RTG4 FPGA DDR Memory Controller User Guide

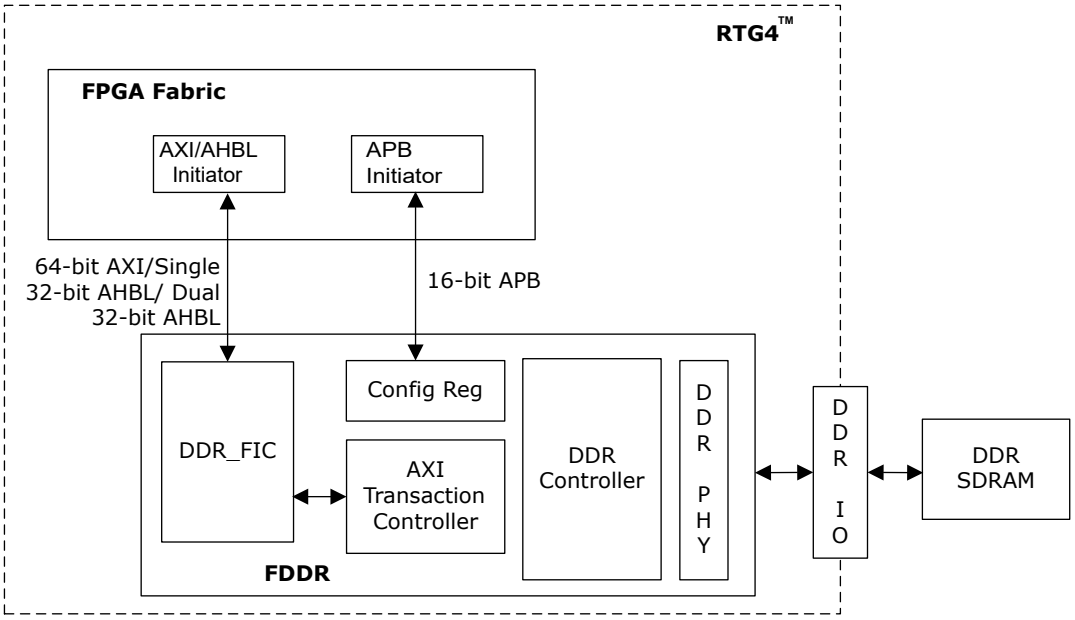
Introduction (Ask a Question)

The Fabric DDR (FDDR) subsystem is a hardened Application Specific Integrated Circuit (ASIC) block for interfacing the DDR2, DDR3, and LPDDR1 memories. The RTG4™ devices have two FDDR blocks. The FDDR subsystem is used to access the DDR memories for high-speed data transfers. It includes the DDR memory controller, DDR PHY, and arbitration logic to support multiple initiators. DDR controller is SEU mitigated and you must not have any performance impact due to SEU events.

The FPGA fabric initiators communicate with the DDR memories interfaced to the FDDR subsystem through the AXI or AHB interfaces. The FDDR subsystem has an APB target interface to configure the FDDR from the FPGA fabric initiator.

The following figure shows the FDDR subsystem.

Figure 1. System-Level FDDR Block Diagram



The FDDR subsystem accepts data transfer requests from AXI or AHBL interfaces. Any read or write transactions to the DDR memories can occur from the AXI or AHBL initiators in the FPGA Fabric through the DDR_FIC interface. The FDDR subsystem can also automatically perform DDR initialization, refresh, and ZQ calibration functions.

Important:

- Libero® SoC keeps the unused FDDR subsystems in Reset mode by driving the FDDR Reset signal to 0. The Netlist Viewer in Libero SoC shows the FDDR_POWER_DOWN buffers, which keep the unused FDDR blocks in Reset mode.
- The maximum DDR3 data-rate supported by RTG4 FDDR is 333 MHz or 667 Mbps. Therefore, Write Leveling is not mandatory and the interface does work if the board layout includes length matching and follows RTG4 board-level guidelines (see 1.17. Printed Circuit Board Guidelines). For Read Leveling, Libero SoC auto-generates predefined static delay ratios for FDDR initialization. These delay values are sufficient if the board layout follows the RTG4 board-level guidelines.

Features (Ask a Question)

The following are the features of the FDDR memory controller:

- Integrated on-chip DDR memory controller and PHY
- Supports LPDDR1, DDR2, and DDR3 memory devices
- Upto 667 Mbps (333 MHz DDR) performance
- Supports memory densities up to 2 GB
- Supports 8, 16, or 32 -bits data bus width modes
- Supports a maximum of 8 memory banks
- Supports a single rank of memory
- Enable or disable Single Error Correction and Double Error Detection (SECDED)
- Supports DRAM burst lengths of 4, 8, or 16, depending on the Bus width mode and the DDR type configured
- Supports sequential and interleaved burst ordering
- Programs internal control of ZQ short calibration cycles for DDR3 configurations
- Supports dynamic scheduling to optimize bandwidth and latency
- Supports self refresh entry and exit on command
- Supports deep power-down entry and exit on command
- Flexible address mapper logic that enables application specific mapping of row, column, bank, and rank bits
- Configurable support for 1T or 2T timing on the DDR SDRAM control signals
- Supports autonomous DRAM power-down entry and exit caused by lack of transaction arrival for programmable time

Memory Configurations (Ask a Question)

The RTG4 FDDR subsystem supports a wide range of common memory types, configurations, and densities, as listed in the following table. If SECDED mode is enabled in the FDDR controller, the external memory module must be connected to the following:

- Data lines, FDDR_DQ_ECC[3:0] when data width is 32 bits
- Data lines, FDDR_DQ_ECC[1:0] when data width is 16 bits
- Data line, FDDR_DQ_ECC[0] when data width is 8 bits

Table 1. Supported Memory (DDR2, DDR3, and LPDDR1) Configurations

Memory Depth	Width (in Bits)	Width (in Bits) (in SECDED Mode)	RTG4™ Devices
128M	32	36	Yes
	16	18	
	8	9	
256M	32	36	
	16	18	
	8	9	
512M	32	36	
	16	18	
	8	9	
1G	32	36	
	16	18	
	8	9	

Performance (Ask a Question)

The following table lists the maximum data rates supported by each data pin in the FDDR subsystem for the supported memory types. For more information, see the RTG4 FPGA Datasheet.

