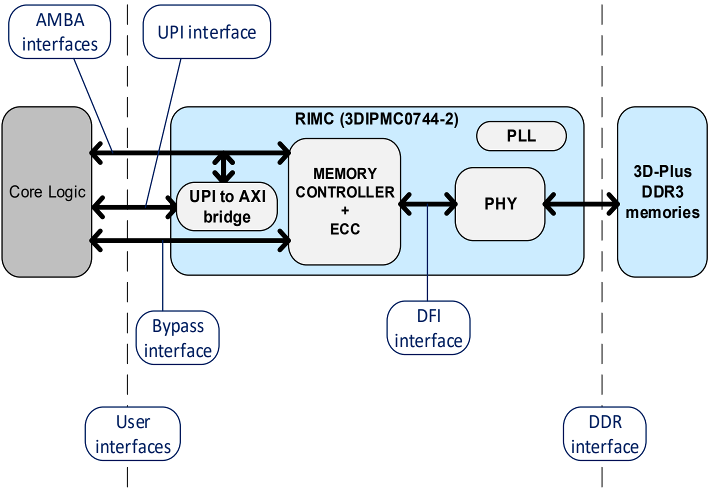
1 FEATURES

- Configurable via AMBA user interface - compliant AXI/AHB/APB
- DDR PHY interface compliant DFI 2.1
- Provides rank and bank management algorithms
- Dynamically configurable via an 8-bit APB slave interface
- Controller Bypass mode provides direct access to memories
- Selectable Hamming or Reed-Solomon Error Correction Code (ECC)
- Configurable DDR3 ranks to increase memory capacity
- Clock & ODT settings compatible with 3D PLUS modules
- Selectable Burst-of-4 or Burst-of-8 accesses
- Selectable Enable/Disable of DDR3 scrubbing
- User-definable scrubbing frequency
- Selectable DRAM refresh time
- Configurable user data width from 8 to 128 bits



2 OVERVIEW

The Radiation Intelligent Memory Controller (RIMC DDR3) is a fully configurable DDR3 SDRAM memory controller IP core designed to work with 3D PLUS DDR3 memory modules to achieve radiation tolerance improvement. The RIMC contains all standard functions of a DDR memory controller for data width applications from 8b up to 144b. It includes alsoadditional functions, such as Single Event Upset (SEU) mitigation and Single Event Functional Interrupt (SEFI) protection (later described in paragraph 4.2) to be able to work in radiation environments.

In addition to the memory controller IP within the RIMC IP core, the user can either select PHY IPs provided in the IP core or develop its own PHY IP.

The RIMC has two primary interfaces, the user interface (UIF) and the DDR memory interface (DIF).

The UIF is comprised of at least one AHB bus, AXI bus or UPI, and also contains an APB bus for user dynamic configuration:

- Slave APB interface dedicated to internal registers
- Optional slave AHB/AXI interface
- Optional UPI interface (interface compatible with MIG from Xilinx)
- Optional Bypass interface

The Memory Controller is defined by the previously described UIF and the DIF, compliant to DFI 2.1, to send commands and data to the DDR memory components through the DDR PHY (which is uniquely configured to work with specific FPGAs/ASICs).

Table Of Content

TABLES and FIGURES

3 RIMC CONFIGURATION

In order to use the RIMC to access to 3D PLUS DDR3 Memory modules: first of all, dedicated RIMC configuration registers must be set up. This paragraph describes how to configure the RIMC through registers and generics.

Memory Setting

The Memory Setting configuration data are dependent on the 3D PLUS memory modules' characteristics, which can be found within 3D PLUS's memory module Detail Specifications. Below are the items that should be configured.

- The burst length (BURST_LEN generic) shall be set to select either a Burst-of-4 or a Burst-of-8
- The user data bus width (DATA_WIDTH generic) should be selected by the application, and the ECC width (ECC_WIDTH generic) should be selected in accordance with paragraph 4.2.1;
- The number of control signals, ranks, banks and rows shall be configured per Table 33.

The other configuration data are dependent on the end-user's application.

Clock Frequency

The Clock should be configured in accordance with paragraph 4.1.

Error Management

Error Management should be configured in accordance with paragraph 4.2; Below are the items which should be configured.

- ECC setting
- Scrubbing setting
- SEFI protection setting

Interface Setting

Interface Setting should be configured in accordance with paragraph 4.3. Below are the items which should be configured.

- Core logic Interface AXI/AHB/APB
- DFI Interface
- Bypass mode (Direct DDR3 Access)
- Uncorrectable errors Interruption

4 MEMORY CONTROLLER

In addition to all of the standard functions of an Synchronous DRAM memory controller, the Memory controller contains the functions described in the following paragraphs.

The RIMC IP is responsible for the following standard functions:

- Bank management
- Conversion of user AMBA commands to DFI compatible DDR commands.

The RIMC IP also embeds the following additional functions:

- Data error management including an ECC and a scrubbing mechanism;
- SEFI detection and correction mechanisms;

4.1 CLOCK, RESET AND INITIALIZATION

The RIMC needs 3 internal clocks. The first clock, SysClk, is dedicated to the Memory Controller as well as the AMBA user interfaces. The second clock, DFIClk, is dedicated to the DFI interface. These 2 clocks shall be synchronous with a frequency ratio of 1:1, 2:1 or 4:1 (FREQ_RATIO). The third clock has a 200MHz fixed frequency, and is used for the Xilinx serie 7 PHY only.

DFIClk and SysClk are generated by an RIMC embedded PLL from ClkIn.

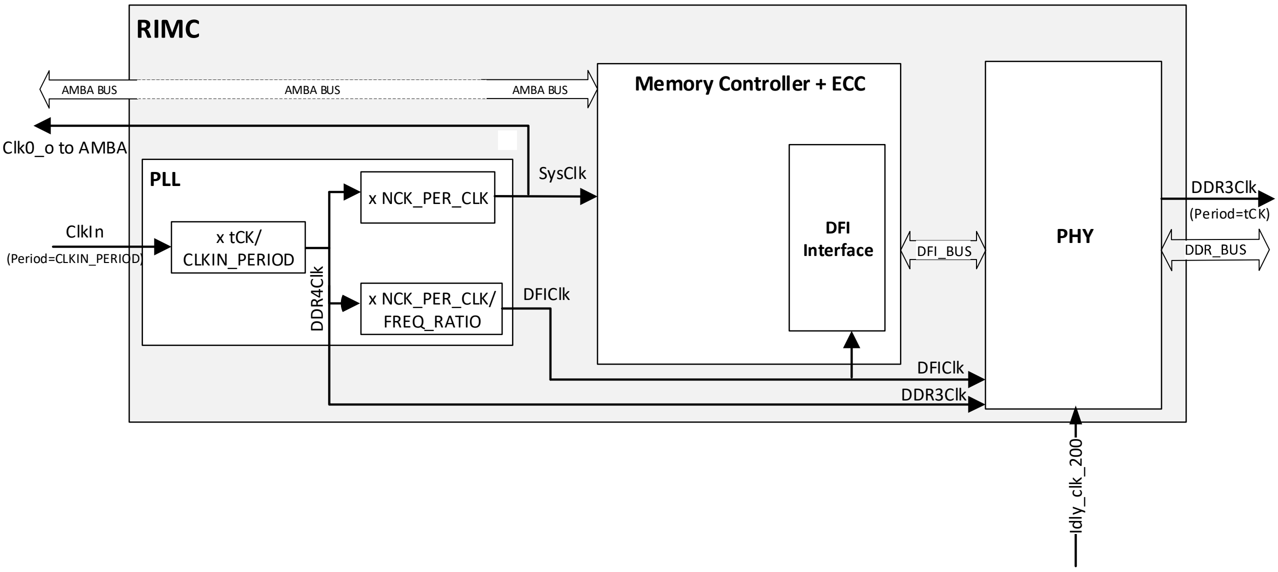


Figure 1: Embedded clock management

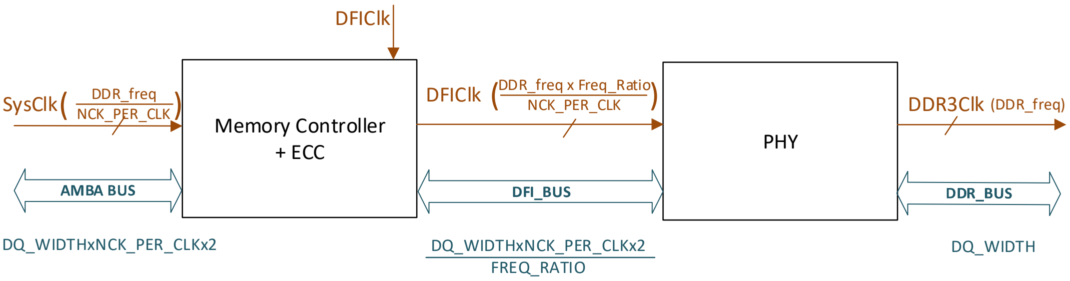


Figure 2: Clocks frequency ratio configuration

FPGA Type	FREQ_RATIO	nCK_PER_CLK
Xilinx 7-series	1	2, 4
Ultrascale	1	4

Table 1: Frequency ratio generics configuration

One asynchronous input reset (Rst_N) must be provided.

