HMIC_S IP

User Guide

(UG022003, V1.6)

(31.08.2023)

Revisions History

Document Revisions

Version	Date	Revisions	Applicable IP and Corresponding Versions
V1.6	31.08.2023	Initial release.	V1.6

IP Revisions

IP Version	Date	Revisions
V1.6	31.08.2023	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
APB	Advanced Peripheral Bus
AXI	AdvancedeXtensible Interface
DDR	Double Data Rate
DFI	DDR PHY Interface
LP	Low Power
MC	Memory Controller
MR	Mode Register
MRS	Mode Register Set
PHY	Physical
RD	Read
UI	User Interface
WR	Write
DCD	DDR Command Decode
DCP	DDR3 Command Procedure
HMIC	High performance Memory Interface Controller
IPC	IP Compiler
PDS	Pango Design Suite
UCE	User Constraint Editor

Related Documentation

The following documentation is related to this manual:

1. ***Pango_Design_Suite_Quick_Start_Tutorial***
2. ***Pango_Design_Suite_User_Guide***
3. ***IP_Compiler_User_Guide***
4. ***Simulation_User_Guide***
5. ***User_Constraint_Editor_User_Guide***
6. ***Physical_Constraint_Editor_User_Guide***
7. ***Route_Constraint_Editor_User_Guide***
8. ***JESD79-3D, DDR3 SDRAM Standard***
9. ***DDR PHY Interface, DFI 3.1 Specification***

Table of Contents

Tables

Figures

Chapter 1 Preface

This chapter describes the scope, structure, and writing standards of this manual to help users quickly find the information they need.

1.1 Introduction of the Manual

This manual serves as the user guide for the DDR3 IP—HMIC_S (High performance Memory Interface Controller Soft core) IP launched by Pango Microsystems. The content of this manual primarily includes the IP user guide and related information. This manual helps users quickly understand the HMIC_S IP features and usage.

1.2 Writing Standards of the Manual

Table 1-1 Description of Writing Standards

Text	**Rules**
Attention	If users ignore the attention contents, they may suffer adverse consequences or fail to operate successfully due to incorrect actions.
Description	Instructions and tips provided for users.

Chapter 2 IP User Guide

This chapter provides a guide on the use of HMIC_S IP, including an introduction to IP, IP block diagram, IP generation process, Example Design, IP interface description, IP register description, typical applications, instructions and considerations, and IP debugging methods. More details on the design process can be found in the following PDS help documentation.

- ***Pango_Design_Suite_Quick_Start_Tutorial***
- ***Pango_Design_Suite_User_Guide***
- ***IP_Compiler_User_Guide***
- ***Simulation_User_Guide***

2.1 IP Introduction

HMIC_S IP is a DDR3 IP launched by Pango Microsystems, which can be configured and generated using the IPC (IP Compiler) tool in the company's PDS (Pango Design Suite).

2.1.1 Key Features

The main features of the HMIC_S IP product are as follows.

- DDR3;
- A maximum bit width of 32 bits;
- User interface: reduced AXI4 bus interface and APB bus interface;
- Configurable low-power modes: Self-Refresh and Power Down;
- The highest data rate of DDR3 up to 800Mbps;
- Burst Length 8 and single Rank.

2.1.2 Applicable Devices and Packages

Table 2-1 HMIC_S IP Applicable Devices and Packages

Applicable Devices	**Supported Packages**
PGL25G	FBG256/MBG324/FBG484
PGL50H	MBG324/MBG484/FBG484
PGL50G	MBG324/MBG484/FBG484
PGL100H	FBG900

2.2 IP Block Diagram

The system block diagram of HMIC_S IP is shown in Figure 2-1.

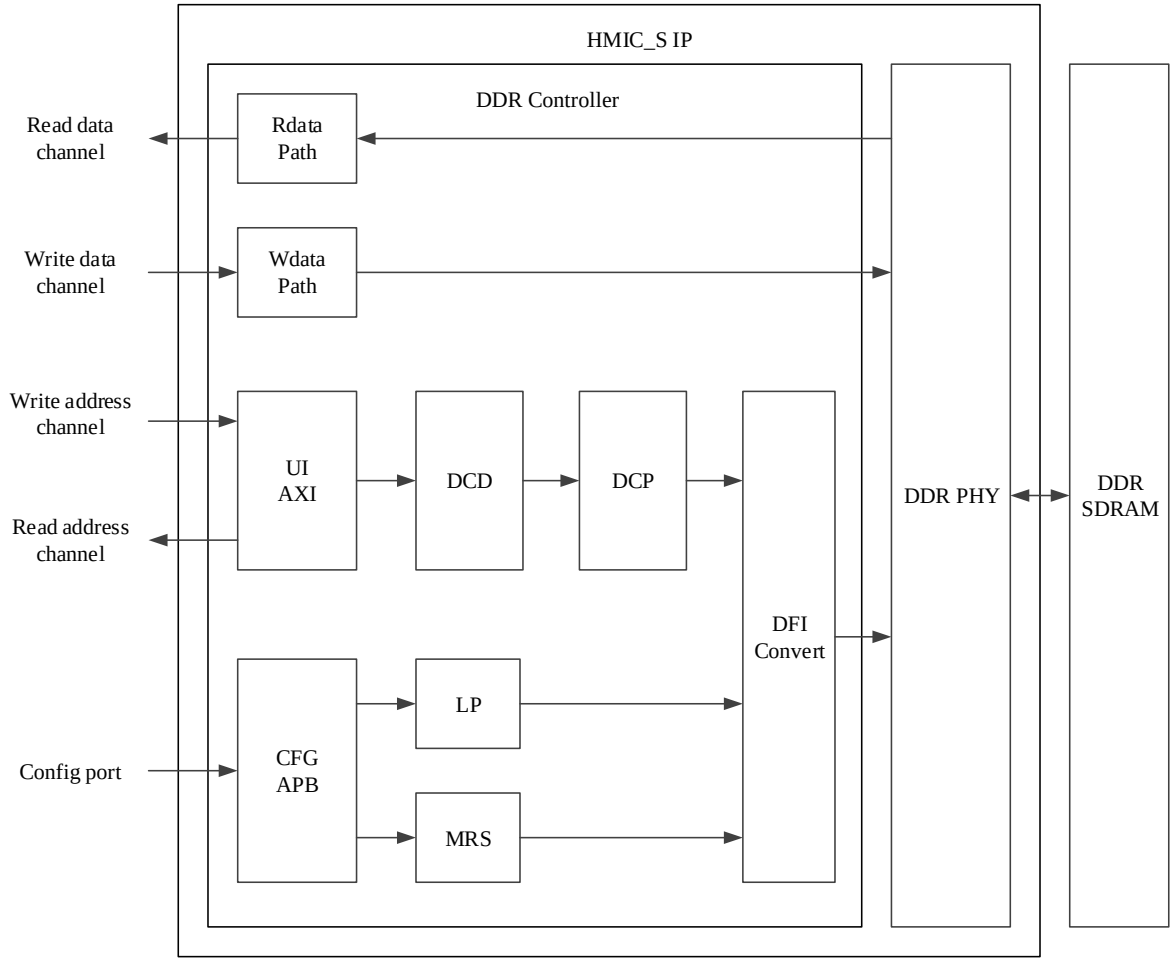


Figure 2-1 HMIC_S IP System Block Diagram

HMIC_S IP supports the operating mode of Controller + PHY. In this mode, the IP includes DDR Controller and DDR PHY functions. Users read and write data through the Simplified AXI4 interface and control low power function and MRS through the APB interface.

- Simplified AXI4 interface

  This interface comprises four parts: write address channel, read address channel, write data channel, and read data channel.

  Users initiate read and write operations through the write and read address channels; their commands are parsed into internal controller commands in the UI AXI module; decomposed into corresponding DDR commands in the DCD (DDR Command Decode) module; implement timing control for DDR in the DCP (DDR3 Command Procedure) module; converted into the DFI interface in the DFI Convert module, transferred to PHY, and finally to the DDR Memory interface.

Write data is passed through the write data channel interface, directly through the Wdata Path module to the DDR PHY, and ultimately transferred to the DDR Memory interface.

Read data from DDR Memory, after being sampled and parsed by the DDR PHY, is synchronized through the Rdata Path module and returned to the user through the read data channel interface.

- Config port

  This interface is an APB configuration interface through which users can read the status of DDR SDRAM and implement low power consumption and MRS control.

2.3 IP Generation Process

2.3.1 Module Instantiation

Customized configurations of HMIC_S IP can be completed through the IPC tool, instantiating the required IP modules. For detailed instructions on using the IPC tool, please refer to "***IP_Compiler_User_Guide***".

The main steps for instantiating the HMIC_S IP module are described below.

2.3.1.1 Selecting IP

After selecting the FPGA device type, the Catalog interface displays the loaded IP models. Select the corresponding version of DDR3 Interface under the "System/DDR/Soft" directory. The IP selection path is shown in Figure 2-2. Then set the Pathname and Instance Name on the right side of the page, the project instantiation interface is shown in Figure 2-3.

Recommendation:

It is recommended to use version 2021.4-SP3.6-ads or above.

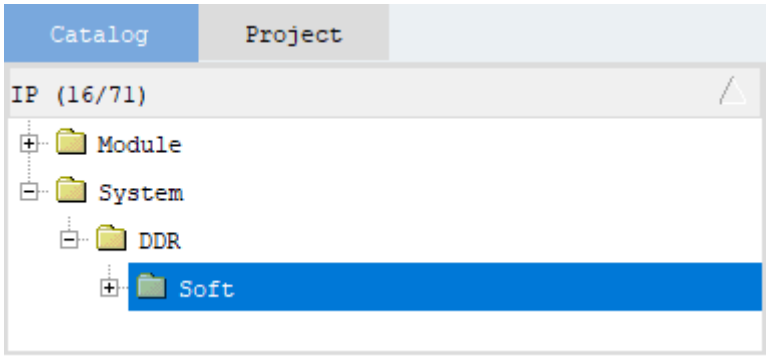


Figure 2-2 Selecting the HMIC_S IP interface

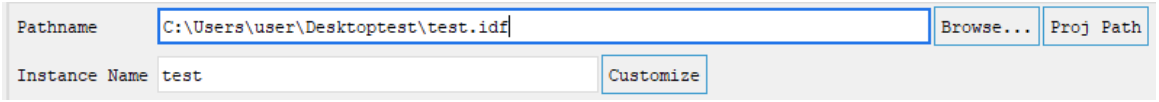


Figure 2-3 Project Instantiation Interface

2.3.1.2 IP Parameter Configuration

After selecting the IP, click <Customize> to enter the HMIC_S IP parameter configuration interface. The left Symbol is the interface block diagram, as shown in Figure 2-4; the Parameter Configuration window is shown on the right side, as shown in Figure 2-5.

