that the write strobes are set to logic high for byte lanes that contain valid data.

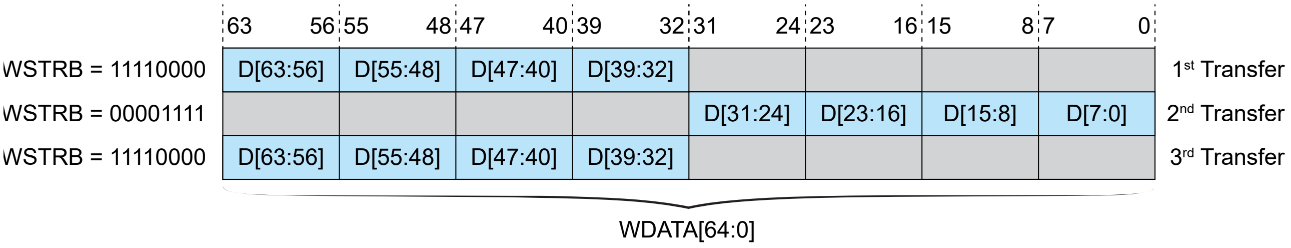
The following figure illustrate an example for a five incrementing burst of 8-bit transfer on a 32-bit bus with a starting address of `0`.

Figure 5: 8-bit Transfer on a 32-bit Bus Example


31 24 23 16 15 8 7 0
WSTRB = 0001
D[7:0]
1st Transfer
WSTRB = 0010
D[15:8]
2nd Transfer
WSTRB = 0100
D[23:16]
3rd Transfer
WSTRB = 1000
D[31:24]
4th Transfer
WSTRB = 0001
D[7:0]
5th Transfer
WDATA[31:0]

The following figure illustrate an example for a three incrementing burst of 32-bit transfer on a 64-bit bus with a starting address of 4.

Figure 6: 32-bit Transfer on a 64-bit Bus Example


63 56 55 48 47 40 39 32 31 24 23 16 15 8 7 0
WSTRB = 11110000
D[63:56] D[55:48] D[47:40] D[39:32]
1st Transfer
WSTRB = 00001111
D[31:24] D[23:16] D[15:8] D[7:0]
2nd Transfer
WSTRB = 11110000
D[63:56] D[55:48] D[47:40] D[39:32]
3rd Transfer
WDATA[64:0]

Learn more: Refer to the AMBA AXI and ACE Protocol Specification for more information about narrow transfer transaction structure.

Arbiter

The arbiter handles simultaneous read and write requests from the AXI `0` and 1 interfaces, and decides which request to send to the memory. The arbiter allows AXI target `0` and target 1 to access the memory simultaneously.

During arbitration, the arbiter applies these rules in order:

1. Ensure a minimum number of consecutive accesses from a target if the target is accessing consecutive address locations.
2. Ensure a minimum share of bandwidth for each target. If a target does not receive its fair share of the bandwidth, give it higher priority.
3. Optimize the bandwidth by giving priority to the following operations (and their combinations):
    - Accesses to open pages or rows.
    - Stringing reads and writes together.
    - Accesses to banks that will be closed or were closed earlier (bank rotation).
4. Prioritize earlier requests not later ones.

Configuration Registers

The configuration registers store the settings required for the DDR DRAM block and external memory device to operate properly. During FPGA configuration, these configuration registers are programmed automatically. In user mode, these resigters are accessible through the I^{2}C interface.

I^{2}C Interface

During user mode, the I^{2}C interface provides access to the configuration registers. This interface supports:

- Initiating training routines to optimize the performance of the PHY
- Reconfiguring the PHY and controller settings on the fly
- Issuing commands to re-initialize, reset, or power down the memory module
- Performing hardware diagnosis

Efinix provides the Auto-Calibration Core, which uses this interface to calibrate and reset the memory module. Refer to Performing Auto-Calibration on page 17 for more information.

Table 13: DDR DRAM I^{2}C Interface Signals

Signal	Direction	Description
CFG_SCL_IN	Input	Clock input.
CFG_SDA_IN	Input	Data input.
CFG_SDA_OEN	Output	SDA output enable.