After deassertion of the reset input port (Rst_N), the RIMC automatically configures itself by reading the following configuration input pins:

- SCRUB_EN: This input port can enable or disable the scrubbing mechanism. When activated the scrubbing performs read and write accesses across the entire memory array after the initialization phase has been completed.
- REF_MODE: This input port is used to select 32ms or 64ms refresh time

After this automatic configuration process, the user may dynamically update the internal configuration using the APB slave interface.

A memory initialization sequence is recommended after powering up the User Memory, and this initialization sequence can be enabled through the INIT_Pad APB register. When enabled, the RIMC IP Core initializes the entire memory array with a fixed value as soon as the Physical layer is ready (after the completion of the Power-up sequence and calibration phase). The Byte value to be written on the entire memory array is contained in the Init_Reg register.

If the initialization phase is not performed and a Read is requested by the user to an address prior to a Write at the same address, the RIMC IP Core may detect potential ECC errors.

4.2 Error management

4.2.1 Error Correction Code (ECC)

The RIMC SEU mitigation scheme uses different configurable ECCs (Hamming or Reed-Solomon) and scrubbing to correct SEUs and Single Event Row Errors (SERE). For example, using a Reed-Solomon code for 32b data and 50% overhead [2 * RS(6;4), m=4, Global Bus = 48bits], the RIMC can correct up to 8 bits of error (row error) in one die per 48b, and 2 SEUs in the same address of different dice per 48b. In cases where scrubbing is applied, the worst case (in which one particle creates 2 upsets in 2 dice) error rate will be 3.8E-9 upset/day/module.

The RIMC implements an Error Correction Code (ECC) with additional DDR3 components. The supported configurations are described in Table 2.

Two different ECC types are handled by the RIMC:

- Hamming, which is able to mitigate SEU
- Reed-Solomon, which is able to mitigate SEU, MBU and SERE.

DDR3 Component Width	Data Bus Width (DATA_WIDTH)	(ECC_WIDTH)	Global Bus Width (DQ_WIDTH)	ECC Type	Correction capability
8	8	0	8	No ECC	none
8	8	8	16	RS(4;2) with m=4	Correct a 4 bits word
8	16	0	16	No ECC	none
8	16	8	24	RS(6;4) with m=4	Correct a 4 bits word
8	16	16	32	2*RS(4;2) with m=4	Correct a 8 bits DDR3 component
8	32	0	0	No ECC	none
8	32	8	40	RS(10;8) with m=4	Correct a 4 bits word
8	32	16	48	2*RS(6;4) with m=4	Correct a 8 bits DDR3 component
8	64	0	64	No ECC	none
8	64	32	96	4*RS(6;4) with m=4	Correct a 16 bits word, (2 contiguous DDR3 components)
8	24/40/48/56/ 72/80/88/96	0	24/40/48/56/ 72/80/88/96	No ECC	none
8	128	16	144	2*(Hamming 64+8)	2 bits in error (Note 1)
16	16	0	16	No ECC	none
16	32	0	32	No ECC	none
16	32	32	64	4*RS(4;2) with m=4	Correct a 16 bits DDR3 component
16	64	8	72	Hamming 64+8	Correct 1 bit in error (Note 1)
16	64	32	96	4*RS(6;4) with m=4	Correct a 16 bits DDR3 component
16	48/80/96	0	48/80/96	No ECC	none
16	128	16	144	2*(Hamming 64+8)	2 bits in error (Note 1)

Table 2: Supported ECC codes

Note 1: This ECC utlilizes a Hamming code. This code CANNOT correct for a DDR3 component failure but it is able to correct a SEU.

To use this feature additional DDR3 components are added in parallel (data path increased) on the board.

The RIMC can be used with 8-bit and 16-bit DDR3 component widths.

Below is a list of available 3D PLUS DDR3 modules with their configurations:

DDR3 Module P/N	Configuration	Access/Clock	Package	DDR3 Component Configuration
3D3D16G16YB4751	512M x 16	300-667Mhz	BGA95	512Mx8
3D3D16G72WB2723	256M x 72	300-667Mhz	BGA199	256Mx16
3D3D24G48YB3732	512M x 48	300-667Mhz	BGA143	512Mx8

Table 3: 3D PLUS DDR3 modules

4.2.2 SCRUBBING FUNCTIONALITY

A scrubbing mechanism can be dynamically enabled with the SCRUB_EN input pin or the ScrubEn_Reg register.

When enabled, the RIMC IP core periodically reads the entire memory array and then writes the corrected data back into the memory array. The frequency of the scrubbing is user-defined through the Scrub_Reg register.

An Example of Scrub_Reg configuration:

- The RIMC is connected to a single 3D3D16G72WB2723 (BGA package)
- User wants to scrub the full memory array in 60 seconds (t).
- ddrClk frequency (F_{DDR}) is 300 MHz
- Burst length is set to 8

Scrub_Reg = BURST_LENGTH * t * F_{DFI} / 256M = 562 clock cycles

In this example the RIMC performs one Read/Write Burst every 562 clock cycles.

4.2.3 SEFI PROTECTION

Traditionally, SEFI mitigation has involved power cycling after the occurrence of a SEFI. However, power cycling will lead to data loss. To avoid data loss, a specific SEFI protection technique has been designed in the RIMC IP Core to prevent SEFI and to replace the traditional power cycling strategy. This SEFI protection is integrated in the RIMC IP core and is transparent to the user.

To configurate the device SEFI protection, the Bypass_Reg register should be configured as below:

b00: SEFI protection set to Strong Level
b01: SEFI protection set to Simple Level

4.3 RIMC INTERFACES

The RIMC is defined by three interfaces (see Figure 3):

- The AMBA interface
- The Direct DDR Access interface
- The DDR PHY interface, compliant to DFI 2.1

These three interfaces are described in the following paragraphs.

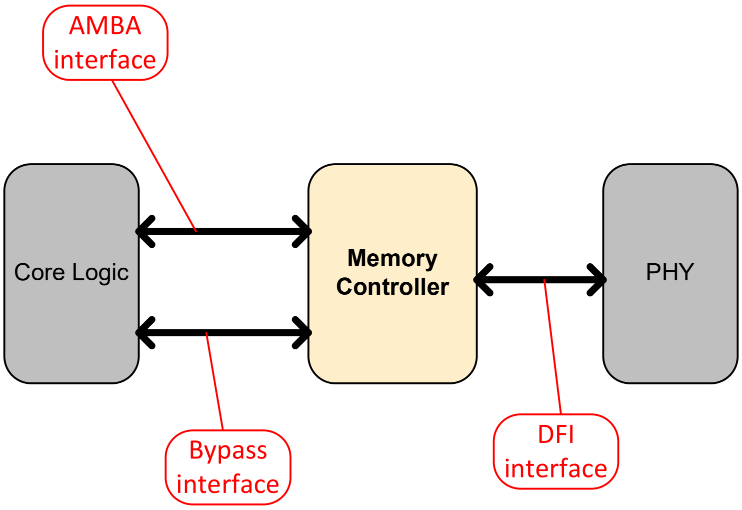


Figure 3: Interfaces with DDR controller

4.3.1 AMBA INTERFACES

Three AMBA interfaces are available for connection to the Memory Controller:

- AMBA AXI: 0 to 8 ports can be instantiated through the NB_AXI_SLV generic.
- AMBA AHB: 0 to 8 ports can be instantiated through the NB_AHB_SLV generic
- AMBA APB: This interface is used to configure the IP.