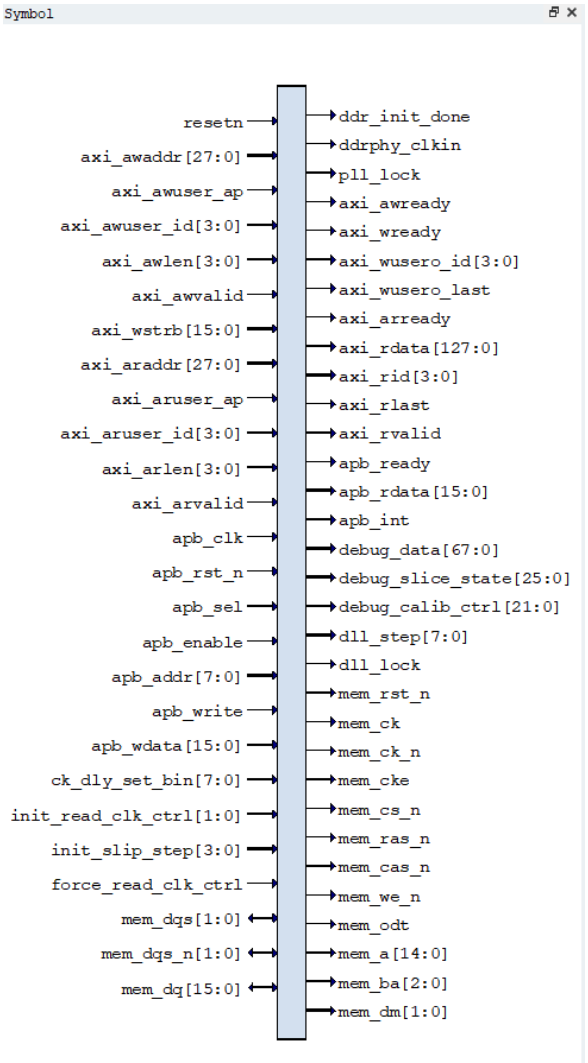


Figure 2-4 HMIC_S IP Interface Block Diagram

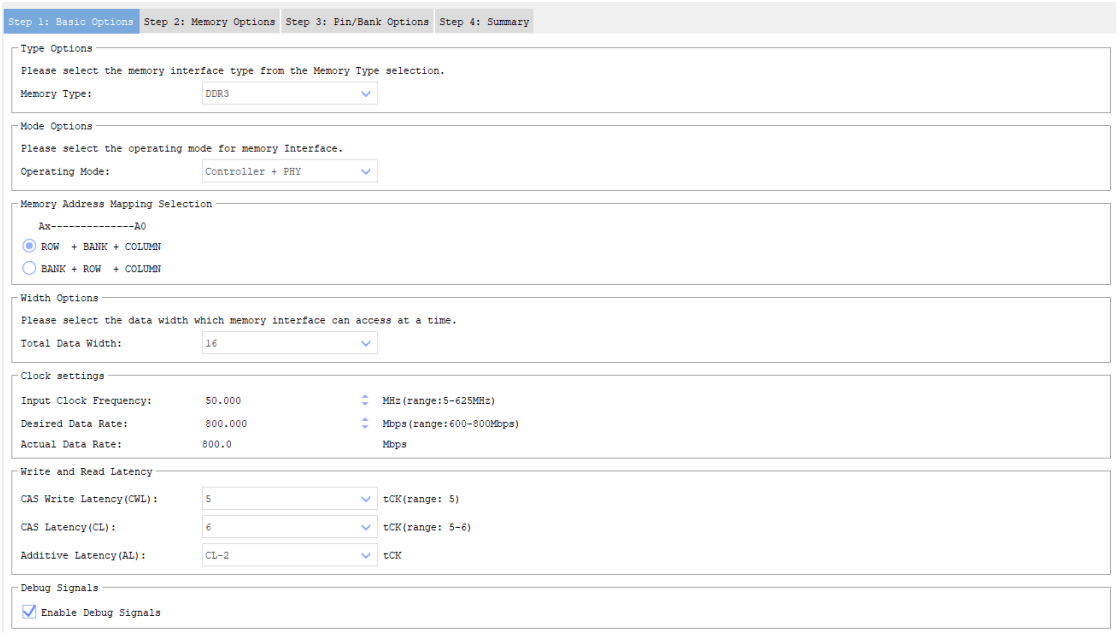


Figure 2-5 HMIC_S IP Parameter Configuration Interface

Attention:

Please be sure to configure the IP parameters in the order of the pages, following Step 1 → Step 2 → Step 3 → Step 4.

Parameter configuration includes four pages, namely Step1: Basic Options, Step2: Memory Options, Step3: Pin/Bank Options, Step4: Summary. The configuration steps for HMIC_S IP are described as follows.

2.3.1.2.1 Step 1: Basic Options

Basic Options is the fundamental configuration page for the IP, as shown in Figure 2-6. Refer to Table 2-2 for detailed parameters.

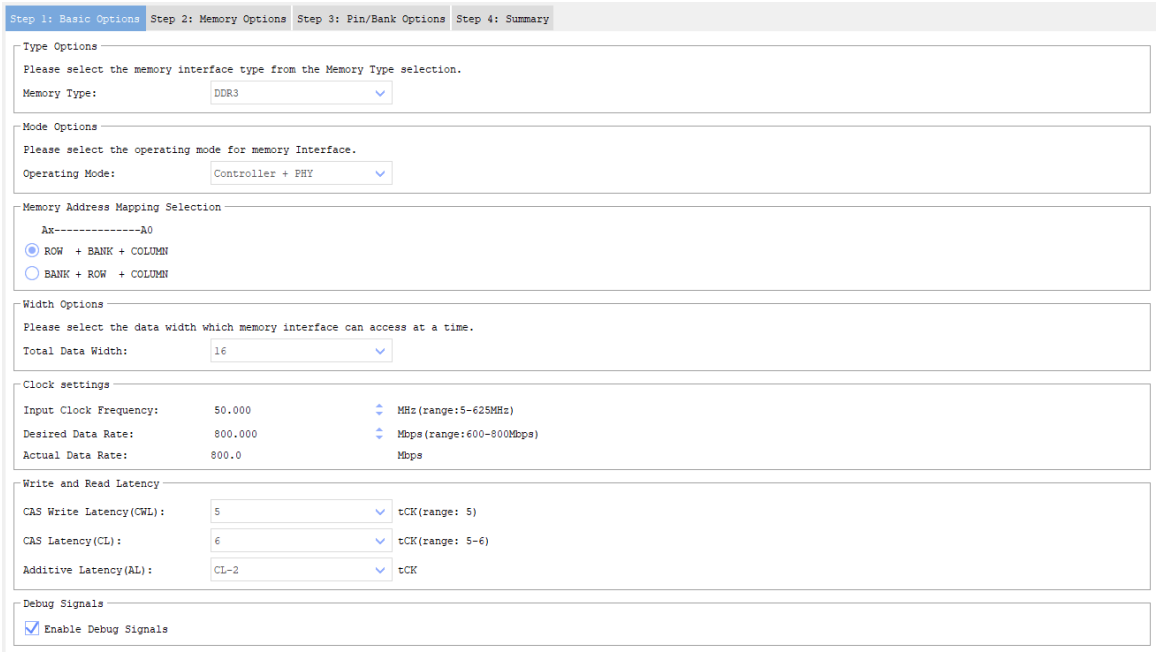


Figure 2-6 Basic Options Page

Table 2-2 Descriptions of Configuration Parameters on the Basic Options Page

Option Domain	**Parameter/Configuration Options**	**Parameter Description**	**Default Value**
Type Options	Memory Type	The type of SDRAM used, currently supported type is: DDR3.	DDR3
Mode Options	Operating Mode	HMIC_S operating mode selection. The supported operating mode is: Controller + PHY; The generated IP code includes both the Controller and PHY.	Controller + PHY
Memory Address Mapping Selection	ROW + BANK + COLUMN	The read and write address mapping method for the Controller AXI interface is selected as: "ROW + BANK + COLUMN". Refer to Figure 2-16.	Selected
	BANK + ROW + COLUMN	The read and write address mapping method for the Controller AXI interface is selected as: "BANK + ROW + COLUMN". Refer to Figure 2-17.	Cleared
Width Options	Total Data Width	The total DQ width of the off-chip SDRAM connected to HMIC_S. Supported total widths include: 1) 32 2) 24 3) 16 4) 8	16

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
Clock settings	Input Clock Frequency	The input clock of HMIC_S, in MHz.	50
	Desired Data Rate	The desired data rate. DDR3 rate up to 800Mbps.	800
	Actual Data Rate	The actual data rate, which should be as close to the desired rate as possible.	-
Write and Read Latency	CAS Write Latency(CWL)	CAS Write Latency configuration, in tCK.	5
	CAS Latency(CL)	CAS Latency configuration, in tCK.	6
	Additive Latency(AL)	Additive Latency configuration, in tCK.	CL-2
Debug Signals	Enable Debug Signal	Debug signal enable switch.	Selected

Note: "-" indicates that there is no default value for this parameter in the IP configuration interface.

2.3.1.2.2 Step 2: Memory Options

Memory Options is the configuration page for Memory parameters, as shown in Figure 2-7. Refer to Table 2-3 for detailed parameters.

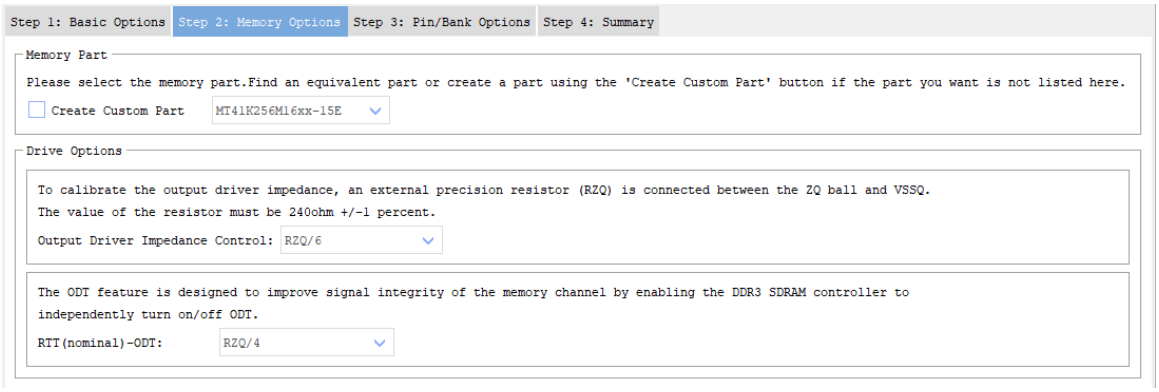


Figure 2-7 Memory Options Page

Table 2-3 Descriptions of Configuration Parameters on the Memory Options Page

Option Domain	Parameter/ Configuration Options	Parameter Description	Default Value
Memory Part	SDRAM device model	The models supported by DDR3 include: MT41J128M8xx-15E MT41J128M16xx-15E MT41J64M16xx-15E MT41J256M8xx-15E MT41K256M16xx-15E	MT41K256M16xx-15E
	Create Custom Part	Custom new SDRAM type enable selection. If the device models supported by the IP do not meet the requirements, users can check this option and customize a new SDRAM type based on the reference model in the Custom Memory Part option box below.	Cleared

Option Domain	Parameter/ Configuration Options	Parameter Description	Default Value
Custom Memory Part[1]	Select Base Part	Custom SDRAM reference model.	MT41K256M16xx-15E
	Timing Parameters	Custom SDRAM Timing parameters, including: trfc, tras, trp, trcd, twr, trefi, trtp, twtr.	trfc: 300 tras: 37.5 trp: 13.5 trcd: 13.5 twr: 15 trefi: 7.8 trtp: 7.5 twtr: 7.5
	Row Address	Row address.	15
	Column Address	Column address.	10
	Bank Address	Bank address.	3
Drive Options	Output Driver Impedance Control	Drive strength options supported by DDR3. For detailed information, please refer to: JESD79-3D, DDR3 SDRAM Standard.	RZQ/6
	RTT(nominal)-ODT		RZQ/4

2.3.1.2.3 Step 3: Pin/Bank Options

Pin/Bank Options is the configuration page for interface parameters, as shown in Figure 2-8. Refer to Table 2-4 for detailed parameters.

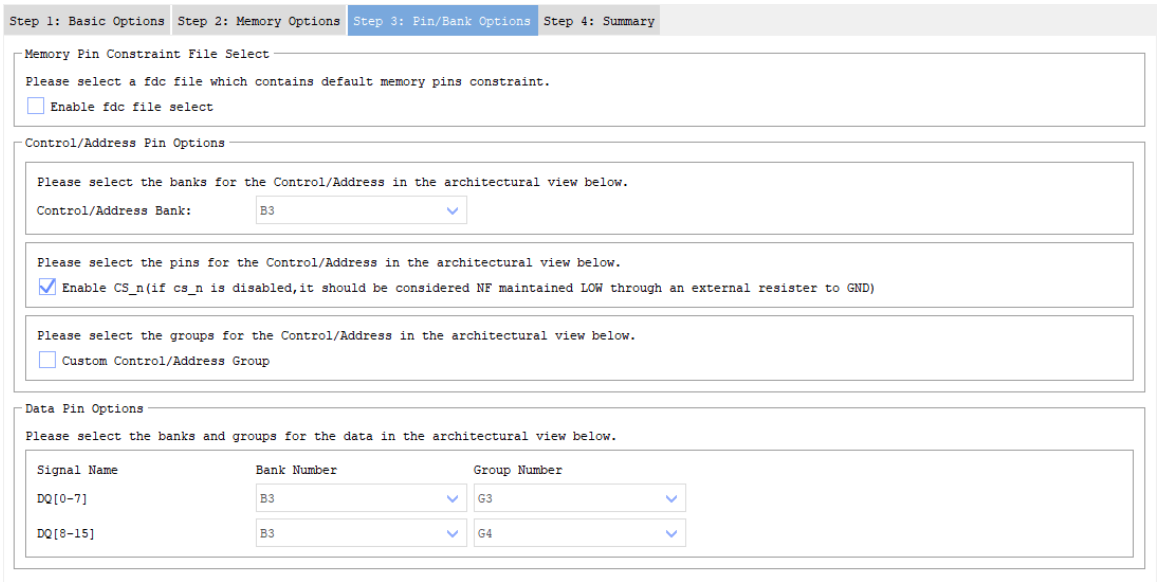


Figure 2-8 Pin/Bank Options Page

1 This option box is displayed when the Create Custom Part option is selected, and it is used to customize a new SDRAM type.

Attention:

All configuration items in "Step 3: Pin/Bank Options" must be configured according to the actual pin assignments on the single board. After generating the IP, constrain the DQ, reset, and status signal pin locations according to the actual pin assignments on the single board, and otherwise the Flow may run incorrectly.