Startup Sequencer

The startup sequencer facilitates the operations required to start the DDR DRAM block. It performs these tasks:

1. Configures the correct settings into the DDR PHY and controller.
2. Initializes the DRAM module by programming the corrrect instruction sequence into the controller.
3. Configures the DRAM module's mode registers settings (MRS).

The startup sequencer performs these tasks while the FPGA is configuring; it completes them before the FPGA enters user mode.

When the FPGA enters user mode, the controller and DRAM module are fully initialized. At that point you can initiate calibration to optimize memory performance or, if the performance of the memory is not critical, you can skip calibration and start performing memory operations.

Table 14: DDR DRAM Startup Sequencer Signals

Signal	Direction	Description
CFG_SEQ_RST	Input	Active-high DDR configuration controller reset.
CFG_SEQ_START	Input	Start the DDR configuration controller.

Address Mapping

From the user perspective, the AXI address space is one continuous byte address space starting from byte `0` up to the size limit of the memory module. Behind the scenes, the DDR controller accesses the memory module using bank, column, and row addresses. The data width accessed by each address space is dependent on DDR controller's data width. For example, a x32 data width controller accesses 4 bytes of memory data for each address.

The relationship between user AXI address and DDR controller address is

x32 data width	DDR controller address = AXI address >> 2
x16 data width	DDR controller address = AXI address >> 1
x8 data width	DDR controller address = AXI address

where > > is a Verilog HDL operator that denotes a bit-wise right shift.

So for x32, the least significant bit (LSB) of the DDR controller address is bit 2 (starting from `0`) of the AXI address. For x16, the LSB of the controller address is bit 1 of AXI address.

The address mapping you choose depends on your memory module and application. Some mappings are more efficient than others, so you need to choose which mapping you want to use based on your system requirements. The DDR DRAM block supports three mappings:

- ROW, BA, COL
- BA, ROW, COL
- ROW, COL_HIGH, BA, COL_LOW

Where:

- *BA*—Bank address bits
- *ROW*—Row address bits
- *COL*—Column address bits
- *COL_HIGH*—MSBs of the column address bits when the column address is divided into 2
- *COL_LOW*—LSBs of the column address bits when the column address is divided into 2

By default, the DDR DRAM block uses the ROW, COL_HIGH, BA, COL_LOW address mapping. You can change the mapping in the DDR block's **Advanced Options** > **Controller Settings** tab in the Interface Designer.

Note: Different memory modules have different bus widths for row and column addresses; therefore, the usable address range is also different.

The following table shows some example settings if you choose ROW, COL_HIGH, BA, COL_LOW.

Table 15: AXI Logical Address to DDR Module Mapping

i = number of columns

Module	Setting
DDR3	ROW-COL[*i*-1:3]-BA-COL[2:0]
LPDDR2/3	Number of columns > 10: ROW-COL[*i*-1:7]-BA-COL[6:0] Number of columns < =10: ROW-BA-COL[*i*-1:0]

The following example translates the AXI address to the controller address and details the row, column, bank address for the memory module.

Table 16: Address Mapping Example for LPDDR3 Module (x32, 4 Gb)

Setting	Mapping
Controller address mapping	row_address[13:0], bank_address[2:0], col_address[9:0] (assuming ROW, BA, COL mapping)
AXI address	32’h01000100
Controller address (first address)(2)	32’h00400040
Bank address (first address)	3’h0
Row address (first address)	14’h0200
Column address (first address)	10’h040

Clocking

The DDR DRAM requires a clock source that runs at half the memory data rate. For example, an 800 Mbps DDR3 interface requires a 400 Mhz clock.

Efinix designed the PLL_BR0 `OUTPUT0` clock to be the DDR DRAM clock source. When using the DDR DRAM block,you must also instantiate PLL_BR0 and set the output frequency of `OUTPUT0` to half of the operating data rate.

Table 17: PHY Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
CLKIN	Input	N/A	High-speed clock to drive the DDR PHY. A PLL must generate this clock. The clock runs at half of the PHY data rate (for example, 800 Mbps requires a 400 MHz clock). The DDR DRAM block uses the PLL_BR0 CLKOUT0 resource as the PHY clock.