Table 2. DDR Speeds

Memory Type	Maximum Data Rate (Mbps)
LPDDR1	266 (133 MHz)
DDR2	66 (333 MHz)
DDR3	667 (333 MHz)

I/O Utilization (Ask a Question)

The following table lists the I/O utilization for RTG4 devices corresponding to the supported DDR bus widths. The remaining I/Os in the banks can be used for general purposes.

Important:

- If the FDDR subsystem is configured for LPDDR, one more I/O is also available for every 8 bits as the LPDDR does not have the FDDR_DQS_N pin.
- The AL40 and AE41 pins cannot be used with output register (OUT_REG) and enable register (EN_REG) of the I/O if FDDR_E is not used. The AL2 and AE1 pins cannot be used with OUT_REG or EN_REG if FDDR_W is not used.
- For general purpose usage of the unused I/Os of the FDDR bank, select one of the I/O standards with the voltage level same as that of the DDR I/Os.
- Self refresh must be disabled if the FDDR banks contain I/Os that are used for both DDR and for general purpose fabric I/Os. For more information, see 1.15.2. Self Refresh.

Table 3. I/O Utilization for RTG4 Devices

FDDR Bus Width	FDDR_E	FDDR_W
36 bits[1]	Bank 0 (85 + 1 pin)	Bank 9 (85+1 pin)
32 bits[1]	Bank 0 (76 + 1 pin)	Bank 9 (76+1 pin)
18 bits[2]	Bank 0 (59 + 5 pins)	Bank 9 (59 + 5 pins)
16 bits[2]	Bank 0 (53 + 5 pins)	Bank 9 (53 + 5 pins)

.....continued		
FDDR Bus Width	FDDR_E	FDDR_W
9 bits[2]	Bank 0 (47 + 5 pins)	Bank 9 (47 + 5 pins)
8 bits[2]	Bank 0 (41 + 5 pins)	Bank 9 (41 + 5 pins)

Important:

1. In 36 or 32 bit DDR bus width mode, an additional pin is reserved. For FDDR_E, the reserved pin is Bank 0- AK35; for FDDR_W, the reserved pin is bank9-AK7.
2. In 18, 16, 9, or 8 -bits DDR bus width mode, five additional pins are reserved. For FDDR_E, the reserved pins (Bank 0) are AK35, AJ31, AK32, AK33, AL35 and for FDDR_W, the reserved pins (Bank 9) are AK7, J11, AK9, AK10, and AL7.

I/O Standards (Ask a Question)

The following table lists the supported I/O standards for different DDR memories.

Table 4. Supported I/O Standards for Different DDR Memories

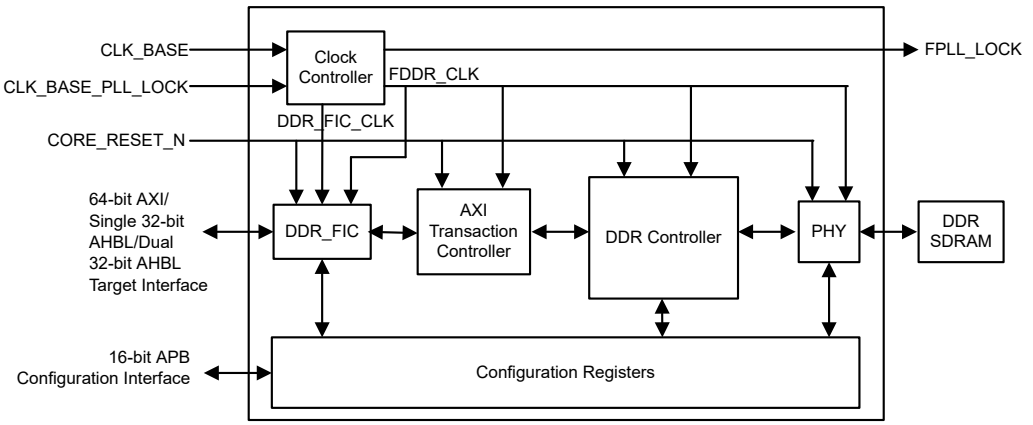
Memory Type	I/O Standards
DDR3	SSTL15I, SSTL15II
DDR2	SSTL18I, SSTL18II
LPDDR	LVCMOS18

Table of Contents

1. Functional Description (Ask a Question)

The following figure shows the functional block diagram of the FDDR subsystem. The main components include the clock controller, DDR fabric interface controller (DDR_FIC), AXI transaction controller, DDR memory controller, and DDR PHY.

Figure 1-1. FDDR Subsystem Functional Block Diagram



1.1 Clock Controller (Ask a Question)

The FDDR subsystem has a dedicated clock controller that generates aligned clocks to all the FDDR sub-blocks for proper operation and synchronous communication with the user logic in the FPGA fabric. The base clock (CLK_BASE) for the FDDR subsystem comes from a fabric Clock Conditioning Circuitry (CCC) or an external source through the FPGA fabric. The FDDR clock controller is associated with a dedicated Fabric Phase-Locked Loop (FPLL) for clock synthesis and deskewing the internal DDR_FIC clock from the base clock.

The FDDR clock controller consists of an FPLL and Fabric Alignment Clock Controller (FACC).