Table 2-4 Description of Configuration Parameters on Pin/Bank Options Page

Option Domain	**Parameter/ Configuration Options**	**Parameter Description**	**Default Value**
Memory Pin Constraint File Select	Enable fdc file select	Custom .fdc files can be imported[2]. Enable: Users can input the path to their fdc file in the text box, or select the fdc file via the dialogue box, to automatically read the Memory interface constraints from the fdc file, and configure the Control/Address and Data Pin in the UI.	Cleared
Control/ Address Pin Options	Control/Address Bank	The Bank where the Memory interface's control and address lines are located.	B3
	Enable CS_n	Mem_cs_n signal enable selection	Selected
	Custom Control/ Address Group	User-defined control and address bus grouping enable selection. Selected: User defines the grouping and pin constraints for each PAD; Cleared: default grouping.	Cleared
	Control and address signals	This option box is displayed when the "Custom Control/Address Group" option is selected, and it is used to select the Group and Pin where the control and address signals corresponding PADs are located.	-
Data Pin Options	Bank Number	Select the Bank where DQ[3] is located.	B3
	Group Number	Select the Group where DQ[3] is located.	G4

Note: "-" indicates that there is no default value for this parameter in the IP configuration interface.

Attention:

1. For the "Custom Control/Address Group" option, it is recommended to use custom grouping due to different PCB traces in actual use.
2. When checking the "Custom Control/Address Group" option for custom pin configuration, ensure that signals are not constrained to the same pin. If pins are constrained to the same location, the UI will highlight the corresponding Pin Number in red.

[2] The signal names of the Memory interface in the user's fdc must match those in the Example Design.

3 DQ[8-15] is displayed when the bit width is greater than 8, DQ[16-23] when it is greater than 16, and DQ[24-31] when it is greater than 24.

2.3.1.2.4 Step 4: Summary

The Summary page is used to print the current configuration information without configuration parameters required, as shown in Figure 2-9.

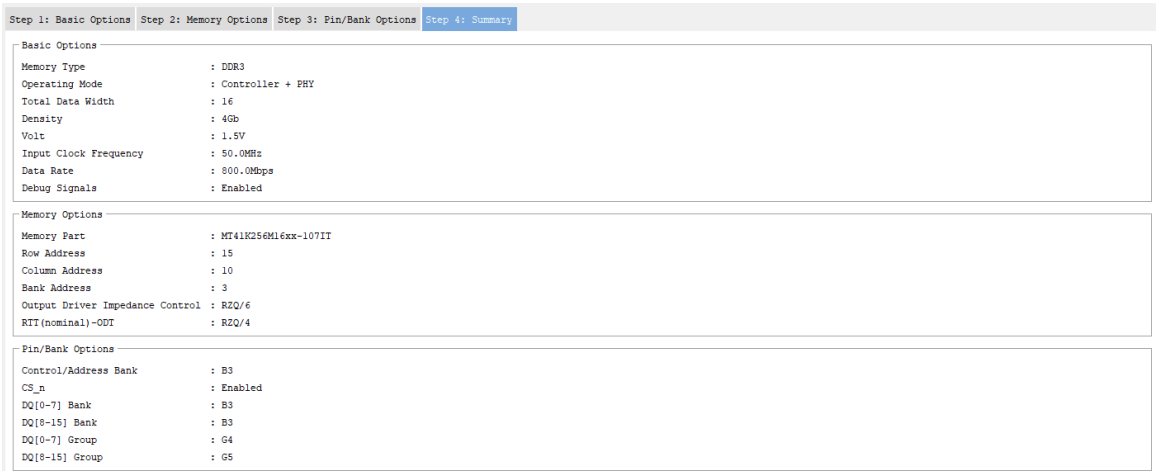


Figure 2-9 Summary Page

2.3.1.3 Generating IP

Upon completion of parameter configuration, click the <Generate> button in the top left corner to generate the HMIC_S IP code according to the user's specific settings. The information report interface for IP generation is shown in Figure 2-10.

Figure 2-10 HMIC_S IP Generation Report Interface

Attention:

The .pds and .fdc files generated by the IP are for reference only, please change the pin constraints according to the actual pin connections when using.

Upon successful IP generation, the files indicated in Table 2-5 will be output to the Project path specified in Figure 2-3.

Table 2-5 Output files generated by the IP

Output File[4]	**Description**
$instname.v	The top-level .v file of the generated IP for Controller + PHY.
$instname_ddrphy_top.v	The top-level .v file of the generated IP for PHY.
$instname.idf	The Configuration file of the generated IP.
/rtl/*	The RTL code file of the generated IP.
/example_design/*	The Test Bench and the corresponding Memory Simulation Model files used by the generated IP's Example Design.
/pnr/*	The project files .pds and pin constraint files .fdc for the generated IP's Example Design.
/sim/*	The simulation directory for the generated IP. sim.tcl is a Modelsim simulation script, makefile is a VCS simulation script, and sim_file_list.f is a list of simulation files.
/sim_lib/*	The directory of the encryption files for the IP.
/rev_1	The default output path for synthesis reports. (this folder is generated only after specifying the synthesis tool)
readme.txt	The readme file describes the structure of the generation directory after the IP is generated.

2.3.2 Constraint Configuration

For specific configuration methods of constraint files, please refer to the help documents under the PDS installation path: "***User_Constraint_Editor_User_Guide***", "***Physical_Constraint_Editor_User_Guide***" and "***Route_Constraint_Editor_User_Guide***".

2.3.3 Simulation Runs

The simulation of HMIC_S IP is based on the Test Bench of the Example Design. For detailed information about Example Design, please refer to "2.4 Example Design".

For more detailed information about the PDS simulation functions and third-party simulation tools, please consult the related help documents in the PDS installation path: "***Pango_Design_Suite_User_Guide***", "***Simulation_User_Guide***".

2.3.3.1 ModelSim Simulation

Simulation operation steps: Open cmd.exe, switch the current directory to "/sim" under the IP generation directory in the command line, run vsim, and open the ModelSim simulation software, as shown in Figure 2-11.

4 <$instname> is the instantiation name entered by the user; "*" is a wildcard used to replace files of the same type.

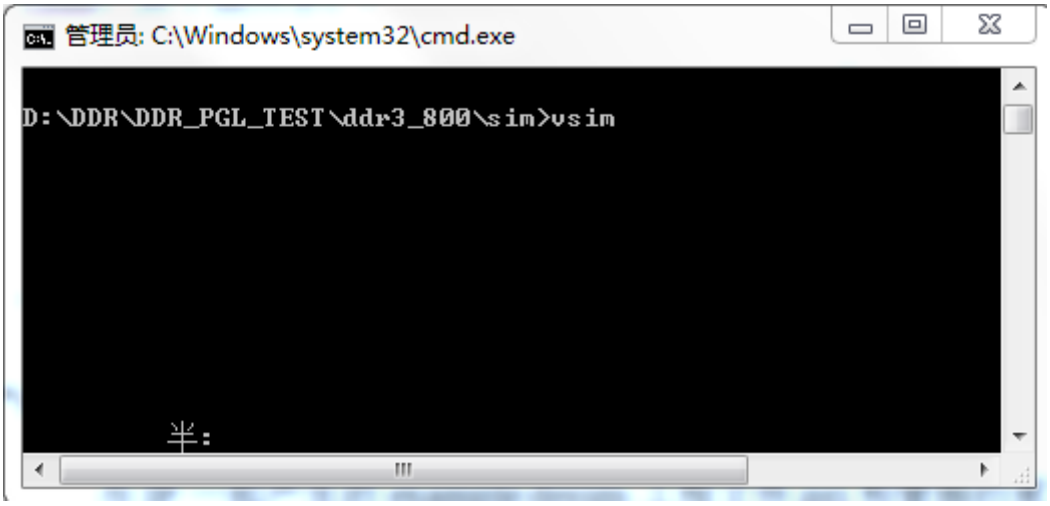


Figure 2-11 Open ModelSim Instruction

Run the sim.tcl script in the ModelSim simulation software to perform simulation, as shown in Figure 2-12.

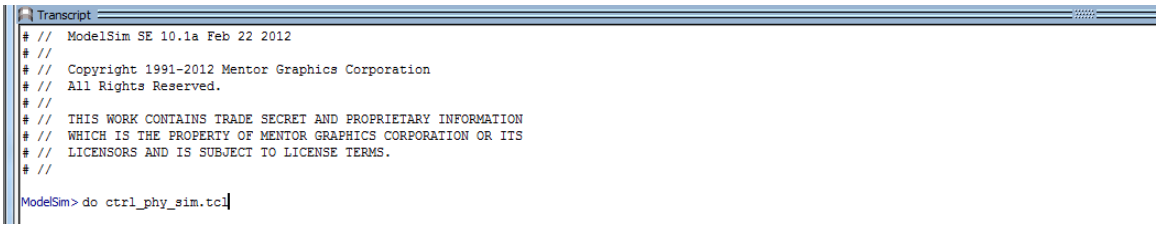


Figure 2-12 ModelSim Simulation Execution Script

2.3.3.2 VCS simulation

Enter the "sim/vcs" directory in Terminal, type "make" and press "Enter" to start the simulation, as shown in Figure 2-13.

Figure 2-13 VCS Simulation Script Execution

2.3.4 Synthesis and Placement/Routing

The specific usage of the PDS synthesis tool and placement/routing tool can be found in the help documents in the PDS installation path.

Attention:

The Example Design project files .pds and pin constraint files .fdc generated with the IP are stored in the "/pnr/example_design" directory. Modify the physical constraints according to the devices and PCB traces used, refer to "2.8 Descriptions and Considerations" for details.

2.3.5 Resource Utilization

Table 2-6 Typical Resource Utilisation Values for HMIC_S IP Based on Applicable Devices

Device	**Configuration Mode**		**Typical Resource Utilisation Values**							
			Controller				**PHY**			
			LUT	**FF**	**PLL**	**USCM**	**LUT**	**FF**	**PLL**	**USCM**
PGL25G	DDR3 x16	Controller +PHY	1388	1311	0	0	2365	1909	0	0
PGL50G	DDR3 x16		1386	1311	0	0	2356	1909	0	0
	DDR3 x32		1394	1311	0	0	3686	3089	0	0
PGL50H	DDR3 x16		1387	1311	0	0	2361	1909	0	0
	DDR3 x32		1387	1311	0	0	3692	3089	0	0
PGL100H	DDR3 x16		1388	1311	0	0	2366	1909	0	0
	DDR3 x32		1387	1311	0	0	3691	3089	0	0

2.4 Example Design

This section mainly introduces the Example Design scheme based on HMIC_S IP (Controller + PHY mode). In this scheme, the user logic acts as AXI Master with the HMIC_S IP as AXI Slave. The user logic writes data through the Write channel of the AXI interface and receives data through the Read channel for data comparison. If there is a data error, the Error LED will light up.

2.4.1 Design Block Diagram

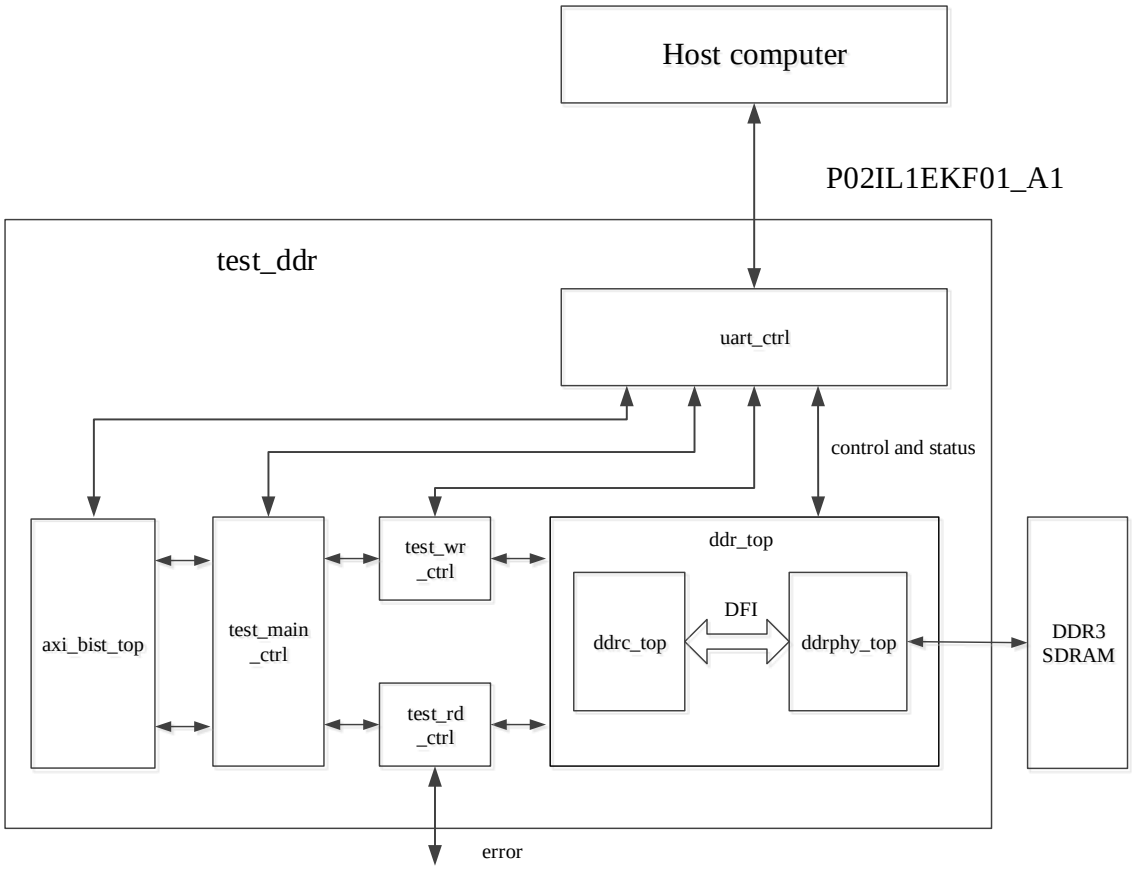


Figure 2-14 Example Design System Block Diagram

The system block diagram of the Example Design is as shown in Figure 2-14, where axi_bist_top is the top-level module for AXI read and write instructions, test_main_ctrl is the control module for AXI read and write instructions, test_wr_ctrl is the control module for AXI write instructions and write data, test_rd_ctrl is the control module for AXI read instructions and read data, and uart_ctrl is the Serial Port conversion module for easy control and internal state reading during debugging.