Important: Efinix strongly recommends that you do not use any LVDS pins (either single-ended I/O or differential pair) as the primary clock to drive the PLL_BR0 or the DDR interface will not be initialized during the configuration phase. Make sure to incorporate a user reset and instantiate the DDR Hard Memory Controller-Reset IP to initialize the DDR interface in user mode. Contact Efinix support if you need LVDS pins as the primary clock for the PLL_BR0 DDR interface

---

(2) The controller address is internally incremented along each AXI transaction to align with the total number of bytes being transmitted.

Reset

The DDR DRAM block has an optional reset pin that lets you perform a full system reset of the AXI interfaces, configuration registers, DDR controller, DDR PHY, and the external memory device.

After reset, you need to program the configuration registers and initialize the memory before performing any memory operations. Use one of these methods:
- Use the Efinix DDR Reset Controller Core, which you can download from the Support Center. This core triggers the startup sequencer, which perform the steps required to start up the memory.
- Use the Efinix Auto-Calibration Core, which transmits signals through the I^{2}C interface to calibrate the memory module.

Figure 7: DDR DRAM Reset Block Biagram

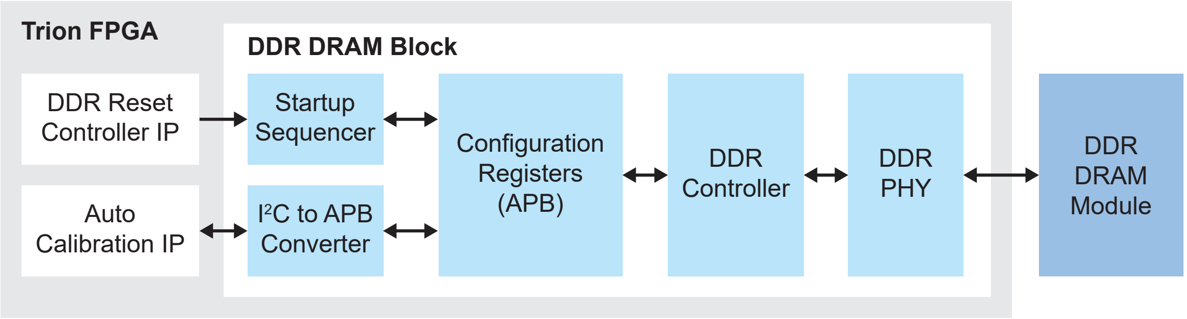

Trion FPGA
DDR DRAM Block
DDR Reset Controller IP
Auto Calibration IP
Startup Sequencer
I²C to APB Converter
Configuration Registers (APB)
DDR Controller
DDR PHY
DDR DRAM Module

Table 18: DDR DRAM Reset Signal

Signal	Direction	Description
CFG_RST_N	Input	Active-low master DDR DRAM reset. After you de-assert RST_N, you need to reconfigure and initialize before performing memory operations.

Reset with the DDR Reset Controller Core

The DDR interface block has three input pins for reset control:

- Master reset (active low)
- Sequencer reset (active high)
- Sequencer start (active high)

Note: These reset pins are not available if you are using I^{2}C calibration.

You provide a single reset signal and a clock to the core, and it generates outputs that drive these reset pins. Optionally, the code generates a status signal. Your system can monitor the status to know when the reset and DDR re-initialization completes and that read/write operations to the DDR AXI interfaces can resume.

The core contains one parameter, `FREQ`, which should correspond to the `clk` signal's frequency.

Note: You do not need to reset the DDR block when the FPGA configures or when the FPGA initially goes into user mode. The DDR reset and initialization is triggered automatically during configuration.