1.1.1 FPLL (Ask a Question)

FPLL is a Radiation Tolerant PLL (RT-PLL) and it ensures that the clock outputs from PLL do not have glitches or have significant differences in jitter when exposed to radiation. The CLK_BASE signal from the FPGA fabric is used as a reference clock to the FPLL and is multiplied to generate a clock frequency of upto 333 MHz. The base clock can be generated from any one of the fabric CCCs or a clock source (internal or external) through global network.

The device core supply (VDD) powers the digital section, and the analog supply (FDDR_PLL_VDDA) powers the analog section. The required voltage for FDDR_PLL_VDDA is 3.3V, and this does not impact the FPLL frequency range. See the RTG4 FPGA Datasheet for the FPLL operational range and characteristics.

FPLL generates a lock signal (FPLL_LOCK) to indicate that is locked onto the CLK_BASE signal. The precision of FPLL_LOCK discrimination can be adjusted using the lock window controls. It represents the phase error window for lock assertion. The lock window can be adjusted between 500 parts per million (ppm) and 64,000 ppm in powers of 2. The integration of the lock period can be adjusted using a built-in lock counter. The lock counter or lock delay indicates the number of reference clock cycles to wait after FPLL is locked to assert the FPLL_LOCK signal. The lock delay is used to avoid false toggling of the FPLL lock signal. The lock counter can be configured between 32 and 1048576 cycles in powers of 2.

1.1.2 FACC (Ask a Question)

Within the FDDR clock controller, FACC is responsible for interfacing with FPLL, generating the aligned clocks required by the FDDR subsystem, and controlling the alignment of FPGA fabric interface clocks.

The clocks generated by FACC are:

- **FDDR_CLK**: Clocks the FDDR subsystem. FDDR_CLK can be operated up to 333 MHz, depending on the type of DDR present in the system.
- **DDR_FIC_CLK**: Clocks the DDR_FIC, and defines the frequency at which the connected FPGA Fabric subsystem operates.
- **FACC Divider**: Divides the high-speed clock coming from the FPLL to generate the DDR_ FIC clock according to the configured division ratios. The possible FDDR_CLK:DDR_FIC_CLK ratios are 1:1 to 16:1.

1.1.3 FPLL Initialization (Ask a Question)

To attain clock alignment between the FPGA fabric and the FDDR subsystem, FPLL must be used to perform deskewing of the FDDR clocks. After FPLL is initialized, it typically takes over 500 divided reference clock cycles for the lock to be achieved. The FPLL lock assertion time is also dependent on the FPLL lock parameters (lock window and lock delay). There is no provision made for operation of the FDDR subsystem except the APB interface, before the FPLL lock is achieved.

1.1.4 PLL Lock Monitoring (Ask a Question)

FDDR has an input, CLK_BASE_PLL_LOCK, to monitor the fabric PLL lock. It must be connected to the lock signal generated by the fabric PLL that is used to generate the base clock to the FDDR.

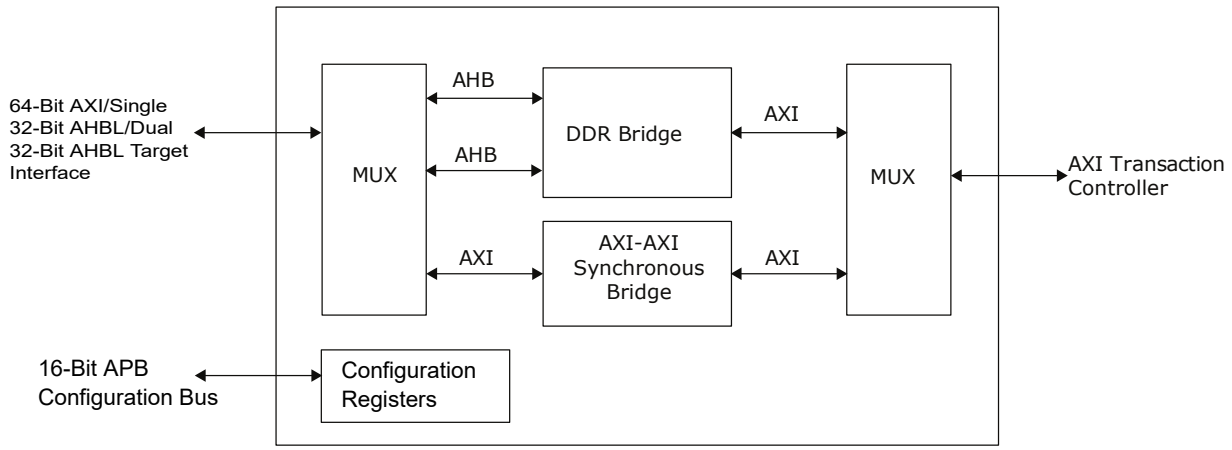
Within the FDDR subsystem, there are two interrupts related to the PLL lock. A lock interrupt, indicates that the FPLL lock is achieved, and further FPLL lock lost the interrupt. Each of these two interrupts has a corresponding interrupt enable bit in the FDDR subsystem registers. It is also possible to read the state of the two PLL lock signals from the FDDR registers.

In the event of loss of FPLL lock, even if the output is not exactly in phase lock with the reference, FPLL still generates a clock. User logic in the FPGA fabric can use the FPLL_LOCK signal to prevent communication with the FDDR subsystem during this time.

1.2 DDR_FIC (Ask a Question)

DDR_FIC facilitates communication between the FPGA fabric initiators and the AXI transaction controller. The following figure shows the DDR_FIC block diagram.

Figure 1-2. DDR_FIC Block Diagram



Fabric initiators can access the FDDR subsystem using the following:

- A single AXI-64 bit interface
- A single AHBL-32 bit interface
- Dual AHBL-32 bit interfaces

If the AXI-64 interface is selected, DDR_FIC acts as an AXI to AXI synchronous bridge. In this mode, the FPGA fabric initiator can perform locked transactions. For this purpose, a user configurable 20 bit down counter keeps track of the duration of the locked transfer. If the transfer is not completed before the down counter reaches zero, a single clock cycle pulse interrupt is generated from DDR_FIC to the fabric interface.