2.4.2 Test Method

In the Example Design, the user logic performs read and write operations on HMIC_S IP and checks the readback data. The detailed test process is as shown in Figure 2-15.

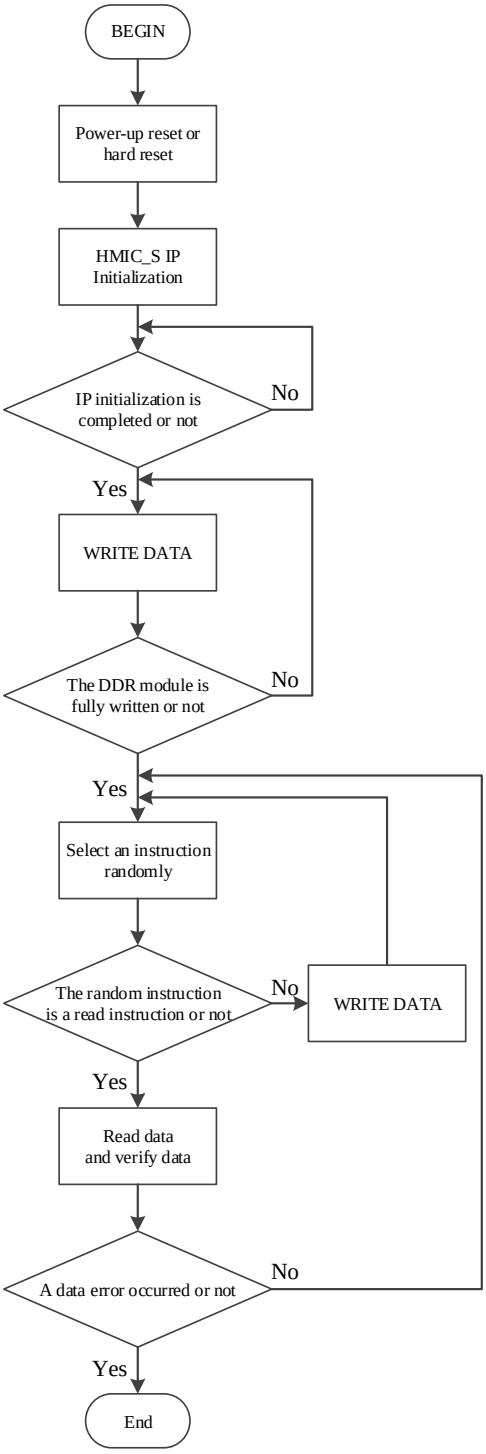


Figure 2-15 Example Design Test Process Diagram

After the system is powered up or a hard reset is initiated, the HMIC_S IP begins initialization. Once initialization is completed (indicated by the ddrc_init_done signal going high), the test_main_ctrl module controls the test_wr_ctrl module to generate write instructions and write data to initialize the DDR chips. After write full, test_main_ctrl starts random read and write operations, and test_rd_ctrl checks the readback data to determine if there are any errors.

Attention:

Do not directly use the Example Design generated by the IP for Flow on-board testing. Constrain pins according to the actual pin connections of the single board, then proceed with Flow on-board testing.

2.5 Descriptions of IP Interfaces

This section provides the HMIC_S IP related interface instructions and timing descriptions.

2.5.1 Controller Interface Description

2.5.1.1 Global Interface

Table 2-7 Global Interface

Port	I/O	Bit width	Valid Values	Description
clk	I	1	-	External clock input.
rst_n	I	1	Low	External reset input.
phy_init_done	I	1	High	Initialization completion flag for ddrphy: 1’b1: ddrphy initialization is complete; 1’b0: ddrphy initialization is incomplete;
ddr_init_done	O	1	High	Initialization completion flag for the IP: 1’b1: ddr IP initialization is complete; 1’b0: ddr IP initialization is incomplete, and external operations on the ddr IP are invalid.

Note: "-" indicates that the parameter does not exist.

2.5.1.2 Simplified AXI4 interface

2.5.1.2.1 Write address channel

Table 2-8 Write Address Channel

Port	I/O	Bit width	Valid Values	Description
axi_awaddr	I	CTRL_ADDR_WIDTH	-	AXI write address.
axi_awuser_ap	I	1	High	AXI write with auto precharge.
axi_awuser_id	I	4	-	AXI write address ID.
axi_awlen	I	4	-	AXI burst length for writing.
axi_awready	O	1	High	AXI write address ready.
axi_awvalid	I	1	High	AXI write address valid.

Note: "-" indicates that the parameter does not exist.

2.5.1.2.2 Read address channel

Table 2-9 Read Address Channel

Port	I/O	Bit width	Valid Values	Description
axi_araddr	I	CTRL_ADDR_WIDTH	-	AXI read address.
axi_aruser_ap	I	1	High	AXI read with auto precharge.
axi_aruser_id	I	4	-	AXI read address ID.
axi_arlen	I	4	-	AXI burst length for reading.
axi_arready	O	1	High	AXI read address ready.
axi_arvalid	I	1	High	AXI read address valid

Note: "-" indicates that the parameter does not exist.

2.5.1.2.3 Write data channel

Table 2-10 Write Data Channel

Port	I/O	Bit width	Valid Values	Description
axi_wdata	I	DQ_WIDTH*8	-	AXI write data.
axi_wstrb	I	DQ_WIDTH*8/8	High	AXI write data strobes.
axi_wready	O	1	High	AXI write data ready.
axi_wusero_id	O	4	-	AXI write data ID.
axi_wusero_last	O	1	High	AXI write data last.

Note: "-" indicates that the parameter does not exist.

2.5.1.2.4 Read data channel

Table 2-11 Read Data Channel

Port	I/O	Bit width	Valid Values	Description
axi_rid	O	4	-	AXI read data ID.
axi_rlast	O	1	High	AXI read data last signal.
axi_rvalid	O	1	High	AXI read data valid.
axi_rdata	O	DQ_WIDTH*8	-	AXI read data.

Note: "-" indicates that the parameter does not exist.

2.5.1.3 Config port

Table 2-12 Config Interface

Port	I/O	Bit width	Valid Values	Description
apb_clk	I	1	High	APB clock.
apb_rst_n	I	1	Low	APB reset.
apb_sel	I	1	High	APB Select.
apb_enable	I	1	High	APB port enable.
apb_addr	I	8	-	APB address bus.
apb_write	I	1	High	APB read/write direction, high level for write, active low for read.
apb_ready	O	1	High	APB port ready.
apb_wdata	O	16	-	APB write data.
apb_rdata	O	16	-	APB read data.

Note: "-" indicates that the parameter does not exist.

2.5.1.4 DFI interface

Table 2-13 DFI Interface

Port	I/O	Bit width	Valid Values	Description
dfi_address	O	4*ROW_ADDR_WIDTH	-	DFI address bus.
dfi_bank	O	4*BADDR_WIDTH	-	DFI bank address bus.
dfi_reset_n	O	4	-	DFI chip reset.
dfi_cs_n	O	4	-	DFI chip select.
dfi_ras_n	O	4	-	DFI row address strobe bus.
dfi_cas_n	O	4	-	DFI column address strobe.
dfi_we_n	O	4	-	DFI write enable signal.
dfi_cke	O	4	-	DFI clock enable.
dfi_odt	O	4	-	DFI on-die termination control bus.
dfi_rddata_valid	I	1	High	Read data valid on DFI interface.
dfi_rddata	I	DQ_WIDTH*8	-	DFI interface data.
dfi_wrdata_en	O	4	High	DFI write enable signal.
dfi_wrdata	O	DQ_WIDTH*8	-	DFI write data.
dfi_wrdata_mask	O	DQ_WIDTH*8/8	High	DFI Write data byte mask.
dfi_init_complete	I	1	High	PHY has completed training operations and is now in a normal state.
dfi_error	I	1	High	PHY training error indication.
dfi_phyupd_req	I	1	-	PHY requests an update.
dfi_phyupd_ack	O	1	-	Feedback signal of dfi_phy_req, allowing PHY to update.

Note: "-" indicates that the parameter does not exist.

2.5.2 Controller Interface Timing Description

The Simplified AXI4 interface uses a trimmed version of the AXI4 protocol. The Config interface uses the APB protocol.

2.5.2.1 Simplified AXI4 interface

For differences between the Simplified AXI4 interface as defined in this design and the standard AXI4 protocol, refer to Table 2-14 and Table 2-15.

Table 2-14 Differences between Simplified AXI4 and Standard AXI4

Channel	**Differences**
Write address channel	Preserve AWID, AWADDR, AWLEN, AWUSER, AWVALID, and AWREADY ports; remove AWSIZE, AWBURST, AWLOCK, AWCACHE, AWPROT, AWQOS, and AWREGION ports.
Write data channel	Preserve WID, WDATA, WSTRB, WLAST, WREADY ports; remove WUSER and WVALID ports.
Write responsechannel	NA.
Read address channel	Preserve ARID, ARADDR, ARLEN, ARUSER, ARVALID, and ARREADY ports; remove ARSIZE, ARBURST, ARLOCK, ARCACHE, ARPROT, ARQOS, and ARREGION ports.
Read data channel	Preserve RID, RDATA, RLAST, and RVALID; remove RRESP and RUSER ports.
Low-power interface	NA.
Clock	The clock for AXI and MC is the same.

Table 2-15 Detailed Differences between Simplified AXI4 and Standard AXI4

Interface Signal Classification	**Standard AXI4**	**Simplified AXI4**	**Signal Source**	**Description**	**Compatibility Method**
Global signals	ACLK	clk	Clock Source	Global clock signal	PHY core division clock
	ARESETn	rst_n	Reset Source	Global reset signal (active low)	MC reset clock
Write address channel signals	AWID[5]	axi_awuser_id	Master	Write address ID	Assign a fixed value when not in use
	AWADDR[5]	axi_awaddr	Master	Write address	Refer to the footnote
	AWLEN[5]	axi_awlen	Master	Write burst length	Default burst size is 8
	AWSIZE[6]	NA	Master	Write burst size	Refer to the footnote
	AWBURST[6]	NA	Master	Burst type	Refer to the footnote

5 For interface timing, please refer to "2.5.2.1.1 Write address channel timing", "2.5.2.1.2 Read address channel timing", "2.5.2.1.3 Write data channel timing", and "2.5.2.1.4 Read data channel timing".

6 On the basis of the DDR3 IP instantiation module, a package module layer is provided, reserving input interfaces not connected to the controller as input signals.

Interface Signal Classification	Standard AXI4	Simplified AXI4	Signal Source	Description	Compatibility Method
	AWLOCK[6]	NA	Master	Lock type	Refer to the footnote
	AWCACHE[6]	NA	Master	Cache type	Refer to the footnote
	AWPROT[6]	NA	Master	Protection type	Refer to the footnote
	AWQOS[6]	NA	Master	Write QOS identifier	Refer to the footnote
	AWREGION[6]	NA	Master	Write domain identifier	Refer to the footnote
	AWUSER[5]	axi_awuser_ap	Master	User-defined, write with auto precharge	Refer to the footnote
	AWVALID[5]	axi_awvalid	Master	Write address valid	Refer to the footnote
	AWREADY[5]	axi_awready	Slave	Write address ready	Refer to the footnote
Write data channel signals	WID[5]	axi_wusero_id	Master	Write ID	Assign a fixed value when not in use
	WDATA[5]	axi_wdata	Master	WRITE DATA	Refer to the footnote
	WSTRB[5]	axi_wstrb	Master	Write strobe	Refer to the footnote
	WLAST[5]	axi_wusero_last	Master	Write last flag	Refer to the footnote
	WUSER[6]	NA	Master	User defined	Refer to the footnote
	WVALID[6]	NA	Master	Write valid	Refer to the footnote
	WREADY[5]	axi_wready	Slave	Write ready	Refer to the footnote
Write response channel signals	BID[7, 8]	NA	Slave	Response ID	Refer to the footnote
	BRESP[7, 8]	NA	Slave	Write response	Refer to the footnote
	BUSER[7, 8]	NA	Slave	User defined	Refer to the footnote
	BVALID[7, 8]	NA	Slave	Write response valid	Refer to the footnote
	BREADY[6, 8]	NA	Master	Write response ready	Refer to the footnote
Read address channel signals	ARID[5]	axi_aruser_id	Master	Read ID	Assign a fixed value when not in use
	ARADDR[5]	axi_araddr	Master	Read data address	Refer to the footnote
	ARLEN[5]	axi_arlen	Master	Burst length	Refer to the footnote
	ARSIZE[6]	NA	Master	Burst size	Refer to the footnote

7 On the basis of the DDR3 IP instantiation module, a package module layer is provided, reserving output interfaces that can be assigned a fixed value as output signals.