Figure 8: Core Block Diagram

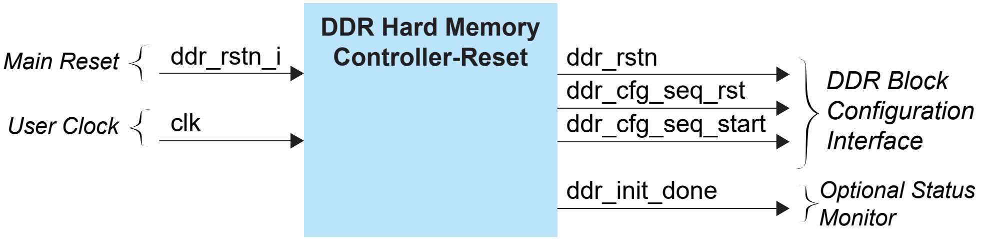
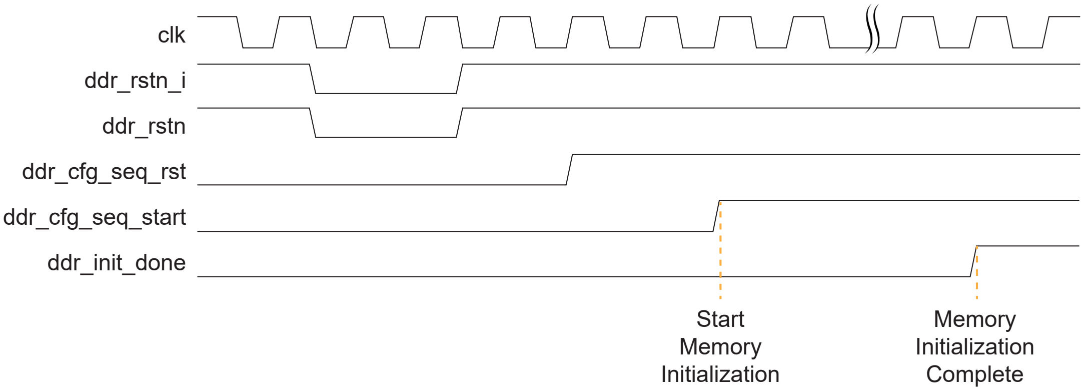

Main Reset
User Clock
ddr_rstn_i
clk
DDR Hard Memory Controller-Reset
ddr_rstn
ddr_cfg_seq_rst
ddr_cfg_seq_start
ddr_init_done
DDR Block Configuration Interface
Optional Status Monitor

Figure 9: Reset Sequencer Example Waveform


clk
ddr_rstn_i
ddr_rstn
ddr_cfg_seq_rst
ddr_cfg_seq_start
ddr_init_done
Start Memory Initialization
Memory Initialization Complete

Learn more: Refer to the DDR Reset Controller Core User Guide for more information about the core.

Performing Auto-Calibration

The DDR protocol requires some or all of the following training operations:

- *Write leveling*—Tunes the `DQS` signal so that it is aligned with `CK` during write operations.
- *Read leveling*—Tunes the `DQS` signal to be at the center of the `DQ` eye during read operations.
- *CA training*—Compensates the skew on the command and address buses so that they can be correctly captured by the memory module.
- *Gate training*—Tunes the internal gate enabled signal to be at the center of the `DQS` preamble.

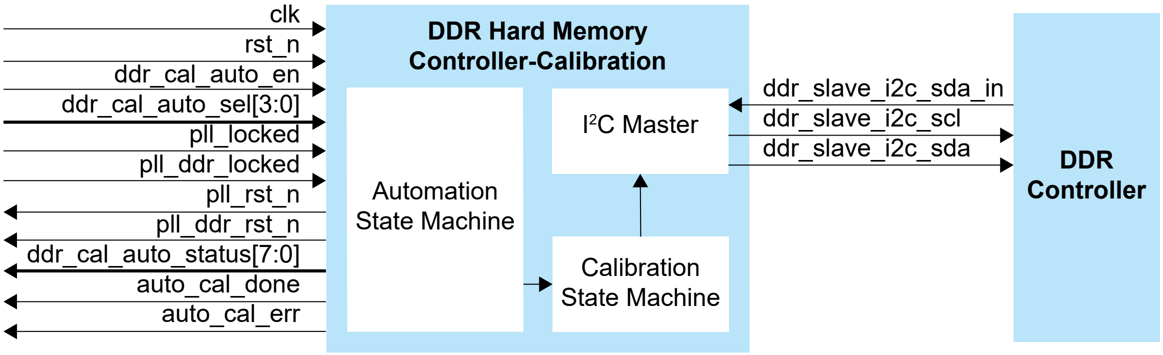
The following table shows the calibration operations each DDR protocol requires:

Table 19: Required Calibration Operations

Operation	LPDDR2	LPDDR3	DDR3
Write leveling	-	✓	✓
Read leveling	✓	✓	✓
Gate training	✓	✓	✓
CA training	-	✓	-

The core consists of an automation state machine, a calibration state machine, and an I^{2}C master.