If single or dual AHBL-32 bit interface is selected, DDR_FIC converts the single or dual 32 bit AHBL initiator transactions from the FPGA fabric to the 64 bit AXI transactions. The DDR bridge, which is embedded as part of DDR_FIC, is enabled in this case. The DDR bridge has an arbiter that uses a round robin priority scheme on read and write requests from the two AHB initiators.

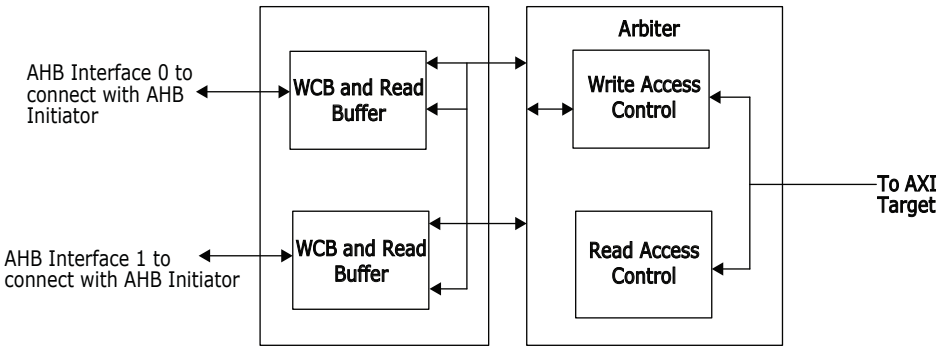
The DDR_FIC input interface is clocked by the FPGA fabric clock, and the AXI transaction controller is clocked by FDDR_CLK from the FDDR clock controller. Clock ratios between FDDR_CLK and DDR_FIC clocks can vary. Supported ratios are from 1:1 to 16:1. Clock ratios can be configured using the Libero SoC software or through the FDDR_FACC_DIVISOR_RATIO register. For more information, see Table 2-105.

1.3 DDR Bridge (Ask a Question)

The DDR bridge facilitates multiple AHB bus initiators to access a single AXI target and optimizes read and write operations from multiple AHB initiators to a single external DDR memory.

The DDR bridge consists of two main components: Read Combining Buffers (RCB) and Write Combining Buffers (WCB), and an arbiter, as shown in the following figure. The DDR bridge buffers the AHB write transactions into write combining buffers before bursting out to the external DDR memory. It also includes read buffers for AHB initiators to efficiently read data from the external DDR memory. All buffers within the DDR bridge are implemented with latches and so are not subject to Single Event Upsets (SEUs). The external DDR memory regions can be configured to be non-bufferable. If an initiator interface requests a write or read to a non-bufferable region, the DDR bridge is bypassed. The size of the non-bufferable address space can also be configured.

Figure 1-3. DDR Bridge Functional Block Diagram



1.3.1 Write Combining Buffer (Ask a Question)

WCB combines multiple write transactions from the AHB initiator into the AXI burst transactions. It has a user configurable burst size of 16 or 32 bytes. Each WCB maintains a base address tag that stores the base address of the data to be combined in the buffer.

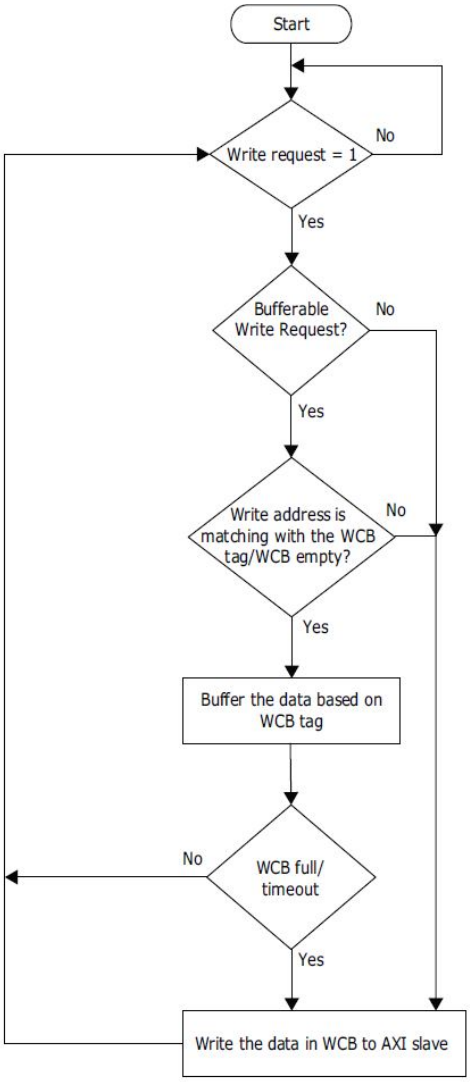
For each write transaction, the address is compared with the WCB tag. If the address matches the tag, data is combined into the buffer. WCB writes to the appropriate byte location based on the offset address of the data. It can also be disabled, if buffering is not required.

WCB has a 10 bit timer (down counter) that starts when the first bufferable write data is loaded into it. The timer starts decrementing at every positive edge of the AHB clock and when it reaches zero, the data in WCB is written to the AXI target.

WCB checks for any other initiator that has initiated a read operation to the same address for which data is already present in a write buffer or for which a write operation is in progress. If the address for a read request matches the write buffer tag, the read request is held until the buffer is written completely to the AXI target.

The following figure shows the flowchart for WCB operation.