At least one port shall be instantiated to be synthesizable, i.e. NB_AXI_SLV >= 1 and NB_AHB_SLV >= 1

The Memory Controller can be configured by the core logic using the slave APB interface dedicated to internal registers.

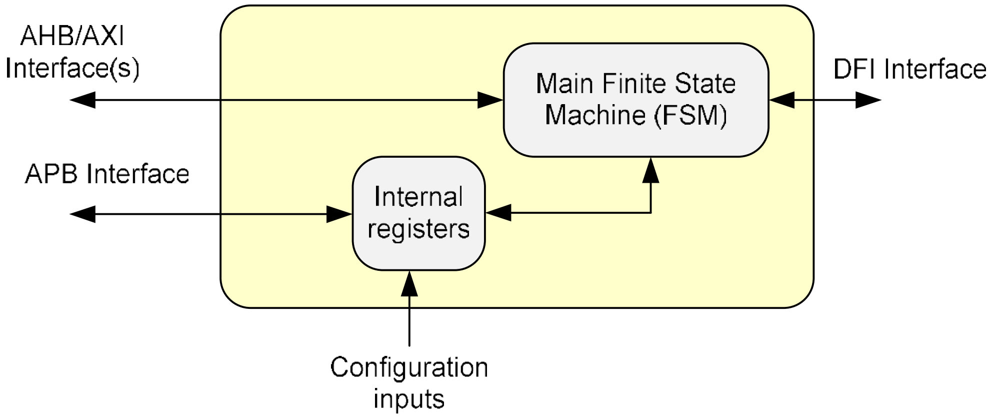


Figure 4: Configuration management

4.3.1.1 AXI interface

The Memory Controller is compliant to the AMBA Advanced eXtensible Interface (AXI) protocol in version 3.0. Up to 8 slaves AXI interfaces can be instantiated inside the RIMC through the NB_AXI_SLV generic.

The data bus width of both the write channel and the read channel can be configured to 16, 32, 64, 128, 256 or 512 bits.

*Note: To realize optimal bandwidth, AXI data width should be greater than or equal to “DDR data bus” width * 2 * nCK_PER_CLK.*

The key features of the AXI protocol are:

- separate address/control and data phases
- support for unaligned data transfers using byte strobes
- burst-based transactions with only start address issued
- separate read and write data channels to enable Direct Memory Access (DMA)
- ability to issue multiple outstanding addresses
- out-of-order transaction completion
- easy addition of register stages to provide timing closure.
- Fastest AMBA interfaces for RIMC

4.3.1.2 AHB interface

The Memory Controller is compliant with AHB slave interface data widths of 16, 32, 64, 128, 256 and 512 bits. RIMC is compatible of AHB version 3.0

*Note: To realize optimal bandwidth, AHB data width should be greater than or equal to "DDR data bus" width * 2 * FREQ_RATIO.*

The AHB slave interface provides the following input signals for each AHB port:

- HSEL: Slave select
- HADDR: Address bus
- HWRITE: Transfer direction (0 for Read / 1 for Write)
- HTRANS: Transfer type
- HSIZE: Transfer size
- HBURST: Burst type
- HWDATA: Write data bus
- HPROT: Protection control. This signal is not used by RIMC IP Core
- HREADY: Transfer done
- HMASTER: Master number. This signal is not used by RIMC IP Core
- HMASTLOCK: Locked sequence

The AHB slave interface provides the following output signals for each AHB port:

- HREADY: Transfer done
- HRESP: Transfer response
- HRDATA: Read data bus
- HSPLIT: Split completion request. The RIMC does not implement the SPLIT functionality so this output is driven LOW.

The slave AHB accepts the following burst types:

- SINGLE
- INCR4
- INCR8
- INCR16

The Memory Controller implements neither the RETRY nor the SPLIT functions. The Memory Controller does not implement the BURST with an unknown number of beats.

An HRESP_ERROR is generated by the RIMC when an uncorrectable error is detected during a Read operation.

4.3.1.3 APB interface

An APB bus is instantiated inside the Memory Controller.

The Memory Controller is compliant with the slave APB interface data width of 8 bits. The Memory Controller is compatible of APB version 2.0 .

The APB interface can be used to configure the Memory Controller internal registers (the complete list of registers is provided in paragraph 4.5).

4.3.2 BYPASS INTERFACE

In addition to the AHB/AXI interfaces the Memory Controller provides a direct path to the DDR3 memory array in a *Controller Bypass* mode.

The following signals are provided at the user interface:

- user_ras_n
- user_cas_n
- user_we_n
- user_addr
- user_ba
- user_cs_n
- user_cke
- user_odt
- user_reset_n
- user_rddata_en
- user_dqiv
- user_dqi
- user_wdata_en
- user_dqo
- user_dm

When using this interface, signals are transferred directly to the DDR3 memory array without any reformatting inside the Memory Controller, therefore it is the user's responsibility to insure that the signals are in conformance with the applicable DDR3 memory datasheets/Detail Specification.

4.3.3 DFI INTERFACE

The Memory Controller implements a DDR PHY Interface (DFI) compatible to DFI 2.1. The DFI is a standardized interface that defines the connectivity between a DDR memory controller and a DDR physical interface (PHY) for DDR1, DDR2, LPDDR2, DDR3, DDR3L.

PHY IP is available from 3D PLUS, supporting multiple targets (see paragraph 8)

4.4 MEMORY CONTROLLER GENERICS/PARAMETERS AND PORTS

The Memory Controller contains the ports defined in

Name	Width	Direction	Functionality
SysClk	1	I	Main input clock used for all the RIMC except DFI interface
DFIClk	1	I	Input clock used for DFI interface. Can be SysClk, 2* SysClk or 4* SysClk frequency
Rst_N	1	I	Input reset synchronous to SysClk, active LOW
DFIRst_N	1	I	Input reset synchronous to DFIClk, active LOW
SCRUB_EN	1	I	0: Scrubbing is disabled (or enabled through the ScrubEn_Reg register) 1: Scrubbing is enabled
REF_MODE	1	I	0: Normal refresh 1: Refresh timing is divided by 2

Name	Width	Direction	Functionality
Ahbsi	NB_AHB_SLV records	I	AHB slave input array of record type
Ahbso	NB_AHB_SLV records	O	AHB slave output array of record type
Apbi	1 record	I	APB slave input
Apbo	1 record	O	APB slave output
AXI interface			
AXI_aw_out	NB_AXI_SLV records	O	Outputs relative to Address Write Channel
AXI_aw_in	NB_AXI_SLV records	I	Inputs relative to Address Write Channel
AXI_w_out	NB_AXI_SLV records	O	Outputs relative to Write Data Channel
AXI_w_in	NB_AXI_SLV records	I	Inputs relative to Write Data Channel
AXI_b_out	NB_AXI_SLV records	O	Outputs relative to Write Response Channel
AXI_b_in	NB_AXI_SLV records	I	Inputs relative to Write Response Channel
AXI_ar_out	NB_AXI_SLV records	O	Outputs relative to Address Read Channel
AXI_ar_in	NB_AXI_SLV records	I	Inputs relative to Address Read Channel
AXI_r_out	NB_AXI_SLV records	O	Outputs relative to Read Data Channel
AXI_r_in	NB_AXI_SLV records	I	Inputs relative to Read Data Channel
DFI interface			
dfi_address	nCK_PER_CLK/FREQ_RATIO *ROW_WIDTH	O	DFI address bus
dfi_bank	nCK_PER_CLK/FREQ_RATIO *BANK_WIDTH	O	DFI bank bus
dfi_cas_n	nCK_PER_CLK/FREQ_RATIO	O	DFI column address strobe bus
dfi_cke	nCK_PER_CLK/FREQ_RATIO *CKE_WIDTH	O	DFI clock enable bus
dfi_cs_n	nCK_PER_CLK/FREQ_RATIO *CS_NUM	O	DFI chip select bus
dfi_odt	nCK_PER_CLK/FREQ_RATIO *ODT_WIDTH	O	DFI on-die-termination control bus
dfi_ras_n	nCK_PER_CLK/FREQ_RATIO	O	DFI row address strobe bus
dfi_reset_n	nCK_PER_CLK/FREQ_RATIO	O	DFI reset bus
dfi_we_n	nCK_PER_CLK/FREQ_RATIO	O	DFI write enable bus