Figure 10: Core Block Diagram


clk
rst_n
ddr_cal_auto_en
ddr_cal_auto_sel[3:0]
pll_locked
pll_ddr_locked
pll_rst_n
pll_ddr_rst_n
ddr_cal_auto_status[7:0]
auto_cal_done
auto_cal_err
DDR Hard Memory Controller-Calibration
I²C Master
Automation State Machine
Calibration State Machine
ddr_slave_i2c_sda_in
ddr_slave_i2c_scl
ddr_slave_i2c_sda
DDR Controller

Manual Gate Delay Tuning

The `DQ` data bus and `DQS` strobe are bidirectional signals that send/receive data to/from the memory module. For proper read and write operation, the DDR controller must switch between read and write mode at the correct time. During a read operation, the DDR controller turns on the `DQS` gate to let the read data in; conversely, it turns off the `DQS` gate for write mode.

Note: If you are using the Auto-Calibration Core, the calibration state machines automatically determine the optimized DQS gating delay value, and you do not need to perform manual gate delay tuning.

The DDR DRAM block has an internal `DQS` gating circuit to manage switches between read and write mode. The DDR controller issues a gate enable signal during read operations. It is crucial to determine the accurate timing window so that the DDR controller captures valid read data. The following figures illustrate the concept of `DQS` gating and the window needed to capture the correct `DQS`.

Figure 11: Conceptual DQS Gating Logic

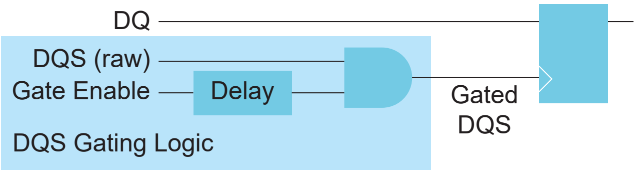
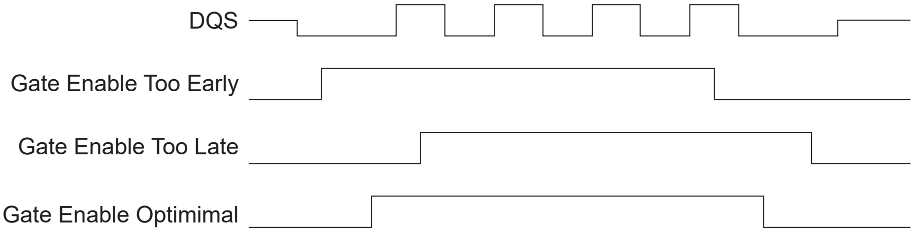


Figure 12: Conceptual Gate Enable Waveform



You manually adjust the gate enable delay in the Interface Designer. This manual process is useful for debug or initial system bring up.

Important: Turning on manual gate delay tuning overrides the settings obtained when you perform auto-calibration. Do not turn on manual gate delay tuning if you are using auto calibration.

In the Interface Designer, you can adjust the gate coarse delay and the gate fine delay.

Adjusting the Gate Coarse Delay

The gate coarse delay gives the DDR controller a rough idea of the round trip delay from the PHY to the memory module. Each gate coarse delay step introduces a half clock cycle delay to the gate enable signal.

The round trip delay is given by following formula:

<Round-trip delay> = *<Board delay for CK (PHY to module)>* + *<board delay for DQS (module to PHY)>* + t_{DQSCK}

where t_{DQSCK} is the skew between `DQS` and `CK`; you obtain this timing specification from the memory module's data sheet or other documentation.

After you determine the round-trip delay, set the gate coarse delay value according to the following table:

Table 20: Gate Coarse Delay Settings

Minimum External Delay (Clock Cycles)	Maximum External Delay (Clock Cycles)	Suggested Value for Gate Coarse Delay
0	0.5	0
>0.5	1	1
>1	1.5	2
>1.5	2	3
>2	2.5	4
>2.5	3	5

Adjusting the Gate Fine Delay

The gate fine delay is an additional adjustment you make after setting the gate coarse delay. Each step of gate fine delay introduces an additional 1/256 clock cycle delay. The gate fine delay is added to the gate coarse delay.

Recommended Settings for LPDDR2 and LPDDR3

	LPDDR2	LPDDR3
Gate Coarse Delay Tuning	2	3
Gate Fine Delay Tuning	91	91

DDR Interface Designer Settings

The following tables describe the settings for the DDR block in the Interface Designer.