Figure 1-4. WCB Operation

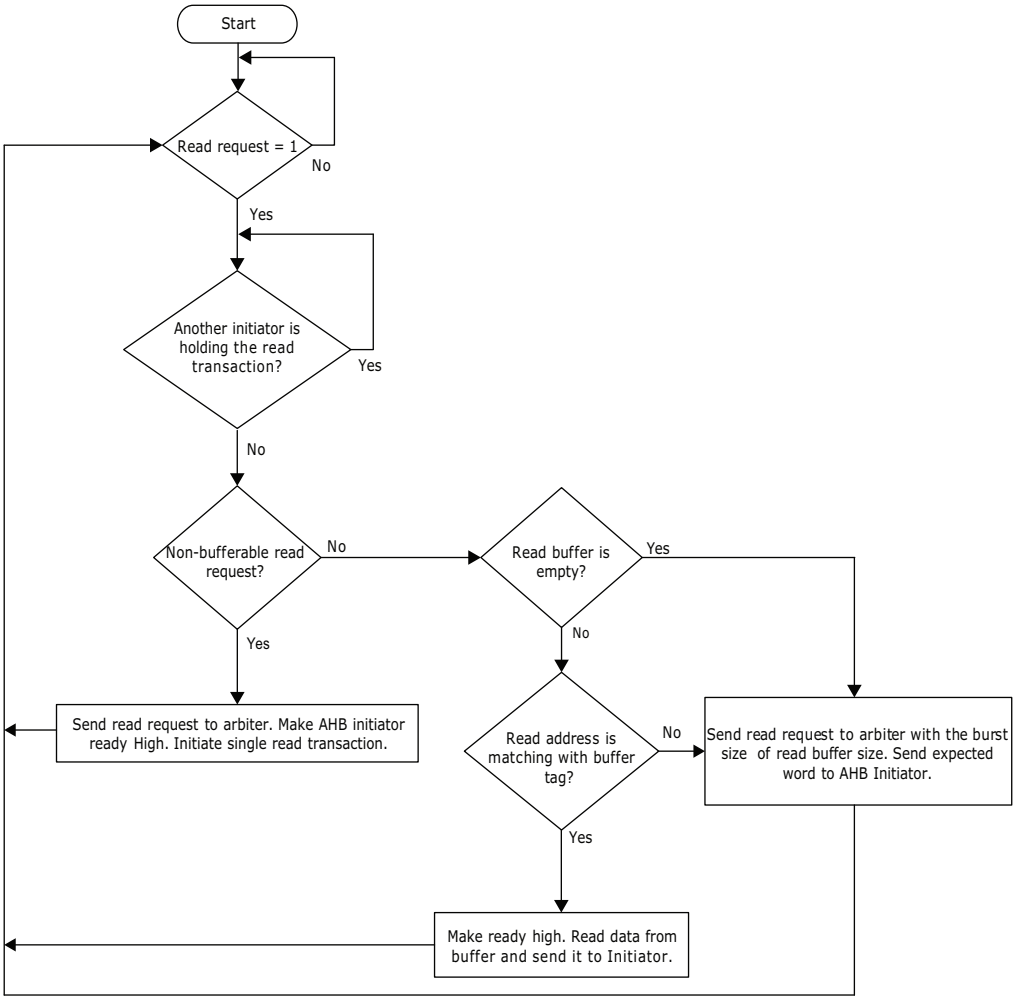


1.3.2 Read Buffer (Ask a Question)

The DDR bridge has a read buffer for each initiator to hold the fetched DDR burst data. Each read buffer has a configurable burst size of 16 or 32 bytes. The read buffer initiates a DDR burst size request for reads in the bufferable region, regardless of the size of request from the initiator. Each read buffer is associated with a specific initiator for reading. It does not check the read addresses of other initiators to determine whether that data can be read from the read buffer. There is no cross buffer read access.

The following figure shows the flowchart for read operation.

Figure 1-5. Read Operation



The read buffer is invalidated under the following conditions:

- If the address from the initiator is outside the tag region, the current data in the read buffer is invalidated.
- To ensure proper data coherency, all initiator's write addresses are tracked. If an address matches that of the read buffer tag, the read entry is invalidated.
- A non-bufferable or locked transaction is initiated by any initiator.
- An Invalidate command is issued.
- A buffer disable command is issued.
- An error response from DDR for the expected word read.

1.3.3 Arbiter (Ask a Question)

The DDR bridge arbiter includes two independent arbitration controllers for read and write requests.

1.3.3.1 Write Access Controller (Ask a Question)

The Write Access Controller (WAC) arbitrates write requests from the WCBs and grants access to one of the requesting initiators based on its priority. A round robin priority scheme is applied between

Initiator Interface 0 and Initiator Interface 1. All transactions from a single initiator have a dedicated initiator ID.

Once a burst transaction is initiated to the external DDR memory, the transactions are completed without an interruption. No other initiator, even a high priority initiator, can interrupt this process. Subsequent write requests from the same initiator are held until the previous write transactions are completed to the external DDR memory. Subsequent write requests from other initiators can be accepted and allowed to write into WCB, but the DDR bridge does not write this data until the previous write transactions are completed to the external DDR memory.

1.3.3.2 Read Access Controller (Ask a Question)

The Read Access Controller (RAC) arbitrates read requests from read buffers and grants access to one of the requesting initiators depending on its priority. A round robin priority scheme is applied between Initiator Interface 0 and Initiator Interface 1. The RAC also routes the read data from the AXI target (FDDR) to the corresponding initiator based on the Read data ID.

1.3.4 Locked Transactions (Ask a Question)

The DDR bridge initiators can initiate locked transfers by asserting the HMASTLOCK signal of the corresponding AHB interface. These locked transactions are initiated only after all the pending write and read transactions are completed.