Name	Width	Direction	Functionality
dfi_wrdata	nCK_PER_CLK/FREQ_RATIO *2*DQ_WIDTH	O	DFI Write data bus
dfi_wrdata_en	nCK_PER_CLK/FREQ_RATIO	O	Write data and data mask enable
dfi_wrdata_mask	nCK_PER_CLK/FREQ_RATIO *2*DQ_WIDTH/8	O	Write data Byte mask
dfi_rddata_en	nCK_PER_CLK/FREQ_RATIO	O	Read data enable
dfi_rddata	nCK_PER_CLK/FREQ_RATIO *2*DQ_WIDTH	I	Read data bus
dfi_rddata_dnv	nCK_PER_CLK/FREQ_RATIO *2*DQ_WIDTH/8	I	DFI data not valid. Not used by RIMC.
dfi_rddata_valid	nCK_PER_CLK/FREQ_RATIO	I	Read data valid indicator
dfi_ctrlupd_req	1	O	MC-inititiated update request
dfi_ctrlupd_ack	1	I	MC-initiated update acknowledge
dfi_phyupd_req	1	I	PHY-inititiated update request
dfi_phyupd_type	1	I	PHY-initiated update select
dfi_phyupd_ack	1	O	PHY-initiated update acknowledge
dfi_dram_clk_disable	nCK_PER_CLK/FREQ_RATIO *CS_NUM	O	DRAM clock disable.
dfi_freq_ratio	2	O	This signal is the image of FREQ_RATIO generic: b00: FREQ_RATIO=1 b01: FREQ_RATIO=2 b10: FREQ_RATIO=4 b11: Reserved
dfi_init_complete	1	I	PHY initialization complete
dfi_rdlvl_load	1	O	Not used
dfi_rdlvl_cs_n	1	O	Not used

Name	Width	Direction	Functionality
dfi_rdlvl_en	1	O	Not used
dfi_rdlvl_edge	1	O	Not used
dfi_rdlvl_delay_X	1	O	Not used
dfi_rdlvl_gate_en	1	O	Not used
dfi_rdlvl_gate_delay_X	1	O	Not used
dfi_wrlvl_load	1	O	Not used
dfi_wrlvl_cs_n	1	O	Not used
dfi_wrlvl_strobe	1	O	Not used
dfi_wrlvl_en	1	O	Not used
dfi_wrlvl_delay_X	1	O	Not used
dfi_rdlvl_resp	1	I	Not used
dfi_rdlvl_mode	1	I	Not used
dfi_rdlvl_gate_mode	1	I	Not used
dfi_wrlvl_mode	1	I	Not used
dfi_wrlvl_resp	1	I	Not used
Bypass interface			
user_ras_n	nCK_PER_CLK	I	Row Address Select
user_cas_n	nCK_PER_CLK	I	Column Address Select
user_we_n	nCK_PER_CLK	I	Write Enable, active LOW
user_address	nCK_PER_CLK* ROW_WIDTH	I	Address bus
user_bank	nCK_PER_CLK* BANK_WIDTH	I	Bank Address bus
user_cs_n	nCK_PER_CLK* CS_NUM	I	Chip Select, active LOW
uer_cke	nCK_PER_CLK* CS_NUM	I	Clock Enable
user_odt	nCK_PER_CLK* CS_NUM	I	On Die Termination
user_reset_n	nCK_PER_CLK	I	Memory reset. Used only for DDR3
user_wdata_en	nCK_PER_CLK	I	Memory write data enable
user_dm	nCK_PER_CLK*2 *DQ_WIDTH/8	I	Memory data mask
user_dqo	nCK_PER_CLK*2 *DQ_WIDTH	I	Memory write data
user_rdata_en	nCK_PER_CLK	I	Memory read data Enable

Name	Width	Direction	Functionality
user_dqi	nCK_PER_CLK*2 *DQ_WIDTH	O	Memory Read Data
user_dqiv	1	O	Memory Read data valid
MEM_CTRL_DEBUG	record	O	Debug signals

Table 4: Memory controller pinout

The Memory Controller contains the generics/parameters defined in

Name	Functionality	Possible values
RTT_NOM	Nominal ODT termination value	"75"
RTT_WR	Dynamic ODT	"OFF", "60", "120"
AL	Additive Latency option	"0", "1", "2", "3", "4", "CL-1", "CL-2"
OUTPUT_DRV	Memory output drive. For DDR3, “HIGH” is 34 Ohm and “LOW” is 40 Ohm	“HIGH”, “LOW”
HDMAX	Data width of AHB and AXI busses	16, 32, 64, 128, 256
GSYNCRST	Type of internal reset	0: Asynchronous reset 1: Synchronous reset
USE_DFF	Type of internal memories	0 : Use normal RAM blocks 1 : Use DFF
ENDIANNESS	Endianness of AHB, APB and AXI busses	0: Little endian 1: Big endian
SIMU	Activated only for simulation. Used to accelerate calibration.	0, 1
NB_AHB_SLV	Number of slave AHB busses.	0 to 8. *Note: NB_AHB_SLV+NB_AXI_SLV shall be between 0 and 8* *If the sum is null, then the MIG interface is used.*
NB_AXI_SLV	Number of slave AXI busses	0 to 8. *Note: NB_AHB_SLV+NB_AXI_SLV shall be between 0 and 8* *If the sum is null, then the MIG interface is used.*
TWO_T_TIME_EN	Mode 2T	0, 1
DFI parameters		

Name	Functionality	Possible values
tphy_wrdata	Specifies the number of DFI PHY clock cycles between when the **dfi_wrdata_en** signal is asserted to when the associated write data is driven on the **dfi_wrdata** signal	Positive integer
tphy_wrlat	Number of DFI clock cycles between a write command and the **dfi_wrdata_en** signal	Positive integer
trddata_en	Number of DFI clock cycles from the assertion of a read command to the assertion of the **dfi_rddata_en** signal	Positive integer
CLK_PERIOD	SysClk clock period, in picosecond	
FREQ_RATIO	Clock ratio between SysClk and DFIClk	1, 2, 4
nCK_PER_CLK	Clock ratio between DFIClk and memory clock	1, 2, 4
DATA_WIDTH	Full data width. See 4.2.1 for details.	8, 16, 32, 64
ECC_WIDTH	Ful ECC width. See 4.2.1 for details.	5, 6, 8, 16
DQ_WIDTH	Full memory width, including DATA and ECC DQ_WIDTH shall be DATA_WIDTH+ECC_WIDTH, rounded up to the next multiple of 8	8, 16, 24, 32, 40, 48, 64, 72, 96
Memory generics		
BURST_LEN	Memory burst length	4, 8
BURST_TYPE	Sequential or interleaved 0: Sequential 1: Interleaved	0, 1
CS_NUM	Number of ranks	1, 2, 4
BANK_WIDTH	Number of banks	2, 3
ROW_WIDTH	Number of rows	14, 15
COL_WIDTH	Number of columns	10
TWR_PS	Write recovery time, in picosecond	Depending on the DDR3 component datasheet
TWTR_PS	Write to read timing, in ps	Depending on the DDR3 component

Name	Functionality	Possible values
		datasheet
TRTP_PS	Read to Precharge timing, in ps	Depending on the DDR3 component datasheet
TRFC_PS	Refresh to active or refresh to refresh command interval	Depending on the DDR component datasheet
CAS_LAT	Read CAS latency	Depending on the DDR3 component datasheet
CWL_LAT	Write CAS latency	Depending on the DDR3 component datasheet

Table 5 : Memory Controller Generic/Parameter list

4.5 MEMORY CONTROLLER REGISTERS

The Memory Controller contains the registers defined in Table 6 . Details are given from table 7 to 31.