Table 21: Base Tab

Parameter	Choices	Notes
DDR Resource	None, DDR_0	Only one resource available.
Instance Name	User defined	Indicate the DDR instance name. This name is the prefix for all DDR signals.
Memory Type	DDR3, LPDDR2, LPDDR3	Choose the memory type you want to use.

Table 22: Configuration Tab

Parameter	Choices	Notes
Select Preset		The **Select Preset** button opens a list of popular DDR memory configurations. Choose a preset to populate the configuration choices. If you do not want to use a preset, you can specify the memory configuration manually.
DQ Width	x8, x16 x8, x16, x32	DQ bus width. The width choices vary depending on the FPGA and package.
Type	DDR3, LPDDR2, LPDDR3	Memory type.
DDR3		
Speed Grade	1066E, 1066F, 1066G, 800D, 800E	Memory speed.
Width	x8, x16	Memory width.
Density	1G, 2G, 4G, 8G	Memory density in bits.
LPDDR2		
Speed Grade	400, 533, 667, 800, 1066	Memory speed.
Width	x16 x16, x32	Memory width. The width choices vary depending on the FPGA and package.
Density	256M, 512M, 1G, 2G, 4G	Memory density in bits.
LPDDR3		
Speed Grade	800, 1066	Memory speed.
Width	x16 x16, x32	Memory width. The width choices vary depending on the FPGA and package.
Density	4G, 8G	Memory density in bits.

Table 23: Advanced Options Tab - FPGA Setting Subtab

Parameter	Choices	Notes
FPGA Input Termination	Varies depending on the memory type	Specify the termination value for the FPGA input/ output pins.
FPGA Output Termination		

Table 24: Advanced Options Tab - Memory Mode Register Settings Subtab

Parameter	Choices	Notes
DDR3		
Burst Length	8	Specify the burst length (only 8 is supported).
DLL Precharge Power Down	On, Off	Specify whether the DLL in the memory device is off or on during precharge power-down.
Memory Auto Self-Refresh	Auto, Manual	Turn on or off auto-self refresh feature in memory device.
Memory CAS Latency (CL)	5 - 14	Specify the number of clock cycle between read command and the availability of output data at the memory device.
Memory Write CAS Latency (CWL)	5 - 12	Specify the number of clock cycle from the releasing of the internal write to the latching of the first data in at the memory device.
Memory Dynamic ODT (Rtt_WR)	Off, RZQ/2, RZQ/4	Specify the mode of dynamic ODT feature of memory device.
Memory Input Termination (Rtt_nom)	Off, RZQ/2, RZQ/4, RZQ/6, RZQ/8, RZQ/12	Specify the input termination value of the memory device.
Memory Output Termination	RZQ/6, RZQ/7	Specify the output termination value of the memory device.
Read Burst Type	Interleaved, Sequential	Specify whether accesses within a give burst are in sequential or interleaved order.
Sef-Refresh Temperature	Extended, Normal	Specify whether the self refresh temperature is normal or extended mode.
LPDDR2		
Burst Length	8	Specify the burst length (only 8 is supported).
Output Drive Strength	34.3, 40, 48, 60, 80, 120	Specify the output termination value of memory device.
Read Burst Type	Interleaved, Sequential	Specify whether accesses within a given burst are in sequential or interleaved order.
Read/Write Latency	RL=3/WL=1, RL=4/WL=2 RL=5/WL=2, RL=6/WL=3 RL=7/WL=4, RL=8/WL=4	Specify the read/write latency of the memory device.
LPDDR3		
DQ ODT	Disable, RZQ1, RZQ2, RZQ4	Specify the input termination value of memory device.
Output Drive Strength	34.3 34.3 pull-down/40 pull up 34.3 pull-down/48 pull up 40 40 pull down/48 pull up 48	Specify the output termination value of memory device.
Read/Write Latency	RL=3/WL=1, RL=6/WL=3 RL=8/WL=4, RL=9/WL=5	Specify the read/write latency of the memory device.