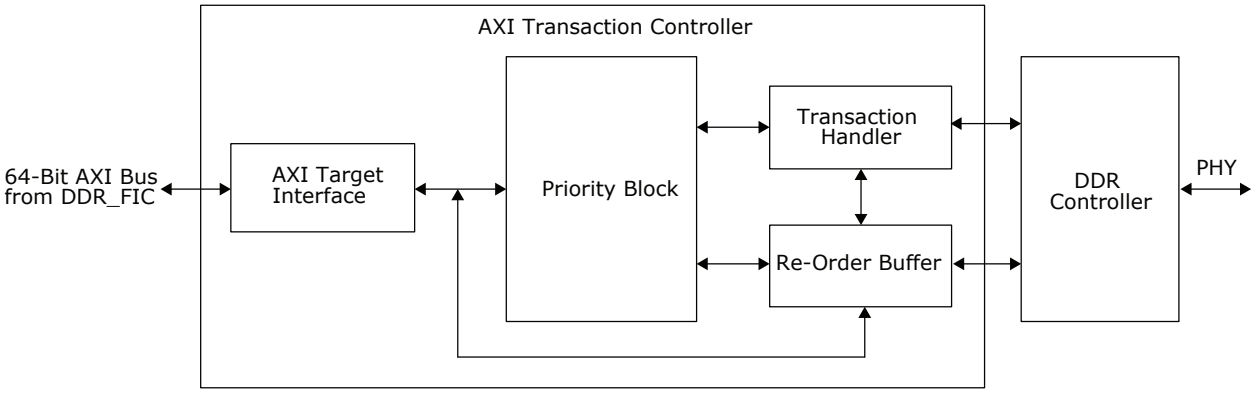
The arbiter has a 20 bit up counter to detect a lock time-out condition. The counter starts counting when a locked transaction is initiated on the bus. When the counter reaches its maximum value, an interrupt is generated. The interrupt can be cleared by setting the DDR_FIC_LCKTOUT bit field in the DDR_FIC_SW_HPB_LOCKOUT_SR register of the SYSREG block. If the interrupt is cleared and the lock signal is still asserted, the counter starts counting again. For more information, see Table 2-93.

The DDR bridge facilitates multiple AHB bus initiators to access a single AXI target and optimizes read and write operations from multiple AHB initiators to a single external DDR memory.

1.4 AXI Transaction Controller (Ask a Question)

The AXI transaction controller receives 64 bit AXI transactions from DDR_FIC and translates them into DDR controller transactions. The following figure shows the block diagram of the AXI transaction controller interfaced with the DDR controller.

Figure 1-6. AXI Transaction Controller Block Diagram



The AXI transaction controller has four major blocks:

- AXI target interface
- Priority block
- Transaction handler
- Re-order buffer

1.4.1 AXI Target Interfaces (Ask a Question)

The AXI transaction controller has a 64 bit AXI target interface from DDR_FIC. The AXI target port is 64 bits wide and is in compliance with the standard AXI protocol. Each transaction has an ID related to the initiator interface. Transactions with the same ID are completed in order while the transactions with different read IDs can be completed in any order depending on when the instruction is executed by the DDR controller. If a initiator requires ordering between the transactions, the same ID must be used.

The AXI target interface has individual read and write ports. The read port queues read AXI transactions and it can hold up to four read transactions. The write port handles only one write transaction at a time and generates the handshaking signals on the AXI interface.

1.4.2 Priority Block (Ask a Question)

The priority block prioritizes AXI read or write transactions and provides control to the transaction handler. AXI read transactions have a higher priority. The fabric initiator, using DDR_FIC can be programmed to have a higher priority by configuring the PRIORITY_ID and PRIORITY_ENABLE_BIT bit fields in the DDRC_AXI_FABRIC_PRI_ID_CR register. For more information, see Table 2-76.

1.4.3 Transaction Handler (Ask a Question)

The transaction handler converts AXI transactions into DDR controller commands. The transaction handler works on a single transaction at a time from the read or write port queue that is selected by the priority block. The transaction handler has a write command controller and read command controller for write and read transactions.

The write command controller fetches the command from the AXI target write port and sends a write instruction to the DDR controller. If SECDED is enabled, a Read Modified Write (RMW) instruction is sent to the DDR controller. The read command controller generates read transactions to the DDR controller.

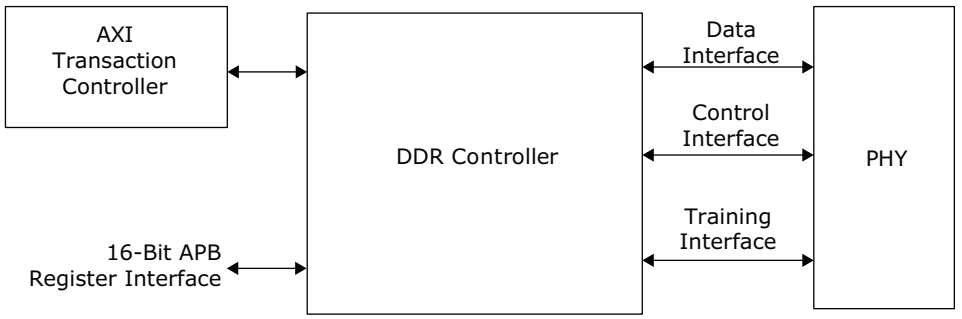
1.4.4 Re-Order Buffer (Ask a Question)

The re-order buffer receives data from the DDR controller and orders the data as requested by the AXI initiator when a single AXI transaction is split into multiple DDR controller transactions depending on the transfer size.

1.5 DDR Controller (Ask a Question)

The DDR controller receives requests from the AXI transaction controller, maps system addresses to DRAM addresses (rank, bank, row, and column), and prioritizes requests to minimize latency of reads (especially high priority reads) and maximize page hits. It also ensures that DRAM is properly initialized, all requests are made to DRAM legally (accounting for associated DRAM constraints), refreshes are inserted as required, and the DRAM enters, and exits various power-saving modes appropriately. The following figure shows the DDR controller connections in the FDDR subsystem.