Registers	Address
ECC_DErr_Reg0	0x00
ECC_DErr_Reg1	0x01
ECC_SErr_Reg0	0x02
ECC_SErr_Reg1	0x03
ECC_DErr_Clr	0x04
ECC_SErr_Clr	0x05
Bypass_Reg_AD	0x06
Init_Pad_Reg	0x07
Init_Pad_Sts	0x08
Init_Pad_Ctl	0x09
ScrubEn_Reg	0x0A
Scrubbing_Reg0	0x0B
Scrubbing_Reg1	0x0C
Scrub_Ctl	0x0D
CorEn_Reg	0x0E
CorBeat_Reg	0x0F
CorVec_Reg0	0x10
CorVec_Reg1	0x11
CorVec_Reg2	0x12
CorVec_Reg3	0x13
CorVec_Reg4	0x14
CorVec_Reg5	0x15
CorVec_Reg6	0x16
CorVec_Reg7	0x17
CorVec_Reg8	0x18
CorVec_Reg9	0x19
CorVec_Reg10	0x1A
CorVec_Reg11	0x1B
ScrubAd_Sts0	0x1C
ScrubAd_Sts1	0x1D
ScrubAd_Sts2	0x1E

Registers	Address
ScrubAd_Sts3	0x1F
Version_Sts	0x20
Reset_ZQ_Ctl	0x21
Reset_ZQ_Sts	0x22
Selftest_Ctl	0x23
Selftest_Start_Add_Reg0	0x24
Selftest_Start_Add_Reg1	0x25
Selftest_Start_Add_Reg2	0x26
Selftest_Start_Add_Reg3	0x27
Selftest_Stop_Add_Reg0	0x28
Selftest_Stop_Add_Reg1	0x29
Selftest_Stop_Add_Reg2	0x2A
Selftest_Stop_Add_Reg3	0x2B
Selftest_Sts	0x2C
Selftest_Err_Cnt_Sts0	0x2D
Selftest_Err_Cnt _Sts1	0x2E
Selftest_Err_Cnt _Sts2	0x2F
Selftest_Err_Cnt _Sts3	0x30
Selftest_Err_Add_Sts0	0x40
Selftest_Err_Add_Sts1	0x41
Selftest_Err_Add_Sts2	0x42
Selftest_Err_Add_Sts3	0x43
Selftest_Err_Add_Sts4	0x44
Selftest_Err_Add_Sts5	0x45
Selftest_Err_Add_Sts6	0x46
Selftest_Err_Add_Sts7	0x47
Selftest_Err_Add_Sts8	0x48
Selftest_Err_Add_Sts9	0x49
Selftest_Err_Add_Sts10	0x4A
Selftest_Err_Add_Sts11	0x4B
Selftest_Err_Add_Sts12	0x4C
Selftest_Err_Add_Sts13	0x4D
Selftest_Err_Add_Sts14	0x4E
Selftest_Err_Add_Sts15	0x4F

Table 6: RIMC registers list

Bits	Field Name	Comments	R/W	Reset value
7-5		Reserved	R	0
4-0	BITERR	Number of the bit in error-1. 0 for no error. This register is used only for EDAC testability, i.e. when TESTEN input pin is high.	R/W	0

Table 7: EDACConf_Reg register

Bits	Field Name	Comments	R/W	Reset value
15-0	CNT	Uncorrectable ECC counter	R	0

Table 8: ECC_DErr_Reg0 and ECC_DErr_Reg1

Bits	Field Name	Comments	R/W	Reset value
15-0	CNT	Correctable ECC counter	R	0

Table 9: ECC_SErr_Reg0 and ECC_DErr_Reg1

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	CLR	0: No effect 1: Clear ECC_DErr_Reg registers	W	0

Table 10: ECC_DErr_Clr register

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	CLR	0: No effect 1: Clear ECC_SErr_Reg registers	W	0

Table 11: ECC_SErr_Clr register

Bits	Field Name	Comments	R/W	Reset value
7-4	-	Reserved	R	0
3	DLL_RESET	0: Normal Mode 1: DLL is reset when refreshing DDR3 mode registers.	R/W	1
2-1	MODE	b00: Mode registers refreshed every 250 us b01: Mode registers refreshed every 1.024 seconds b10:No refresh of Mode registers b11: Reserved	R/W	0
0	BYP	0: Normal mode 1: Bypass mode. In this mode the only available user interface is the bypass interface	R/W	0

Table 12: Conf_Reg register

Bits	Field Name	Comments	R/W	Reset value
7-0	VAL	Byte value used for initialization of the whole memory array.	R/W	0

Table 13: Init_Pad_Reg register

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	STS	0: Padding is in IDLE 1: Padding is in progress	R	0

Table 14: Init_Pad_Sts register

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	START	0: No effect 1: Start the padding of the whole memory array	W	0

Table 15: Init_Pad_Ctl register

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	EN	0: Scrubbing disabled 1: Scrubbing Enabled	R/W	0

Table 16: ScrubEn_Reg register

Bits	Field Name	Comments	R/W	Reset value
15-0	FREQ	Register used to select the frequency of the scrubbing. 1 DDR3 Burst performed each FREQ SysClk clock cycles. *Note: This register shall have a value above or equal to 128.*	R/W	128

Table 17: Scrubbing_Reg0 and Scrubbing_Reg1

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	RST	0: No effect 1: Reset ScrubAd_Sts0 to ScrubAd_Sts3 registers	W	0

Table 18: Scrub_Ctl register

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	EN	Enable the corruption of data. Used to test the correct behavior of the ECC mechanism	R/W	0

Table 19: CorEn_Reg register

Bits	Field Name	Comments	R/W	Reset value
95-0	COR_BIT	Bit to corrupt (used only when CorEn_Reg register = 0x01)	R/W	0

Table 20: CorVec_Reg0 to CorVec_Reg11

Bits	Field Name	Comments	R/W	Reset value
7-3	-	Reserved	R	0
2-0	COR_BEAT	Beat to corrupt (used only when CorEn_Reg register = 0x01)	R/W	0

Table 21: CorBeat_Reg

Bits	Field Name	Comments	R/W	Reset value
31-0	AD	Address of the memory in which the scrubbing detected the 1st error	R	0

Table 22: ScrubAd_Sts0 to ScrubAd_Sts3

Bits	Field Name	Comments	R/W	Reset value
7-0	VERSION	Version of the RTL code	R	(1)

Table 23: Version_Sts

Bits	Field Name	Comments	R/W	Reset value
7-3	-	Reserved	W	
2	PHY_RST_CTL	1: Start phy reset of DDR3 components	W	
1	ZQ_LG_CTL	1: Start ZQ long calibration	W	
0	ZQ_SH_CTL	1: Start ZQ short calibration	W	

Table 24: Reset_ZQ_Ctl

Bits	Field Name	Comments	R/W	Reset value
7-3	-	Reserved	R	0
2	PHY_RST_STS	1: indicate when phy reset of DDR3 components is completed	R	0
1	ZQ_LG_STS	1: indicate ZQ long calibration is completed	R	0
0	ZQ_SH_STS	1: indicate ZQ short calibration is completed	R	0

Table 25: Reset_ZQ_Sts

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	W	
0	ST_SELF_TEST	1: start self test	W	

Table 26: Selftest_Ctl

Bits	Field Name	Comments	R/W	Reset value
31-0	START_ADD	Address to start the selftest	R/W	0

Table 27: Selftest_Start_Add_Reg0 to Selftest_Start_Add_Reg3

Bits	Field Name	Comments	R/W	Reset value
31-0	STOP_ADD	Address to stop the selftest	R/W	1

Table 28: Selftest_Stop_Add_Reg0 to Selftest_Stop_Add_Reg3

Bits	Field Name	Comments	R/W	Reset value
7-1	-	Reserved	R	0
0	STS_SELF_TEST	1: indicate self test is completed	R	0

Table 29: Selftest_Sts

Bits	Field Name	Comments	R/W	Reset value
31-0	ERR_CNT_STS	Indicate number of stuckbits detected	R	0

Table 30: Selftest_Err_Cnt_Sts0 to Selftest_Err_Cnt_Sts3

Bits	Field Name	Comments	R/W	Reset value
127-96	ERR_ADD3_STS	Report fourth address in error	R	0
95-64	ERR_ADD2_STS	Report third address in error	R	0
63-32	ERR_ADD1_STS	Report second address in error	R	0
31-0	ERR_ADD0_STS	Report first address in error	R	0

Table 31: Selftest_Err_Add_Sts0 to Selftest_Err_Add_Sts15

5 RIMC (MEMORY CONTROLLER + PHY)

The Physical layer IP (PHY) is dependant on the FPGA/ASIC technology being used. The following PHY are available for use inside the RIMC:

- Xilinx 7-series FPGA *
- Ultrascale FPGA **

* As of today not space qualified.
** Under Space qualification

N.B: Those FPGAs are validated targets running with our IP. For all information regarding FPGA Space qualification status, please contact directly FPGA supplier.