8 For MC, as there is no write response channel provided internally, it is recommended that the master device ignores the write response channel function.

Interface Signal Classification	Standard AXI4	Simplified AXI4	Signal Source	Description	Compatibility Method
	ARBURST[6]	NA	Master	Burst type	Refer to the footnote
	ARLOCK[6]	NA	Master	Lock type	Refer to the footnote
	ARCACHE[6]	NA	Master	Cache type	Refer to the footnote
	ARPROT[6]	NA	Master	Inclusion type	Refer to the footnote
	ARQOS[6]	NA	Master	Read address QOS identifier	Refer to the footnote
	ARREGION[6]	NA	Master	Read address domain identifier	Refer to the footnote
	ARUSER[5]	axi_aruser_ap	Master	User-defined, read with auto precharge	Refer to the footnote
	ARVALID[5]	axi_arvalid	Master	Read address valid	Refer to the footnote
	ARREADY[5]	axi_arready	Slave	Read address ready	Refer to the footnote
Read data channel signals	RID[5]	axi_rid	Slave	Read ID	Refer to the footnote
	RDATA[5]	axi_rdata	Slave	Read data	Refer to the footnote
	RRESP[7]	NA	Slave	Read response	Refer to the footnote
	RLAST[5]	axi_rlast	Slave	Read last data	Refer to the footnote
	RUSER[7]	NA	Slave	User defined	Refer to the footnote
	RVALID[5]	axi_rvalid	Master	Read valid	Refer to the footnote
	RREADY[6]	NA	Master	Read ready	Refer to the footnote
Low-power interface signals	CSYSREQ[6]	NA	Clock controller	System low power request	Refer to the footnote
	CSYSACK[6]	NA	Peripheral devices	Low power request response	Refer to the footnote
	CACTIVE[6]	NA	Peripheral devices	Clock activity	Refer to the footnote

The Simplified AXI4 interface comprises four channels: Write Address, Read Address, Write Data, and Read Data, each operating independently. Generally, the address must be sent before transmitting and receiving data. Users can determine if the returned data matches the request data using different ID numbers.

There are two types of correspondence between Simplified AXI4 interface addresses and Memory addresses, corresponding to the "Memory Address Mapping Selection" option area in the configuration interface. When "ROW+BANK+COLUMN" is selected, the correspondence between Simplified AXI4 interface addresses and Memory addresses is as shown in Figure 2-16; when "BANK+ROW+COLUMN" is selected, the correspondence is as shown in Figure 2-17.

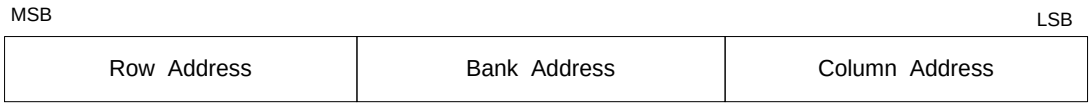


Figure 2-16 Memory Address 1 Mapped from Simplified AXI4 Interface Address

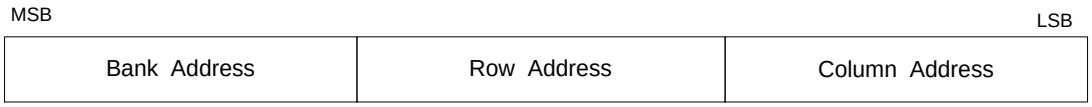


Figure 2-17 Memory Address 2 Mapped From Simplified AXI4 Interface Address

2.5.2.1.1 Write address channel timing

Signals included in the write address channel: axi_awready, axi_awvalid, axi_awaddr, axi_awuser_ap, axi_awuser_id, axi_awlen. A typical timing is shown in Figure 2-18.

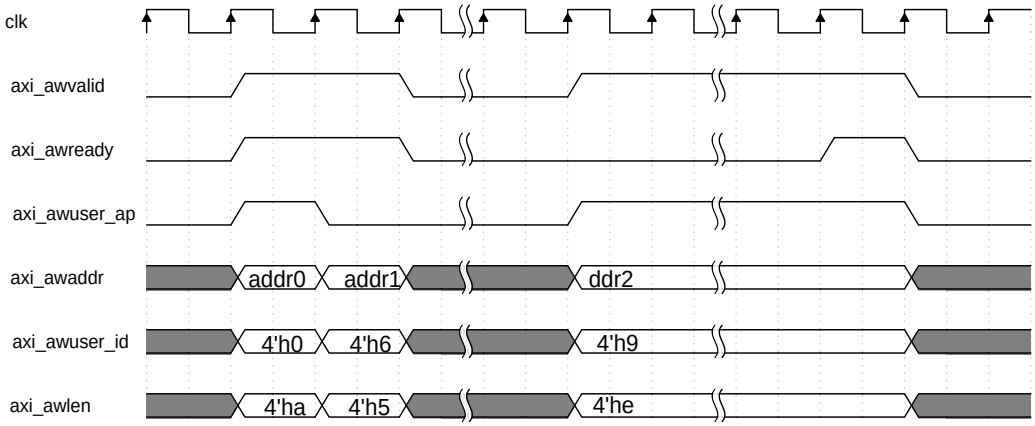


Figure 2-18 Typical Timing for Write Address

- The handshake is completed when axi_awready and axi_awvalid are both valid simultaneously.
- The burst length is controlled by axi_awlen, which is the value of axi_awlen plus one.
- Handshake process: starting from the rising edge of the clock when axi_awvalid is valid, axi_awaddr, axi_awuser_ap, axi_awuser_id, axi_awlen must remain unchanged until the handshake is completed and released.

2.5.2.1.2 Read address channel timing

Signals included in the read address channel: axi_arready, axi_arvalid, axi_araddr, axi_aruser_ap, axi_aruser_id, axi_arlen. A typical timing is shown in Figure 2-19.

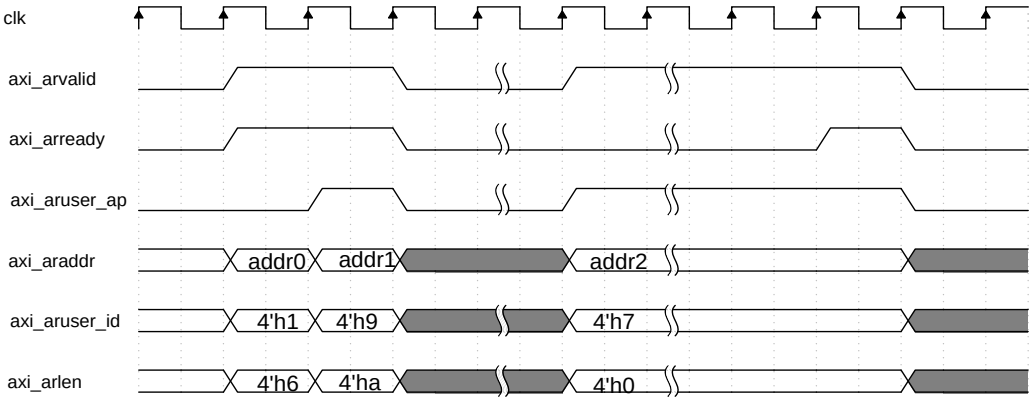


Figure 2-19 Typical Timing for Read Address

- The handshake is completed when axi_arready and axi_arvalid are both valid simultaneously.
- The burst length is controlled by axi_arlen, which is the value of axi_arlen plus one.
- Handshake process: starting from the rising edge of the clock when axi_arvalid is valid, axi_araddr, axi_aruser_ap, axi_aruser_id, axi_arlen must remain unchanged until the handshake is completed and released.

2.5.2.1.3 Write data channel timing

Signals included in the write data channel: axi_wready, axi_wusero_id, axi_wusero_last, axi_wdata, and axi_wstrb. A typical timing is shown in Figure 2-20.

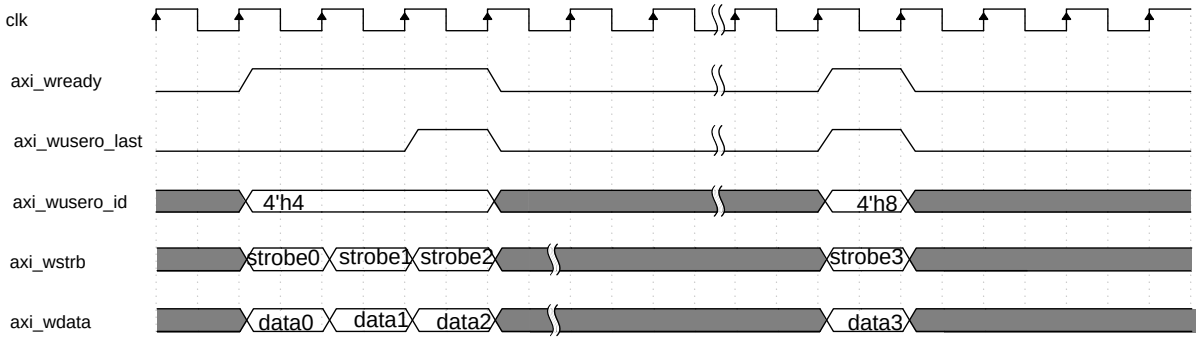


Figure 2-20 Typical Timing for Write Data

- The transmission is indicated as valid when axi_wready is valid.
- The end of a transmission is indicated when axi_wusero_last is valid.
- During transmission: axi_wready, axi_wusero_id, and axi_wusero_last are received synchronously, axi_wdata and axi_wstrb are transmitted synchronously.

2.5.2.1.4 Read data channel timing

Signals included in the read data channel: axi_rdata, axi_rid, axi_rlast, and axi_rvalid. A typical timing is shown in Figure 2-21.

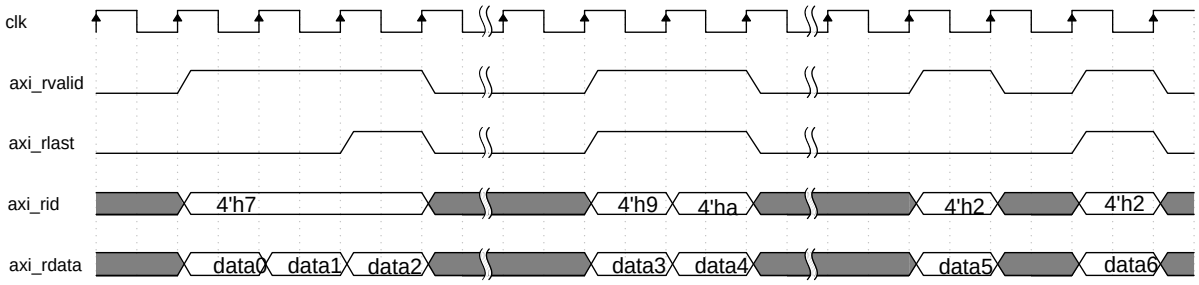


Figure 2-21 Typical Timing for Read Data

- The transmission is indicated as valid when axi_rvalid is valid.
- The end of a transmission is indicated when axi_rlast is valid.

2.5.2.2 Config port

The Config interface uses the standard APB protocol. By configuring the corresponding registers, DDR3 SDRAM can switch between Power Down, Self-Refresh, MRS, and Normal states. By reading the corresponding registers, the current state of the DDR3 can be queried.

The APB interface is for half-duplex operation communication, with independent read and write data lines and multi-function control and address lines. Each handshake consumes at least two cycles of apb_clk.

Signals included in the APB interface: apb_enable, apb_ready, apb_rdata, apb_sel, apb_write, apb_wdata, apb_clk.

The handshake is completed when apb_enable and apb_ready are both valid simultaneously.

2.5.2.2.1 APB interface write timing

The typical APB interface write timing is as shown in Figure 2-22.