Figure 1-7. DDR Controller Block Diagram



The following sections describe the key functions of the DDR controller:

- Address mapping
- Transaction scheduling
- Write combine
- SECDED
- 1T 2T timing
- On-Die Termination (ODT) controls
- Performance
- DRAM constraints

1.5.1 Address Mapping (Ask a Question)

Read and write requests to the DDR controller requires a system address. The DDR controller maps this system address with the rank, bank, row, and column addresses to the DRAM.

The address mapper maps linear request addresses to the DDR memory addresses by selecting the source bit that maps to all the applicable DDR memory address bits. The address map interface registers can be configured to map the source address bits to the DRAM address.

Each DDR memory address bit has an associated register vector to determine its source. The source address bit number is determined by adding the internal base of a given register to the programmed value for that register, as shown in the following equation.
[Internal base] + [register value] = [source address bit number]

For example, reading the description for REG_DDRC_ADDRMAP_COL_B3, the internal base is 3; so when the full data bus is in use, the column bit, 4 is determined by 3 + [register value]. For more information, see Table 2-10.

If this register is programmed to 2, then the source address bit is: 3 + 2 = 5.

The DDR Configurator assigns values to the address mapping registers depending on the selected number of columns, rows, and banks. The following figure provides the default mapping of the memory row, bank, and column address to the user interface address domain.

Figure 1-8. Address Mapping

Full bus width mode

AXI/AHB Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row mapping (DDR2/DDR3)		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
Row mapping (LPDDR)			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Bank mapping(DDR2/DDR3)																		2	1	0												
Bank mapping(LPDDR)																			1	0												
column mapping																					9	8	7	6	5	4	3	2	1	0		

Half bus width mode

AXI/AHB Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row mapping(DDR2/DDR3)			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Row mapping (LPDDR)				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
Bank mapping(DDR2/DDR3)																			2	1	0											
Bank mapping(LPDDR)																				1	0											
column mapping																						9	8	7	6	5	4	3	2	1	0	

Quarter bus width mode

AXI/AHB Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row mapping(DDR2/DDR3)				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
Row mapping (LPDDR)					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Bank mapping(DDR2/DDR3)																				2	1	0										
Bank mapping(LPDDR)																					1	0										
column mapping																							9	8	7	6	5	4	3	2	1	0

Following is the list of address mapping registers.

- DDRC_ADDR_MAP_BANK_CR
- DDRC_ADDR_MAP_COL_1_CR
- DDRC_ADDR_MAP_COL_2_CR
- DDRC_ADDR_MAP_COL_3_CR
- DDRC_ADDR_MAP_ROW_1_CR
- DDRC_ADDR_MAP_ROW_2_CR

While configuring the registers, ensure that two DDR memory address bits are not determined by the same source address bit.

Important:

- Some registers map to multiple source address bits (REG_DDRC_ADDRMAP_ROW_B0_11).
- For the DDR controller to arrive at the right address, FDDR maps the system address or AXI address bits [4:0].
  - In full bus width mode, the system address bits [4:0] are used to map the lower column address bits—C0, C1, and C2.
  - In half bus width mode, the system address bits [4:0] are used to map the lower column address bits—C0, C1, C2, and C3.
  - In quarter bus width mode, the system address bits [4:0] are used to map the lower column address bits—C0, C1, C2, C3, and C4.

The FDDR configurator uses {Row, Bank, Column} address mapping as shown in the following example.

1.5.1.1 Example: FDDR Configurator Address Map Configuration (Ask a Question)

In this example, the address map registers are configured to access a 512 MB DDR3 SDRAM memory, MT41J512M8RA, from the FDDR subsystem. 1.10.2. Example 2: Connecting 32 Bit DDR3 to FDDR_PADs with SECDED shows the connections between the 32-bit DDR3 memory to FDDR_PADs with SECDED. The 512M x 8 bit DDR3 memory module has 3 Bank Address lines, 16 rows, and 10 columns.

- The column address bits 3 to 9 are mapped from the system address bit[5] to system address bit[11]. To map the column 3 bit (C3) to address [5], the field is configured to 3, as the base value is 2. Similarly, the other column address bits are configured as follows:
  - DDRC_ADDR_MAP_COL_1_CR = 0x3333
  - DDRC_ADDR_MAP_COL_2_CR = 0x3FFF
  - DDRC_ADDR_MAP_COL_3_CR = 0x3300
- The bank address bit 0 to 2 are mapped to system address bit[12] to system address bit[14]. To map the bank bit 0 to address [12], the field is configured to A, as the base value is 2. Similarly, the other bank address bits are configured as follows:
  - DDRC_ADDR_MAP_BANK_CR = 0xAAA
- The row address bits 0 to 15 are mapped to system address bit[15] to system address bit[27]. To map the bank bit0 to address [15], the field is configured to 9, as the base value is 6. Similarly, the other bank address bits are configured as follows:
  - DDRC_ADDR_MAP_ROW_1_CR = 0x9999
  - DDRC_ADDR_MAP_ROW_2_CR = 0x9FF

Important: The FDDR subsystem can access the 2 GB address space (0x00000000 - 0x7FFFFFFF). But in this example, 512 MB (0x00000000 - 0x1FFFFFFF) DDR3 SDRAM is connected to the 16 address lines of FDDR. The memory visible in the other memory space is a mirror of this 512 MB memory.