5.1 RIMC INTERFACES

The RIMC provides the following interfaces:

- The AMBA interfaces (already described in paragraph 4.3.1)
- The Bypass interface (already described in paragraph 4.3.2)
- The DDR3 memory interface
- The UPI interface

5.1.1 DDR3 INTERFACE

The RIMC is compliant to the JEDEC DDR3 memory interface specification.

5.1.2 UPI INTERFACE

An UPI interface is present at the user interface of the RIMC to allow a direct replacement of a MIG DDR controller from Xilinx with the RIMC.
A bridge is provided inside of the RIMC to convert the MIG into a standard AXI interface.

5.2 RIMC FOR 7 SERIES GENERICS/PARAMETERS AND PORTS

The RIMC contains the ports defined in Table 32.

The RIMC contains the generics defined in Table 33.

Name	Width	Direction	Functionality
ClkIn	1	I	Input clock used to generate other clock used by the RIMC. Used when SYSCLK_TYPE = “SINGLE_ENDED” or “NO_BUFFER”
ClkIn_p	1	I	Input clock used to generate other clock used by the RIMC. Used when SYSCLK_TYPE = “DIFFERENTIAL”
ClkIn_n	1	I	Input clock used to generate other clock used by the RIMC. Used when SYSCLK_TYPE = “DIFFERENTIAL”
idly_clk_200	1	I	Input clock at 200 MHz
Rst_N	1	I	Input asynchronous reset
Clk0_o	1	O	Output clock, to be used for all user interfaces
rst0_n_o	1	O	Output reset, synchronous to Clk0_o
TESTEN	1	I	When set the signal enable functionality to test the design
Clk_ref_user	1	I	Clk_ref generated by the user when GEN_CLK =0
Clk0_user	1	I	Clk0 generated by the user when GEN_CLK =0
Mem_refclk_user	1	I	Mem_refclk generated by the user when GEN_CLK =0
Freq_refclk_user	1	I	Freq_refclk generated by the user when GEN_CLK =0
Sync_pulse_user	1	I	Sync_pulse generated by the user when GEN_CLK =0
Pll_locked_user	1	I	Pll_locked generated by the user when GEN_CLK =0
Rst_user	1	I	Rst generated by the user when GEN_CLK =0
Rst_phaser_ref_user	1	I	Rst_phaser_ref generated by the user when GEN_CLK =0
Ref_dll_lock_user	1	O	Ref_dll_lock output to the user when GEN_CLK =0
SCRUB_EN	1	I	0: Scrubbing is disabled (or enabled through the ScrubEn_Reg register) 1: Scrubbing is enabled
REF_MODE	1	I	0: Normal refresh

Name	Width	Direction	Functionality
			1: Refresh timing is divided by 2
Ahbsi	NB_AHB_SLV records	I	AHB slave input array of record type
Ahbso	NB_AHB_SLV records	O	AHB slave output array of record type
Apbi	1 record	I	APB slave input
Apbo	1 record	O	APB slave output
AXI interface			
AXI_aw_out	NB_AXI_SLV records	I	Outputs relative to Address Write Channel
AXI_aw_in	NB_AXI_SLV records	O	Inputs relative to Address Write Channel
AXI_w_out	NB_AXI_SLV records	I	Outputs relative to Write Data Channel
AXI_w_in	NB_AXI_SLV records	O	Inputs relative to Write Data Channel
AXI_b_out	NB_AXI_SLV records	O	Outputs relative to Write Response Channel
AXI_b_in	NB_AXI_SLV records	I	Inputs relative to Write Response Channel
AXI_ar_out	NB_AXI_SLV records	I	Outputs relative to Address Read Channel
AXI_ar_in	NB_AXI_SLV records	O	Inputs relative to Address Read Channel
AXI_r_out	NB_AXI_SLV records	O	Outputs relative to Read Data Channel
AXI_r_in	NB_AXI_SLV records	I	Inputs relative to Read Data Channel
DDR3 interface			
ddr_ck	CLK_WIDTH	O	DDR clock
ddr_ck_n	CLK_WIDTH	O	DDR negative clock
ddr_addr	ROW_WIDTH	O	DDR address bus
ddr_ba	BANK_WIDTH	O	DDR bank bus
ddr_cas_n	1	O	DDR column address strobe bus
ddr_cke	CKE_WIDTH	O	DDR clock enable bus
ddr_cs_n	CS_WIDTH	O	DDR chip select bus
ddr_odt	ODT_WIDTH	O	DDR on-die-termination control bus

Name	Width	Direction	Functionality
ddr_ras_n	1	O	DDR row address strobe bus
ddr_rst_n	1	O	DDR reset bus.
ddr_dm	DM_WIDTH	O	DDR data mask
ddr_dq	DQ_WIDTH	IO	DDR data
ddr_dqs	DQS_WIDTH	IO	DDR data strobe
ddr_dqs_n	DQS_WIDTH	IO	DDR negative data strobe
ddr_we_n	1	O	DDR write enable
UPI Interface			
app_cmd	3	I	b000: Write b001: Read Others: Reserved
app_addr	31	I	Gives information about the address of the memory location to be accessed. This bus contains the bank address, the row address and the column address
app_en	1	I	Indicate that the command on signal app_cmd and app_addr is valid
app_rdy	1	O	Indicate that RIMC accept the command
app_wdf_data	2*nCK_PER_CLK*DATA_WIDTH	I	User input data
app_wdf_mask_data	2*nCK_PER_CLK*DATA_WIDTH /8	I	User mask data
app_wdf_wren	1	I	Write data to be written in memory
app_wdf_end	1	I	End of access on UPI interface
app_wdf_rdy	1	O	Indicate that RIMC accept the data to be written in memory
app_rd_data_valid	1	O	Status signal indicating read data is valid on the app_rd_data bus
app_rd_data_end	1	O	Status signal indicating end of read data access
app_rd_data	2*nCK_PER_CLK*DATA_WIDTH	O	Read data from the memory
Bypass interface			
user_ras_n	nCK_PER_CLK	I	Row Address Select
user_cas_n	nCK_PER_CLK	I	Column Address Select
user_we_n	nCK_PER_CLK	I	Write Enable, active LOW

Name	Width	Direction	Functionality
user_address	nCK_PER_CLK* ROW_WIDTH	I	Address bus
user_bank	nCK_PER_CLK* BANK_WIDTH	I	Bank Address bus
user_cs_n	nCK_PER_CLK* CS_NUM	I	Chip Select, active LOW
uer_cke	nCK_PER_CLK* CS_NUM	I	Clock Enable
user_odt	nCK_PER_CLK* CS_NUM	I	On Die Termination
user_reset_n	nCK_PER_CLK	I	Memory reset. Used only for DDR3
user_wdata_en	nCK_PER_CLK	I	Memory write data enable
user_dm	nCK_PER_CLK*2 *DQ_WIDTH/8	I	Memory data mask
user_dqo	nCK_PER_CLK*2 *DQ_WIDTH	I	Memory write data
user_rdata_en	nCK_PER_CLK	I	Memory read data Enable
user_dqi	nCK_PER_CLK*2 *DQ_WIDTH	O	Memory Read Data
user_dqiv	1	O	Memory Read data valid
rimc_dbg	record	O	Debug signals

Table 32: RIMC 7 series Core Ports

Name	Functionality	Possible values
GEN_UPI	Define if the UPI interface can be used by the user	0, 1
GEN_CLK	Define if clock are generated inside the RIMC or provided by the user	0, 1
SYSCLK_TYPE	Define if a clock buffer shall be added to the SysClk input clock	“SINGLE_ENDED”, “NO_BUFFER”
REFCLK_TYPE	Define if a clock buffer shall be added to the idly_clk_200 input clock	“SINGLE_ENDED”, “NO_BUFFER”
BANK_TYPE	Used only for Xilinx 7-series PHY.	“HP_LP”, “HR_LP”, “DEFAULT”
DATA_IO_PRIM_TYPE	Used only for Xilinx 7-series PHY.	“HP_LP”, “HR_LP”, “DEFAULT”
HDMAX	Data width of AHB and AXI busses	16, 32, 64, 128, 256