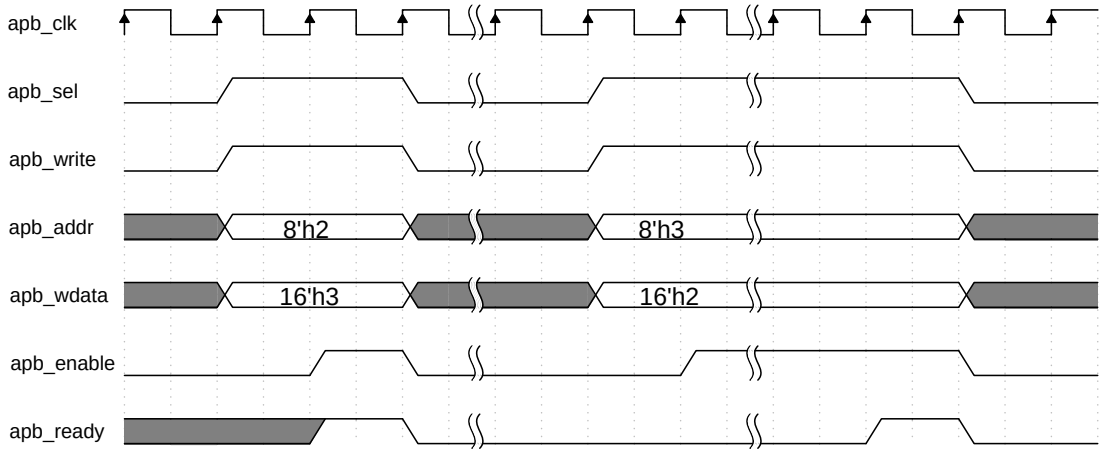


Figure 2-22 Typical Write Timing of the APB Interface

- The first clock cycle: apb_sel and apb_write are pulled high, apb_addr and apb_wdata are assigned initial values, which must remain stable until the handshake is completed and then released.
- The second clock cycle: apb_enable is pulled high until the handshake is completed and then released.

2.5.2.2.2 APB interface read timing

The typical APB interface read timing is as shown in Figure 2-23.

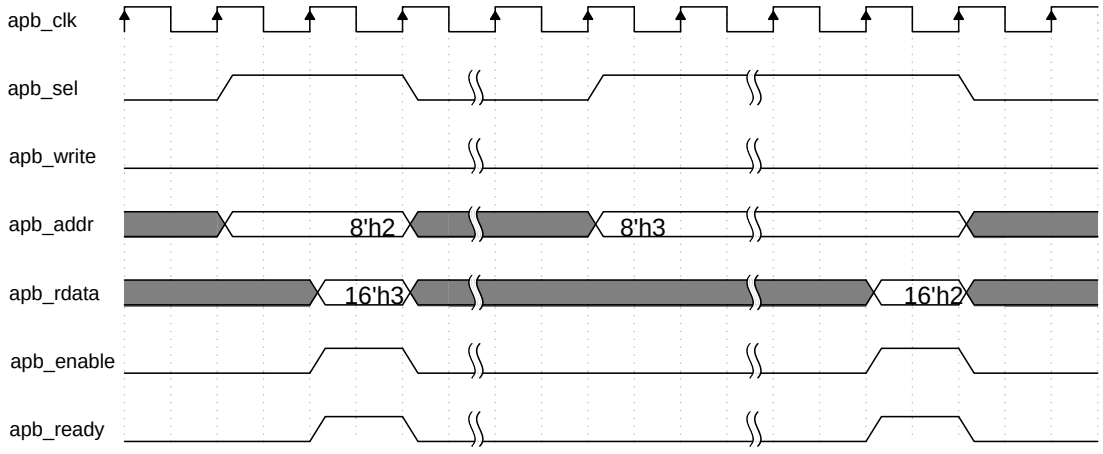


Figure 2-23 Typical Read Timing of the APB Interface

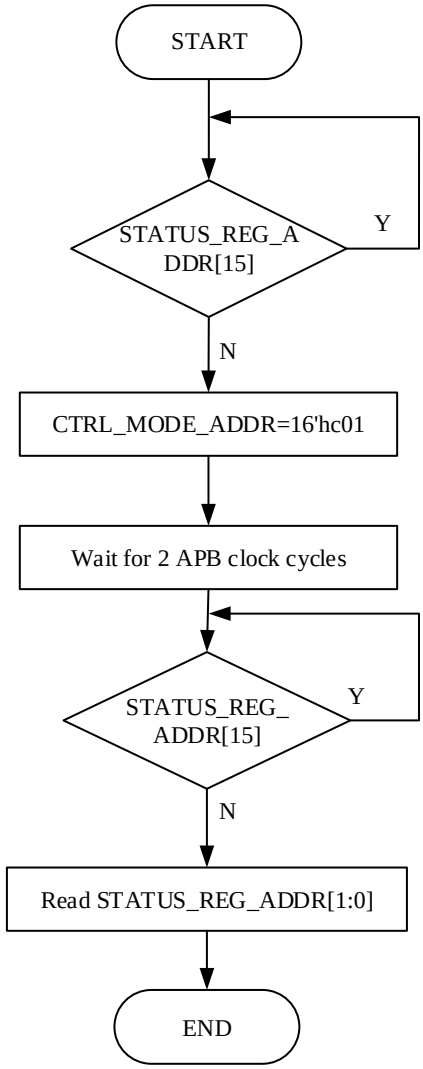
- The first clock cycle: apb_sel is pulled high, apb_write is pulled down, and apb_addr is set to an initial value, which must remain stable until the handshake is completed and then released.
- The second clock cycle: apb_enable is pulled high until the handshake is completed and then released.
- Valid data: apb_rdata is valid only during the handshake.

2.5.2.2.3 DDR3 status switching and querying method

To prevent users' status switching requests from interfering with normal DDR3 timings, certain operating rules must be followed when performing status switching (check the STATUS_REG_ADDR register to determine if the current DDR3 status and request have been responded to; control the CTRL_MODE_DATA register to achieve switching between DDR3 statuses).

- When sending a new status switch request, it is necessary to first check whether the current status switching request has been responded to, otherwise it will lead to errors;
- The status switching request and the status switching trigger enable must be sent synchronously, that is, bit 0 and bit[15:14] of CTRL_MODE_DATA must be configured at the same time.
- When configuring the Mode Registers within DDR3, the MODE_REG_0_ADDR, MODE_REG_1_ADDR, MODE_REG_2_ADDR, MODE_REG_3_ADDR registers must be configured first, followed by the CTRL_MODE_DATA register.
- After sending a status switching request, wait for two apb_clk cycles before checking the relevant status registers to refer to if a response has been received.

The state request switching process is as shown in Figure 2-24, with controlling DDR3 to enter Power Down status.



Step1: Query if the current state has been responded to

Step2: Send a new state request

Step3: Wait for 2 APB clock cycles

Step4: Query if the new state has been responded to

Step5: The new request has been responded to, read the new state

Figure 2-24 DDR3 Status Request Switching Flowchart

2.5.3 PHY Interface Description

2.5.3.1 Clock and reset interface

Table 2-16 Clock and Reset Interface

Port	I/O	Bit width	Valid Values	Description
ref_clk	I	1	-	External reference clock input.
ddr_rstn	I	1	-	External reset input.
pll_lock	O	1	High	PLL lock indication, high level indicates that it is locked.
ddrphy_clkin	I	1	-	PHY's system clock, output to the controller's working clock.
ddrphy_ioclk	I	9	-	Data bus IO clock.
ioclk_gate_clk	I	1	-	PLL gate signal sampling clock

Note: "-" indicates that the parameter does not exist.

2.5.3.2 DFI interface

Table 2-17 DFI Interface

Port	I/O	Bit width	Valid Values	Description
dfi_address	I	4*ROW_ADDR_WIDTH	-	DFI address bus.
dfi_bank	I	4*BADDR_WIDTH	-	DFI bank address bus.
dfi_reset_n	I	4	-	DFI chip reset.
dfi_cs_n	I	4	-	DFI chip select.
dfi_ras_n	I	4	-	DFI row address strobe bus.
dfi_cas_n	I	4	-	DFI column address strobe.
dfi_we_n	I	4	-	DFI write enable signal.
dfi_cke	I	4	-	DFI clock enable.
dfi_odt	I	4	-	DFI on-die termination control bus.
dfi_rddata_valid	O	1	High	Read data valid on DFI interface.
dfi_rddata	O	DQ_WIDTH*8	-	DFI interface data.
dfi_wrdata_en	I	4	High	DFI write enable signal.
dfi_wrdata	I	DQ_WIDTH*8	-	DFI write data.
dfi_wrdata_mask	I	DQ_WIDTH*8/8	High	DFI Write data byte mask.
dfi_init_complete	O	1	High	PHY has completed training operations and is now in a normal state.
dfi_error	O	1	High	PHY training error indication.
dfi_phyupd_req	O	1	-	PHY requests an update.
dfi_phyupd_ack	I	1	-	Feedback signal of dfi_phy_req, allowing PHY to update.

Note: "-" indicates that the parameter does not exist.

2.5.3.3 Memory interface

Attention:

The Memory interface in this IP is subject to the protocol. If the user selects a module that includes interfaces not contained in the protocol, please refer to the corresponding module Datasheet to add and manage the interfaces properly.

Table 2-18 Memory Interface

Port	I/O	Bit width	Valid Values	Description
mem_a	O	ROW_ADDR_WIDTH	-	DDR row and column address bus.
mem_ba	O	BANK_ADDR_WIDTH	-	DDR Bank address.
mem_ck	O	1	-	DDR input system clock.
mem_ck_n	O	1	-	DDR input system clock.
mem_cke	O	1	High	DDR input system clock valid.
mem_dm	O	DM_WIDTH	High	DDR input data Mask.
mem_odt	O	1	-	DDR ODT.
mem_cs_n	O	1	Low	DDR chip selection.
mem_ras_n	O	1	Low	Row address enable.
mem_cas_n	O	1	Low	Column address enable.
mem_we_n	O	1	Low	DDR write enable signal.
mem_reset_n	O	1	Low	DDR reset.
mem_dq	I/O	DQ_WIDTH	-	DDR data.
mem_dqs	I/O	DQS_WIDTH	-	DDR data strobe clock.
mem_dqs_n	I/O	DQS_WIDTH	-	DDR data strobe clock.

Note: "-" indicates that the parameter does not exist.

2.5.3.4 Debug interface

Table 2-19 Debug Interface

Port	I/O	Bit width	Description
debug_data	O	34*DQS_WIDTH	Debug data for each set of DDRPHY, 8bit DQ share a single DDRPHY.
debug_calib_ctrl	O	24	Debug data for Tainning status.
debug_slice_state	O	13*MEM_DQS_WIDTH	training status.
dll_step	O	8	DLL output delay step.
dll_lock	O	1	DLL output lock flag signal, active high.
force_read_clk_ctrl	I	1	dqs gate position-fixed enable, active high. 0: dqs gate position changes during the training process; 1: dqs gate position remains unchanged, always at the initial value.
init_slip_step	I	4*MEM_DQS_WIDTH	Initial value for dqs gate coarse adjustment position.
init_read_clk_ctrl	I	2*MEM_DQS_WIDTH	Initial value for DQS gate fine adjustment position.

Port	I/O	Bit width	Description
ck_dly_en	I	1	Command and address signal output delay adjustment enable for the memory interface, active low. 0: The command and address signal output delay remains unchanged, always at the setting value of init_ck_dly_step; 1: The command and address signal output delay is generated by the training process.
init_ck_dly_step	I	8	Command and address signal output delay step.
ddrphy_gate_update_en	I	1	Gate update enable signal.
update_com_val_err_flag	O	1	Drift code value jump anomaly indicator.
rd_fake_stop	I	1	Dummy read termination signal.

2.5.4 PHY Interface Timing Description

2.5.4.1 DFI interface specification

When users directly access the PHY Layer, they need to comply with the DFI-like interface specification defined in this document. Through this interface, the user can perform the following operations.

- Access (read/write) DDR SDRAM;
- Put DDR3 into Power Down or Self-Refresh status;
- Dynamically configure the values of the internal registers in DDR SDRAM;
- Read the status of the DDR SDRAM.

The DFI interface used in this design differs from the standard "***DFI 3.1 Specification***", and differences can be seen in Table 2-20 and Table 2-21.

Table 2-20 Differences between the DFI Interface of this Design and Standard DFI

Interface	Differences
ControlInterface	No requirements for tctrl_delay and tcmd_lat.
Write DataInterface	Except for the dfi_wrdata_cs_n signal, it is compatible with the standard DFI.
Read DataInterface	Only the dfi_rddata and dfi_rddata_valid signals are retained.
UpdateInterface	A custom interface is used.
StatusInterface	Only dfi_init_complete is retained.
TrainingInterface	NA.
Low PowerControl Interface	NA.
ErrorInterface	Compatible.