Table 25: Advanced Options Tab - Memory Timing Settings Subtab

Parameter	Choices	Notes
tFAW, Four Bank Active Window (ns)	User defined	Enter the timing parameters from the memory device's data sheet.
tRAS, Active to Precharge Command Period (ns)		
tRC, Active to Actrive or REF Command Period (ns)		
tRCD, Active to Read or Write Delay (ns)		
tREFI, Average Periodic Refresh Interval (ns)		
tRFC, Refresh to Active or Refresh to Refresh Delay (ns)		
tRP, Precharge Command Period (ns)		
tRRD, Active to Active Command Period (ns)		
tRTP, Internal Read to Precharge Delay (ns)		
tWTR, Internal Write to Read Command Delay (ns)		

Table 26: Advanced Options Tab - Controller Settings Subtab

Parameter	Choices	Notes
Controller to Memory Address Mapping	BANK-ROW-COL ROW-BANK-COL ROW-COL_HIGH-BANK-COL_LOW	Specify the mapping between the address of AXI interface and column, row, and bank address of memory device.
Enable Auto Power Down	Active, Off, Pre-Charge	Specify whether to allow automatic entry into power-down mode (pre-charge or active) after a specific amount of idle time.
Enable Self Refresh Controls	No, Yes	Specify whether to enable automatic entry into self-refresh mode after specific amount of idle period.

Table 27: Advanced Options Tab - Gate Delay Tuning Settings Subtab

Parameter	Choices	Notes
Enable Gate Delay Override	On or off	Turning this option on allows you to fine-tine the gate-delay values. This is an expert only setting.
Gate Coarse Delay Tuning	0 - 5	
Gate Fine Delay Tuning	0 - 255	

Table 28: Control Tab

Option	Notes
Disable Control	When selected, this option disables calibration and user reset.
Enable Calibration	Turn on to enable optional PHY calibration pins (master reset, SCL, and SDA pins). Efinix recommends that you use the default pin names. The names are prefixed with the instance name you specified in the Base tab.
User Reset	Turn on to enable optional reset pins (master reset and sequencer start/reset). Efinix recommends that you use the default pin names. The names are prefixed with the instance name you specified in the Base tab.

Table 29: AXI 0 and AXI 1 Tabs

Parameter	Choices	Notes
Enable Target 0 Enable Target 1	On or off	Turn on to enable the AXI 0 interface. Turn on to enable the AXI 1 interface.
AXI Clock Input Pin name	User defined	Specify the name of the AXI input clock pin.
Invert AXI Clock Input	On or off	Turn on to invert the AXI clock.
Shared Read/Write Address Channel tab Write Response Channel tab Read Data Channel tab Write Data Channel tab	User defined	This tab defines the AXI signal names. Efinix recommends that you use the default names. The signals are prefixed with the instance name you specified in the Base tab.

Supported DDR DRAM Modules

Trion® FPGAs support all JEDEC-compliant DDR DRAM modules (DDR3, DDR3L, LPDDR3, LPDDR2).

The following table lists modules that Efinix® has verified with Trion® FPGAs.

Table 30: DDR DRAM Modules Verified with Trion

Memory Type	Manufacturer	Part Number
LPDDR2	Micron	EDB1316BD
	Winbond	W97AH6KBVX2I
	Alliance Memory	AS4C128M16MD2A-25BIN
LPDDR3	Micron	MT52L256M32D1PF
	Winbond	W63CH6MBVACE
DDR3L	Micron	MT41K512M16
	SK Hynix	H5TC4G63CFR-RDA
	Samsung	K4B4G1646E
	Samsung	K4B2G1646F
	Nanya	NT5CC128M16JR

Revision History

Table 31: Revision History

Date	Version	Description
April 2022	1.7	Added note about not using LVDS RX as a primary clock resource to drive the PLL BR0. (DOC-768)
April 2022	1.6	Updated write operation example with atype waveform. (DOC-749)
March 2022	1.5	Added Narrow Transfer topic. (DOC-706)
October 2021	1.4	Updated AXI Logical Address to DDR Module Mapping table. (DOC556)
September 2020	1.3	Added supported DDR PHY frequency. Removed drive strength programmability from features list. Added BGA400 support DDR DRAM DQ width.
July 2020	1.2	Added support for Winbond W63CH6MBVACE DDR DRAM module.
May 2020	1.1	Removed all instances of DDR3U. Updated row address (first address) and column address (first address) mapping example for LPDDR3 Module (x32, 4 Gb).
March 2020	1.0	Initial release.