Name	Functionality	Possible values
GSYNCRST	Type of internal reset	0: Asynchronous reset 1: Synchronous reset
USE_DFF	Type of internal memories used for the FIFO functions	0 : Internal flip-flops 1 : Embedded RAM
ENDIANNESS	Endianness of AHB, APB and AXI busses	0: Little endian 1: Big endian
NB_AHB_SLV	Number of slave AHB busses.	1 to 8. *Note: NB_AHB_SLV+NB_AXI_SLV shall be between 0 and 8* *If the sum is null, then the MIG interface is used.*
NB_AXI_SLV	Number of slave AXI busses	1 to 8. *Note: NB_AHB_SLV+NB_AXI_SLV shall be between 0 and 8* *If the sum is null, then the MIG interface is used.*
SIMU	Activated only for simulation. Used to accelerate calibration.	0, 1
phy_config	Used only for Xilinx 7-series PHY	-
RTT_NOM	Nominal ODT termination value	"75"
RTT_WR	Dynamic ODT	"OFF", "60", "120"
AL	Additive Latency option	"0", "1", "2", "3", "4", "CL-1", "CL-2"
OUTPUT_DRV	Memory output drive. For DDR3, “HIGH” is 34 Ohm and “LOW” is 40 Ohm	“HIGH”, “LOW”
nCK_PER_CLK	Clock ratio between DFIClk and memory clock	1, 2, 4
CLKIN_PERIOD	SysClk clock period, in picosecond	
tCK	ddr3_ck clock period, in picosecond	
TWO_T_TIME_EN	Mode 2T	0, 1
DM_WIDTH	Data mask width	1, 2, 3, 4, 5, 6, 8, 9, 12
DQS_WIDTH	Data Strobe width	1, 2, 3, 4, 5, 6, 8, 9, 12
DATA_WIDTH	Full data width. See 4.2.1 for details.	8, 16, 32, 64

Name	Functionality	Possible values
ECC_WIDTH	Ful ECC width. See 4.2.1 for details.	5, 6, 8, 16
DQ_WIDTH	Full memory width, including DATA and ECC DQ_WIDTH shall be DATA_WIDTH+ECC_WIDTH, rounded up to the next multiple of 8	8, 16, 24, 32, 40, 48, 64, 72, 96
Memory generics		
BURST_LEN	Memory burst length	4, 8
CS_WIDTH	Width of memory chip select bus	1 to 4
CKE_WIDTH	Width of memory clock enable bus	1 to 12
ODT_WIDTH	Width of memory ODT bus	1 to 12
CS_NUM	Number of ranks	1, 2, 4
ODT_NUM	Number of rank for ODT	1, 2, 4
BANK_WIDTH	Number of banks	2, 3
ROW_WIDTH	Number of rows	14, 15, 16
COL_WIDTH	Number of columns	10
CLK_WIDTH	Number of clock outputs	1, 2, 3, 4, 5, 6, 7, 8, 9, 12
TWR_PS	Write recovery time, in picosecond	Depending on the DDR component datasheet
TWTR_PS	Write to read timing, in ps	Depending on the DDR component datasheet
TRTP_PS	Read to Precharge timing, in ps	Depending on the DDR component datasheet
TRFC_PS	Refresh to active or refresh to refresh command interval	Depending on the DDR component datasheet
CAS_LAT	Read CAS latency	According to the ddr3 clock frequency
CWL_LAT	Write CAS latency	According to the ddr3 clock frequency

Table 33: RIMC 7series generics list

5.1 RIMC for Ultrascale generics/parameters and ports

The RIMC contains the ports defined in Table 34.

The RIMC contains the generics defined in Table 35.

Name	Width	Direction	Functionality
ClkIn	1	I	Input clock used to generate other clock used by the RIMC. Used when SYSCLK_TYPE = “SINGLE_ENDED” or “NO_BUFFER”
ClkIn_p	1	I	Input clock used to generate other clock used by the RIMC. Used when SYSCLK_TYPE = “DIFFERENTIAL”
ClkIn_n	1	I	Input clock used to generate other clock used by the RIMC. Used when SYSCLK_TYPE = “DIFFERENTIAL”
Rst_N	1	I	Input asynchronous reset
Clk0_o	1	O	Output clock, to be used for all user interfaces
rst0_n_o	1	O	Output reset, synchronous to Clk0_o
TESTEN	1	I	When set the signal enable functionality to test the design
Mmcm_lock_user	1	I	Mmcm_lock generated by the user when GEN_CLK =0
Div_clk_user	1	I	Div_clk generated by the user when GEN_CLK =0
Riu_clk_user	1	I	Riu_clk generated by the user when GEN_CLK =0
Div_clk_rst_user	1	I	Div_clk_rst generated by the user when GEN_CLK =0
Riu_clk_rst_user	1	I	Riu_clk_rst generated by the user when GEN_CLK =0
Reset_ub_user	1	I	Reset_ub generated by the user when GEN_CLK =0
PllGate_user	1	I	PllGate generated by the user when GEN_CLK =0
Sys_clk_in_user	1	O	Sys_clk_in output to the user when GEN_CLK =0
Mmcm_clk_in_user	1	O	Mmcm_clk_in output to the user when GEN_CLK =0
Pll_lock_user	1	O	Pll_lock output to the user when GEN_CLK =0
SCRUB_EN	1	I	0: Scrubbing is disabled (or enabled through the ScrubEn_Reg register) 1: Scrubbing is enabled

Name	Width	Direction	Functionality
REF_MODE	1	I	0: Normal refresh 1: Refresh timing is divided by 2
Ahbsi	NB_AHB_SLV records	I	AHB slave input array of record type
Ahbso	NB_AHB_SLV records	O	AHB slave output array of record type
Apbi	1 record	I	APB slave input
Apbo	1 record	O	APB slave output
AXI interface			
AXI_aw_out	NB_AXI_SLV records	I	Outputs relative to Address Write Channel
AXI_aw_in	NB_AXI_SLV records	O	Inputs relative to Address Write Channel
AXI_w_out	NB_AXI_SLV records	I	Outputs relative to Write Data Channel
AXI_w_in	NB_AXI_SLV records	O	Inputs relative to Write Data Channel
AXI_b_out	NB_AXI_SLV records	O	Outputs relative to Write Response Channel
AXI_b_in	NB_AXI_SLV records	I	Inputs relative to Write Response Channel
AXI_ar_out	NB_AXI_SLV records	I	Outputs relative to Address Read Channel
AXI_ar_in	NB_AXI_SLV records	O	Inputs relative to Address Read Channel
AXI_r_out	NB_AXI_SLV records	O	Outputs relative to Read Data Channel
AXI_r_in	NB_AXI_SLV records	I	Inputs relative to Read Data Channel
DDR3 interface			
ddr_ck	CLK_WIDTH	O	DDR clock
ddr_ck_n	CLK_WIDTH	O	DDR negative clock
ddr_addr	ROW_WIDTH	O	DDR address bus
ddr_ba	BANK_WIDTH	O	DDR bank bus
ddr_cas_n	1	O	DDR column address strobe bus
ddr_cke	CKE_WIDTH	O	DDR clock enable bus
ddr_cs_n	CS_WIDTH	O	DDR chip select bus
ddr_odt	ODT_WIDTH	O	DDR on-die-termination

Name	Width	Direction	Functionality
			control bus
ddr_ras_n	1	O	DDR row address strobe bus
ddr_rst_n	1	O	DDR reset bus.
ddr_dm	DM_WIDTH	O	DDR data mask
ddr_dq	DQ_WIDTH	IO	DDR data
ddr_dqs	DQS_WIDTH	IO	DDR data strobe
ddr_dqs_n	DQS_WIDTH	IO	DDR negative data strobe
ddr_we_n	1	O	DDR write enable
UPI Interface			
app_cmd	3	I	b000: Write b001: Read Others: Reserved
app_addr	31	I	Gives information about the address of the memory location to be accessed. This bus contains the bank address, the row address and the column address
app_en	1	I	Indicate that the command on signal app_cmd and app_addr is valid
app_rdy	1	O	Indicate that RIMC accept the command
app_wdf_data	2*nCK_PER_CLK*DATA_WIDTH	I	User input data
app_wdf_mask_data	2*nCK_PER_CLK*DATA_WIDTH /8	I	User mask data
app_wdf_wren	1	I	Write data to be written in memory
app_wdf_end	1	I	End of access on UPI interface
app_wdf_rdy	1	O	Indicate that RIMC accept the data to be written in memory
app_rd_data_valid	1	O	Status signal indicating read data is valid on the app_rd_data bus
app_rd_data_end	1	O	Status signal indicating end of read data access
app_rd_data	2*nCK_PER_CLK*DATA_WIDTH	O	Read data from the memory
Bypass interface			
user_ras_n	nCK_PER_CLK	I	Row Address Select
user_cas_n	nCK_PER_CLK	I	Column Address Select