Table 2-21 Detailed Comparison of Differences between DFI Interface and Standard DFI

Interface Signal Classification	Standard DFI	Simplified DFI	Signal Source	Description
Control signals	dfi_address	dfi_address	MC	DFI address bus.
	dfi_bank	dfi_bank	MC	DFI bank address bus.
	dfi_cas_n	dfi_cas_n	MC	DFI column address strobe, only applicable to DDR3.
	dfi_cid	NA	MC	DFI Chip ID.
	dfi_cke	dfi_cke	MC	DFI clock enable.
	dfi_cs_n	dfi_cs_n	MC	DFI chip select.
	dfi_odt	dfi_odt	MC	DFI on-die termination control bus
	dif_ras_n	dif_ras_n	MC	DFI row address strobe bus, only applicable to DDR3.
	dfi_reset_n	dfi_reset_n	MC	DFI chip reset.
	dfi_we_n	dfi_we_n	MC	DFI write enable signal, only applicable to DDR3.
Write Data Signals	dfi_wrdata	dfi_wrdata	MC	DFI write data.
	dfi_wrdata_cs_n	NA	MC	DFI Write Data Chip Select.
	dfi_wrdata_en	dfi_wrdata_en	MC	DFI write enable signal.
	dfi_wrdata_mask	dfi_wrdata_mask	MC	DFI Write data byte mask.
Read Data Signals	dfi_rddata	dfi_rddata	PHY	DFI interface data.
	dfi_rddata_cs_n	NA	MC	DFI Read Data Chip Select.
	dfi_rdata_dbi_n	NA	PHY	Read Data DBI.
	dfi_rddata_en	NA	MC	Read data enable.
	dfi_rddata_valid	dfi_rddata_valid	PHY	Read data valid on DFI interface.
	dfi_rddata_dnv	NA	PHY	DFI data not valid.
Update Signals	dfi_ctrlupd_ack	NA	PHY	MC-initiated update acknowledge.
	dfi_ctrlupd_req	NA	MC	MC-initiated update request.
	dfi_phyupd_ack	dfi_phyupd_ack	MC	PHY-initiated update acknowledge.
	dfi_phyupd_req	dfi_phyupd_req	PHY	PHY-initiated update request.
	dfi_phyupd_type	NA	PHY	PHY-initiated update select.
Status Signals	dfi_alert_n	NA	PHY	CRC or parity error indicator.
	dfi_data_byte_disable	NA	MC	Data byte disable.
	dfi_dram_clk_disable	NA	MC	DRAM clock disable.
	dfi_freq_ratio	NA	MC	DFI frequency ratio indicator.
	dfi_init_complete	dfi_init_complete	PHY	PHY initialization complete.
	dfi_init_start	NA	MC	DFI setup stabilization or frequency change initiation.
	dfi_parity_in	NA	MC	Parity value.

Interface Signal Classification	Standard DFI	Simplified DFI	Signal Source	Description
DFI Training Signals	dfi_calvl_capture	NA	MC	CA training capture.
	dfi_phy_calvl_cs_n	NA	PHY	CA training chip select.
	dfi_calvl_en	NA	MC	PHY CA training logic enable.
	dfi_calvl_req	NA	PHY	PHY-initiated CA training request.
	dfi_calvl_resp	NA	PHY	CA training response.
	dfi_lvl_pattern	NA	MC	Training pattern.
	dfi_lvl_periodic	NA	MC	Training length indicator (full or periodic).
	dfi_phylvl_ack_cs_n	NA	MC	DFI PHY training chip select acknowledge.
	dfi_phylvl_req_cs_n	NA	PHY	DFI PHY training chip select request.
	dfi_phy_rdlvl_cs_n	NA	PHY	Read training chip select for read data eye training.
	dfi_phy_rdlvl_gate_cs_n	NA	PHY	Read training chip select for gate training.
	dfi_phy_wrlvl_cs_n	NA	PHY	Write leveling chip select.
	dfi_rdlvl_en	NA	MC	PHY read data eye training logic enable.
	dfi_rdlvl_gate_en	NA	MC	PHY gate training logic enable.
	dfi_rdlvl_gate_req	NA	PHY	PHY-initiated gate training request.
	dfi_rdlvl_req	NA	PHY	PHY-initiated read data eye training request.
	dfi_rdlvl_resp	NA	PHY	Read training response.
	dfi_wrlvl_en	NA	MC	PHY write leveling logic enable.
	dfi_wrlvl_req	NA	PHY	PHY write leveling request.
	dfi_wrlvl_resp	NA	PHY	Write leveling response.
	dfi_wrlvl_strobe	NA	MC	Write leveling strobe.
Low Power Control Signals	dfi_lp_ack	NA	PHY	Low power acknowledge
	dfi_lp_ctrl_req	NA	MC	Low power opportunity control request.
	dfi_lp_data_req	NA	MC	Low power opportunity data request.
	dfi_lp_wakeup	NA	MC	Low power wakeup time.
Error Signals	dif_error	dif_error	PHY	DFI Error.
	dfi_error_info	NA	PHY	DFI Error Info.

2.5.4.2 DFI interface timing

The working clock frequency ratio of MC to PHY is 1:4, meaning each MC instruction corresponds to 4 Phase PHY instructions. Instructions and data can be arranged across the PHY's 4 Phases as required. When sending control instructions through the DFI interface, the user must strictly adhere

to the DDR SDRAM requirements for instruction and data timing delays. DFI instructions can only be received after dfi_init_complete is pulled high, with PHY data alignment for read operations.

2.5.4.2.1 DFI interface write timing

To write data to DDR3 SDRAM, a write instruction must be sent first, followed by data, with typical timing as shown in Figure 2-25.

- Control lines for write operations: dfi_cs_n, dfi_ras_n, dfi_cas_n, dfi_we_n, dfi_cke, dfi_odt.
- Address lines for write operations: dfi_bank, dfi_address.
- Data lines for write operations: dfi_wrdata_en, dfi_wrdata, dfi_wrdata_mask.

2.5.4.2.2 DFI interface read timing

To read data from DDR3 SDRAM, a read instruction must first be sent, then data received through the PHY via DFI, with typical timing as shown in Figure 2-26.

- Control lines for read operations: dfi_cs_n, dfi_ras_n, dfi_cas_n, dfi_we_n, dfi_cke, dfi_odt.
- Address lines for read operations: dfi_bank, dfi_address.
- Data lines for read operations: dfi_rddata, dfi_rddata_valid.

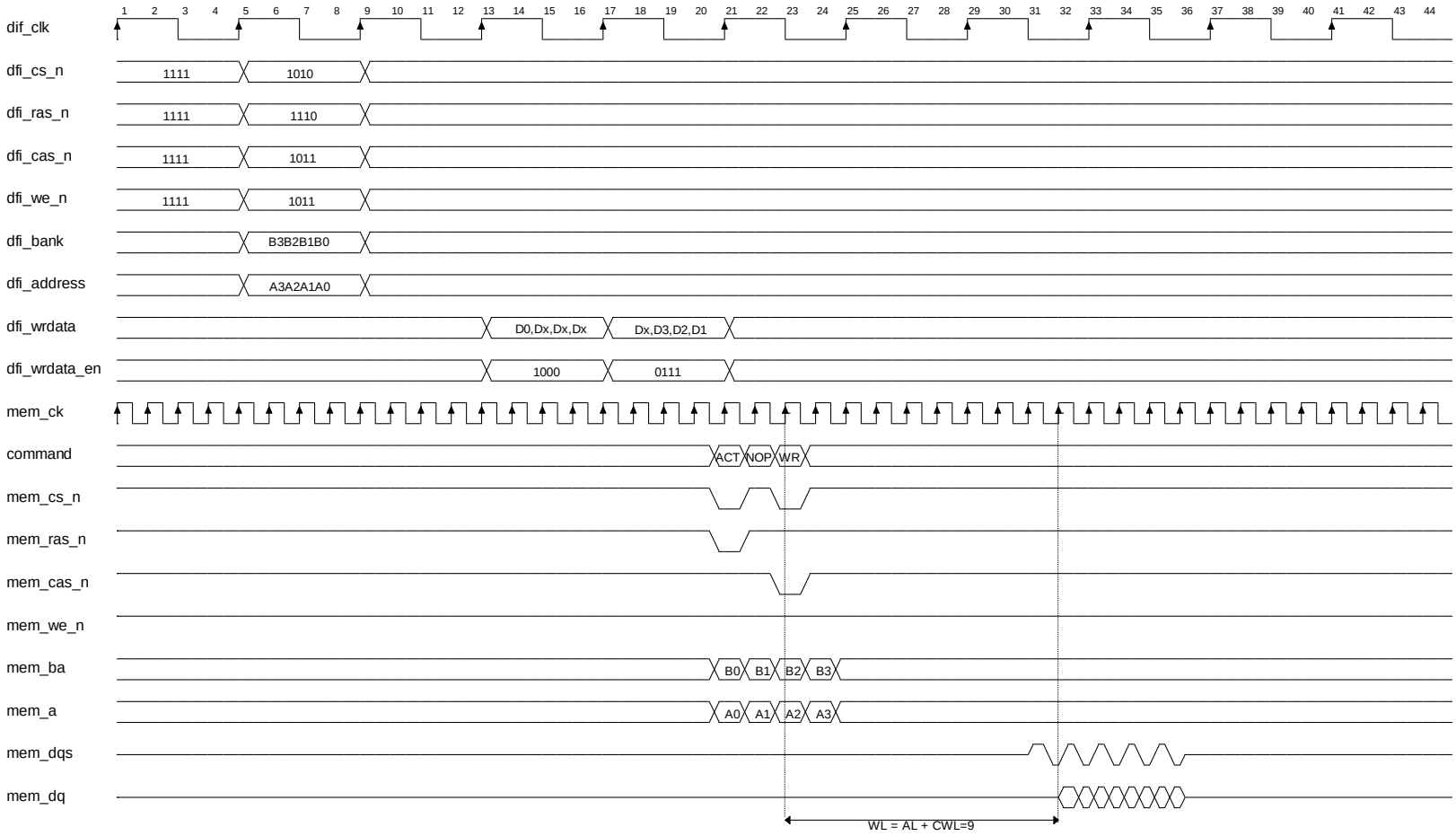


Figure 2-25 DFI Interface Write Operation Timing

Note: This timing diagram is a simulation with WL=9.

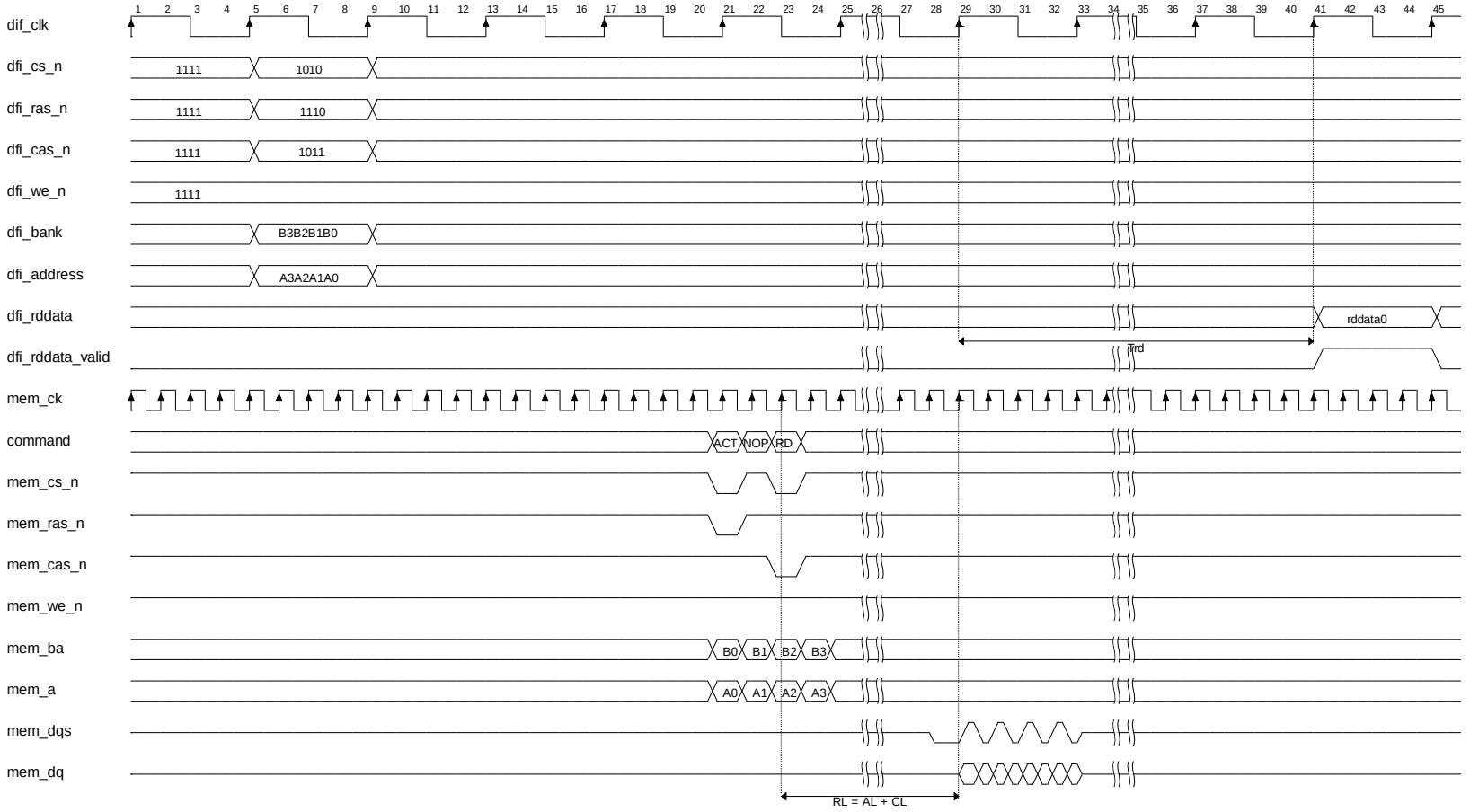


Figure 2-26 DFI Interface Read Operation Timing

Note: This timing diagram is a simulation with Trd of 7 dfi_clk cycles; the actual on-board timing may differ from the above.