Name	Width	Direction	Functionality
user_we_n	nCK_PER_CLK	I	Write Enable, active LOW
user_address	nCK_PER_CLK* ROW_WIDTH	I	Address bus
user_bank	nCK_PER_CLK* BANK_WIDTH	I	Bank Address bus
user_cs_n	nCK_PER_CLK* CS_NUM	I	Chip Select, active LOW
uer_cke	nCK_PER_CLK* CS_NUM	I	Clock Enable
user_odt	nCK_PER_CLK* CS_NUM	I	On Die Termination
user_reset_n	nCK_PER_CLK	I	Memory reset. Used only for DDR3
user_wdata_en	nCK_PER_CLK	I	Memory write data enable
user_dm	nCK_PER_CLK*2 *DQ_WIDTH/8	I	Memory data mask
user_dqo	nCK_PER_CLK*2 *DQ_WIDTH	I	Memory write data
user_rdata_en	nCK_PER_CLK	I	Memory read data Enable
user_dqi	nCK_PER_CLK*2 *DQ_WIDTH	O	Memory Read Data
user_dqiv	1	O	Memory Read data valid
rimc_dbg	record	O	Debug signals

Table 34: RIMC Ultrascale Core Ports

Name	Functionality	Possible values
GEN_UPI	Define if the UPI interface can be used by the user	0, 1
GEN_CLK	Define if clock are generated inside the RIMC or provided by the user	0, 1
SYSCLK_TYPE	Define if a clock buffer shall be added to the SysClk input clock	“SINGLE_ENDED”, “NO_BUFFER”
BANK_TYPE	Used only for Xilinx 7-series PHY.	“HP_IO”, “HR_IO”
HDMAX	Data width of AHB and AXI busses	16, 32, 64, 128, 256

Name	Functionality	Possible values
GSYNCRST	Type of internal reset	0: Asynchronous reset 1: Synchronous reset
Type of internal memories used for the FIFO functions	0 : Internal flip-flops 1 : Embedded RAM	Type of internal memories used for the FIFO functions
ENDIANNESS	Endianness of AHB, APB and AXI busses	0: Little endian 1: Big endian
NB_AHB_SLV	Number of slave AHB busses.	0 to 8. *Note: NB_AHB_SLV+NB_AXI_SLV shall be between 0 and 8* *If the sum is null, then the MIG interface is used.*
NB_AXI_SLV	Number of slave AXI busses	0 to 8. *Note: NB_AHB_SLV+NB_AXI_SLV shall be between 0 and 8* *If the sum is null, then the MIG interface is used.*
SIMU	Activated only for simulation. Used to accelerate calibration.	0, 1
phy_config	Used only for Xilinx 7-series PHY	-
RTT_NOM	Nominal ODT termination value	"75"
RTT_WR		"OFF", "60", "120"
AL	Additive Latency option	"0", "1", "2", "3", "4", "CL-1", "CL-2"
OUTPUT_DRV	Memory output drive. For DDR3, “HIGH” is 34 Ohm and “LOW” is 40 Ohm	“HIGH”, “LOW”
CLKIN_PERIOD	SysClk clock period, in picosecond	
tCK	ddr3_ck clock period, in picosecond	
TWO_T_TIME_EN	Mode 2T	0, 1
DM_WIDTH	Data mask width	1, 2, 3, 4, 5, 6, 8, 9, 12
DQS_WIDTH	Data Strobe width	1, 2, 3, 4, 5, 6, 8, 9, 12
DATA_WIDTH	Full data width. See 4.2.1 for details.	8, 16, 32, 64
ECC_WIDTH	Ful ECC width. See 4.2.1 for details.	5, 6, 8, 16

Name	Functionality	Possible values
DQ_WIDTH	Full memory width, including DATA and ECC DQ_WIDTH shall be DATA_WIDTH+ECC_WIDTH, rounded up to the next multiple of 8	8, 16, 24, 32, 40, 48, 64, 72, 96
Memory generics		
CS_WIDTH	Width of memory chip select bus	1 to 4
CKE_WIDTH	Width of memory clock enable bus	1 to 12
ODT_WIDTH	Width of memory ODT bus	1 to 12
CS_NUM	Number of ranks	1, 2, 4
ODT_NUM	Number of rank for ODT	1, 2, 4
BANK_WIDTH	Number of banks	2, 3
ROW_WIDTH	Number of rows	14, 15, 16
COL_WIDTH	Number of columns	10
CLK_WIDTH	Number of clock outputs	1, 2, 3, 4, 5, 6, 7, 8, 9, 12
TWR_PS	Write recovery time, in picosecond	Depending on the DDR component datasheet
TWTR_PS	Write to read timing, in ps	Depending on the DDR component datasheet
TRTP_PS	Read to Precharge timing, in ps	Depending on the DDR component datasheet
TRFC_PS	Refresh to active or refresh to refresh command interval	Depending on the DDR component datasheet
CAS_LAT	Read CAS latency	According to the ddr3 clock frequency
CWL_LAT	Write CAS latency	According to the ddr3 clock frequency

Table 35: RIMC Ultrascale generics list

6 RIMC PERFORMANCES

6.1 MAXIMUM FREQUENCY

	Memory Controller	Physical Layer
Xilinx Kintex 7	200 MHz	800 MHz
Xilinx Ultrascale	200 MHz	800 MHz

Table 36 : Frequency result of RIMC

6.2 IP SIZE

The following sizes are given for the complete RIMC IP composed of both the Memory Controller and the PHY for a given FPGA target.

Table 37 gives the estimated sizes for different FPGA platforms based on the following Hypothesis :

- DDR3 72 bits (64 data + 8 ECC)
- 2 user AMBA interfaces, each one supporting 512 bits
- nCK_PER_CLK = 2 for Kintex 7

The precise IP Size depends on the configuration.

FPGA	Slice registers	Slice LUTs
Xilinx Kintex 7	16847	15069
Xilinx Ultrascale	27361	21296

Table 37: Xilinx resource usage

7 DELIVERABLES

The RIMC IP is delivered with the following directory structure:

- 3DIPMC0744-1:
  - Tech_independant: This directory contains all files for the Memory Controller IP.
    - source: This directory contains all of the VHDL file.
      - Memory_ctrl: This directory contains all of the VHDL files of the Memory Controller core
  - Tech_dependant: This directory contains all source files specific to a technology.
    - Ultrascale This directory contains all files specific to the Ultrascale
      - Source contain all hdl source files
        - rimc: This directory contains all of the VHDL files of the Ultrascale RIMC
        - ddr3_phy: This directory contains all of the VHDL files Ultrascale physical layer
    - V7_series This directory contains all files specific to the 7 serie
      - Source contain all hdl source files
        - rimc: This directory contains all of the VHDL files of the 7serie RIMC
        - ddr3_phy: This directory contains all of the VHDL files 7 serie physical layer
  - doc:
    - Contains the datasheet of the RIMC DDR3
    - Contains the RIMC DDR3 User Guide

8 TARGET COMPATIBILITY

The DDR3 memory controller IP core is delivered as high level VHDL source code. The memory controller is technology independent HDL code and can be implemented on any targeted technology. The IP provided physical layer and connection between memory controller and physical layer for following targets: the kintex 7 and Ultrascale.

9 PART NUMBER / ORDER INFORMATION

3DIPMC0744-1

Main Sales Office

FRANCE	3D PLUS 408, rue Hélène Boucher ZI. 78532 BUC Cedex	Tel : 33 (0)1 30 83 26 50	Fax : 33 (0)1 39 56 25 89	Web : www.3d-plus.com e-mail : sales@3d-plus.com	**DISTRIBUTOR**
USA	3D PLUS USA, Inc 151 Callan Avenue, Suite 310 San Leandro, CA 94577	Tel : (510) 824-5591	Tel : (510) 824-5591	e-mail : sales@3d-plus.com	

10 REVISION HISTORY

Rev. 1,November 2018

- Initial Datasheet release Xilinx, series 7

Rev. 2, April 2019

- Datasheet update with KU060 FPGA.

Rev. 3, August 2022

- Datasheet update with version 2 of the IP.