2.5.5 Debug Data Description

2.5.5.1 debug_data description

For the description of the meanings of each field of debug_data, refer to Table 2-22.

Table 2-22 Definitions of debug_data Fields

Bits	Debug Port	Description
3:0	coarse_slip_step	Coarse adjustment position obtained from dqs gate training
5:4	read_clk_ctrl	Fine adjustment position obtained from dqs gate training
8:6	gate_win_size	Number of valid windows for dqs gate training
9	samp_win_valid	Number of valid read data dq windows is greater than the minimum number to view
17:10	samp_win_size	Number of valid read data dq windows
25:18	samp_center_position	Centre position of read data dq window
33:26	wrlvl_step	dqs delay steps obtained when write leveling is complete

2.5.5.2 Description of debug_calib_ctrl

For the description of debug_calib_ctrl fields, refer to Table 2-23.

Table 2-23 Definitions of debug_calib_ctrl Fields

Bits	Debug Port	Description
3:0	dbg_upcal	Update control state machine
8:4	dbg_rdcal	MPR read calibration control state machine
13:9	dbg_wrlvl	write leveling calibration control state machine
18:14	dbg_init	Power-up initialization process control state machine
21:19	dbg_main	Master state machine for training

2.5.5.3 Description of dbg_slice_state

For the description of dbg_slice_state fields, refer to Table 2-24.

Table 2-24 Definitions of dbg_slice_state Fields

Bits	Debug Port	Description
2:0	dbg_slice_rdchk_state	State output of the read calibration data comparison state machine.
6:3	dbg_slice_rdcal_state	State output of the read calibration control DQS delay state machine.
9:7	dbg_slice_gate_state	State output of the adjustment of DQS gate delay state machine.
12:10	dbg_slice_wrlvl_state	State output of the control state machine for write leveling delay.

2.6 Description of the IP Register

This section provides the HMIC_S IP related register description and access methods.

2.6.1 Register Description

2.6.1.1 Internal register of the Controller

2.6.1.1.1 MODE_REG_0_ADDR

This register has a bit width of 16 bits, and the access address is 0x00.

Table 2-25 Definitions of MODE_REG_0_ADDR Bits

Bits	Item	Reset Values	Access Type	Description
1:0	BL	2'b00	R	Burst Length.
2	CL[0]	Instantiation value	R/W	CAS Latency, CL bit 0.
3	RBT	Instantiation value	R/W	Read Burst Type.
6:4	CL[3:1]	Instantiation value	R/W	CAS Latency, CL bits [3:1], with a maximum value of 8.
7	TM	1'b0	R	Mode.
8	DLL	0	R	DLL Reset.
11:9	WR	Instantiation value	R/W	Write recovery for autoprecharge.
12	PPD	1	R	DLL Control for Precharge PD.
15:13	Reserved	0	R	Reserved.

2.6.1.1.2 MODE_REG_1_ADDR

This register has a bit width of 16 bits, and the access address is 0x01.

Table 2-26 Definitions of MODE_REG_1_ADDR Bits

Bits	Item	Reset Values	Access Type	Description
0	DLL	1'b0	R	DLL Enable.
1	D.I.C[0]	1'b0	R/W	Output Driver Impedance Control, D.I.C bit 0.
2	Rtt_Nom[0]	Instantiation value	R/W	Rtt_Nom, bit 0.
4:3	AL	2'b10	R	Additive Latency.
5	D.I.C[1]	Instantiation value	R/W	Output Driver Impedance Control, D.I.C bit 1.
6	Rtt_Nom[1]	1'b0	R/W	Rtt_Nom bit 1.
7	Level	1'b0	R	Write leveling enable. For detailed values, please refer to JESD79-3E.
8	Reserved	1'b0	R	Reserved.
9	Rtt_Nom[2]	Instantiation value	R/W	Rtt_Nom bit 2.
10	Reserved	1'b0	R	Reserved.
11	TDQS	1'b0	R	TDQS enable.
12	Qoff	1'b0	R	Qoff.
15:13	Reserved	3'b000	R	Reserved.

2.6.1.1.3 MODE_REG_2_ADDR

This register has a bit width of 16 bits, and the access address is 0x02.

Table 2-27 Definitions of MODE_REG_2_ADDR Bits

Bits	Item	Reset Values	Access Type	Description
2:0	PASR	3'b0	R	Partial Array Self-Refresh (Optional).
5:3	CWL	Instantiation value	R/W	CAS write Latency.
6	ASR	1'b0	R	Auto Self-Refresh.
7	SRT	1'b0	R	Self-Refresh Temperature Rang.
8	Reserved	1'b0	R	Reserved.
10:9	Rtt_WR	2'b00	R/W	Rtt_WR.
15:11	Reserved	5'b0	R	Reserved.

2.6.1.1.4 MODE_REG_3_ADDR

This register stores all configuration bits of MR3, with a bit width of 16 bits and an access address of 0x03. Refer to Table 2-28 for the definition of each Bit.

Table 2-28 Definitions of MODE_REG_3_ADDR Bits

Bits	Item	Reset Values	Access Type	Description
[15:0]	Reserved	16'h0	R	For detailed values, please refer to JESD79-3E.

2.6.1.1.5 CTRL_MODE_DATA

This register stores user-issued instructions (MRS, Low Power, Normal) with an access address of 0x04. Refer to Table 2-29 for the definition of each Bit.

Table 2-29 Definitions of CTRL_MODE_DATA Bits

Bits	Item	Reset Values	Access Type	Description
0	enable	1'b0	R/W	Trigger enable for DDR core status switching.
13:1	Reserved	13'h0	R	Reserved.
15:14	select	2'b00	R/W	DDR Core status selection. 00: Normal; 01: MRS; 10: Self-Refresh; 11: Power Down.

2.6.1.1.6 STATUS_REG_DATA

This register stores the current status of DDR, with an access address of 0x05. Refer to Table 2-30 for the definition of each Bit.

Table 2-30 Definitions of STATUS_REG_DATA Bits

Bits	Item	Reset Values	Access Type	Description
1:0	mode_state	2'b0	R	Stores the current status of DDR3. 2'b00: Normal State; 2'b01:Self-Refresh State; 2'b10:Power Down State; 2'b11:MRS State.
14:2	Reserved	13'b0	R	Reserved
15	busy	1'b0	R	Whether the current request of DDR3 has been responded to: 1'b0: Current request has been responded to 1'b1: Current request has not been processed

2.6.1.2 PHY internal registers

DDR PHY internal registers support DDR3 and are fully consistent with the protocol. For detailed information, please refer to "***JESD79-3D, DDR3 SDRAM Standard***".

2.6.2 Register Access

The internal registers of the Controller are accessed via the APB interface, with register data transferred through apb_wdata and apb_rdata, and the register address through apb_addr. Refer to "2.6.1 Register Description" for the address values of each register.

The method for accessing the PHY internal registers is consistent with the method for accessing the Memory internal Mode Register.

2.7 Typical Applications

For typical applications of HMIC_S IP, please refer to "2.4 Example Design".

2.8 Descriptions and Considerations

2.8.1 Simplified AXI4 Interface Burst Calculation

Start_Address = ADDR;

Burst_Length= len;

End_Address = ADDR + len * 8.

2.8.2 Clock Constraints

Multiple clocks within the HMIC_S IP require constraints, namely ref_clk, ddrphy_clkin, and ioclk. The relation between these clocks is shown as Figure 2-27. For specific constraint methods, please refer to the .fdc files in the "IP/pnr" directory.

Herein, ref_clk is the input reference clock, ddrphy_clkin is obtained by PLL multiplication followed by division through GTP_IOCLKDIV, and ioclk is acquired by PLL multiplication. ddrphy_clkin is the system clock for the IP's soft logic, while ioclk is a fast clock.

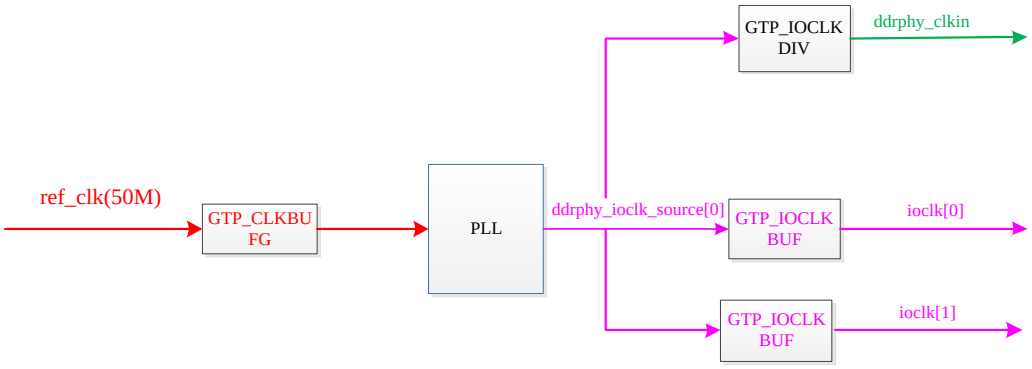


Figure 2-27 IP Internal Clock Structure

2.8.3 IO Constraints

Position constraints for the chip's internal modules must be consistent with the location constraints provided by the .fdc file, the constraint location cannot be changed, and the routing should be modified according to specific usage.

Constraints on pin positions and level standards, among other parameters, are set in the UCE (User Constraint Editor), for which specific configuration can be referred to from the .fdc files in the "IP/pnr" directory. The requirements for pin parameter settings are as follows.

- VCCIO needs to be set to 1.5V (DDR3 as 1.5V);
- mem_ck is a differential output signal and requires SSTL15D_I level;
- mem_dqs is a bidirectional differential signal and requires SSTL15D_I level;
- All other signals are single-end signals and require SSTL15_I level;
- mem_dqs/mem_dq signals need terminal resistors at the FPGA end, and it is recommended to use internal termination resistors by setting DDR_TERM_MODE to ON;
- All other parameters can use the default settings.

2.8.4 Route Constraints

If the DDR project requires priority route constraints, the priority route information is stored in the inst_name.rcf file in the "/pnr" directory; the Example Design project will automatically load this file. If users create their own project, they need to manually load the file. Please refer to the help document under the PDS installation path: "***Route_Constraint_Editor_User_Guide***".

2.8.5 DM_GROUP_EN Parameter Configuration Description

For scenarios where DM signals are placed in the same GROUP, the parameter DM_GROUP_EN has been added to ddrphy_top. When this parameter is configured to 0, it's determined that DM signals and DQ signals are in the same GROUP; when configured to 1, DM signals are all placed in the GROUP with the lower eight DQs, compatible with client application scenarios.

Attention:

This application scenario currently only supports a bit width of 16 bits.

2.9 IP Debugging Method

2.9.1 Key Indicator Signal

For key information during the operation of DDRPHY, single-bit indicators are created for easy observation, which can be connected to external LEDs or monitored in other ways to quickly determine the operating status of DDRPHY. For descriptions of key indicator signal, refer to Table 2-31.

Table 2-31 Key Indicator Signal

Signal Name	I/O	Bit width	Description
ddr_init_done	O	1	PHY has completed training operations and is now in a normal state.
heart_beat_led	O	1	Heartbeat signal. When the ddrphy system clock is normal, it flashes approximately once per second. (Signal generated in Example Design)
err_flag_led	O	1	Data detection error signal 1: Data errors detected, which need to be manually cleared after troubleshooting. 0: No data error detected. (Signal generated in Example Design)

2.9.2 Internal Status and Control Signals

For other internal status signals and control signals used for debugging that do not change in real-time, refer to Table 2-19. These signals can be conveniently read and written through the Serial Port. Flexible debugging can be achieved by developing accompanying scripts.

For a Serial Port access diagram, refer to Figure 2-14. In the DDRPHY Example Design, the uart_ctrl module is a Serial Port communication module, and the host computer reads and writes internal status signals and control signals through the Serial Port and uart_ctrl module.

For the Serial Port configuration used in DDRPHY Example Design, refer to Table 2-32.

Table 2-32 DDRPHY Example Design Serial Port Configuration

Baud Rate	**Start Bit**	**Data Bits**	**Stop Bits**	**Checksum Bits**	**Flow Control**
115200bps	1bit	8bit	1bit	None	None

Description:

A set of Serial Port scripts was developed during the internal debugging process. Please contact AE to obtain if needed.

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its series products at any time without prior notice.