1.5.2 Transaction Scheduling (Ask a Question)

The controller optimizes read and write requests to minimize the DDR read and write latencies using a 16-location Content Addressable Memory (CAM) that stores outstanding requests. The DDR controller schedules the read and write transactions to the DDR memory. Based on the commands from the AXI transaction controller, the transitions are classified in to three types:

- Low-Priority Reads (LPR)
- High-Priority Reads (HPR)
- Writes (WR)

Each type of transaction has a queue and the queued transactions can be in normal state or in critical state. The transactions in a queue moves from normal state to critical state when that transaction is not serviced for a count of MAX_STARVE_X32 clocks. The MAX_STARVE_X32 values for each queue can be configured using the DDR controller performance registers. For more information, see 1.9. Functional Timing Diagrams. The DDR controller completes the critical transactions with a high priority.

1.5.3 Write Combine (Ask a Question)

The DDR controller combines multiple writes to the same address into a single write to the DDR memory. When a new write encounters the queued write, the DDR controller overwrites

the queued write data with the new write data and performs only a single write transaction. To disable the write combine set of the register the bits from the REG_DDRC_DIS_WC field of the DDRC_ADDR_MAP_COL_3_CR register is set to 1. For more information, see Table 2-31.

1.5.4 SECDED (Ask a Question)

The DDR controller supports built-in SECDED capability for correcting single-bit errors and detecting two-bit errors. The SECDED feature can be enabled in the DDR memory controller configuration window. When SECDED is enabled, the DDR controller adds 8 bits of SECDED data to every 64 bits of data. Enabling SECDED might cause a reduction in throughput.

For 64-bit aligned addressing. the SECDED feature corrects all the errors only when memory selection parameters listed in Table 1-11 are followed. For an x32 interface, if the AXI interface width is 64 bit, the maximum supported burst length is four for correcting all the single bit errors and for detecting double bit errors.

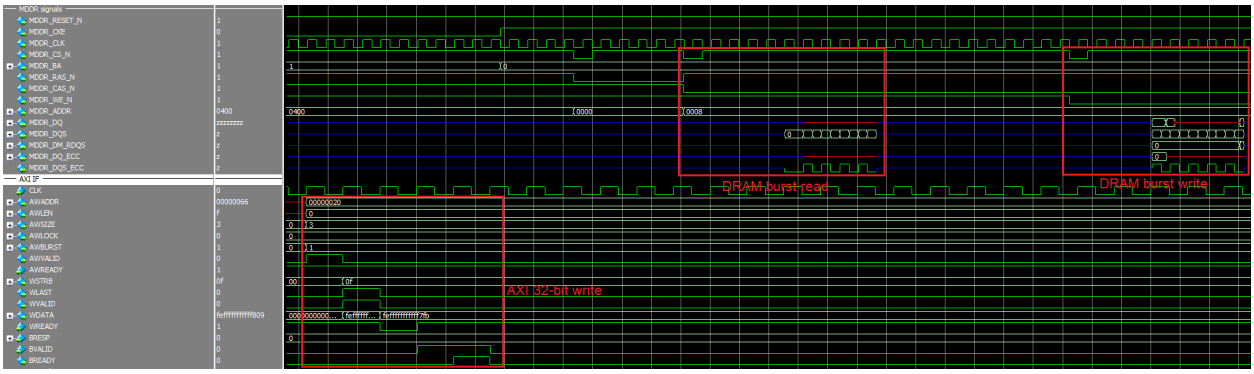
The DDR controller computes ECC for every 64 bit data. When SECDED is enabled, a write operation computes and stores a SECDED code with the data, and a read operation reads and checks the data against the stored SECDED code. It is, therefore, possible to receive single/dual bit errors when reading uninitialized memory locations. To avoid this, all the memory locations must be written before being read.

For a non 64-bit write operation the DDR controller performs 256 bit Read Modify Write (RMW) operation. This read modify write operation is always performed on 256 bit aligned addresses. For example, if DDR controller receives a 32 bit write operation to address 0x4, then the DDR controller performs the following operations:

- Reads the 256 bit data from 0x0 (256 bit aligned address for 0x4)
- Modifies 32 bits (bit 33 to bit 64) of that 256 bit data with the user's 32 bit data
- Computes the ECC and writes 288 bits (256 bit data + 32 bit ECC) to address 0x0

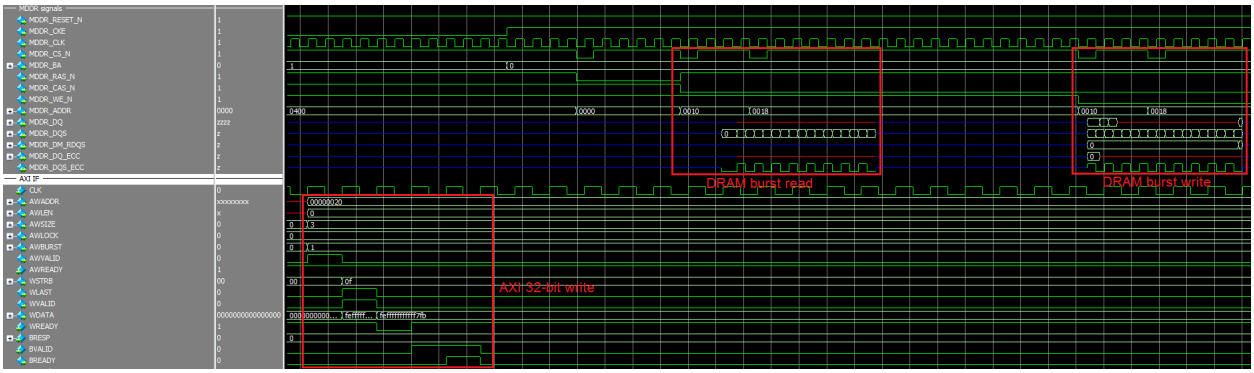
The following figure shows the DDR controller burst transactions to DRAM for unaligned 64 bit AXI write transaction. The DDR controller is configured for DDR3 memory, 32 bit burst width, and burst length 8.

Figure 1-9. DDR RMW Operation when DDR Bus Width is 32 bit and Burst Length is 8



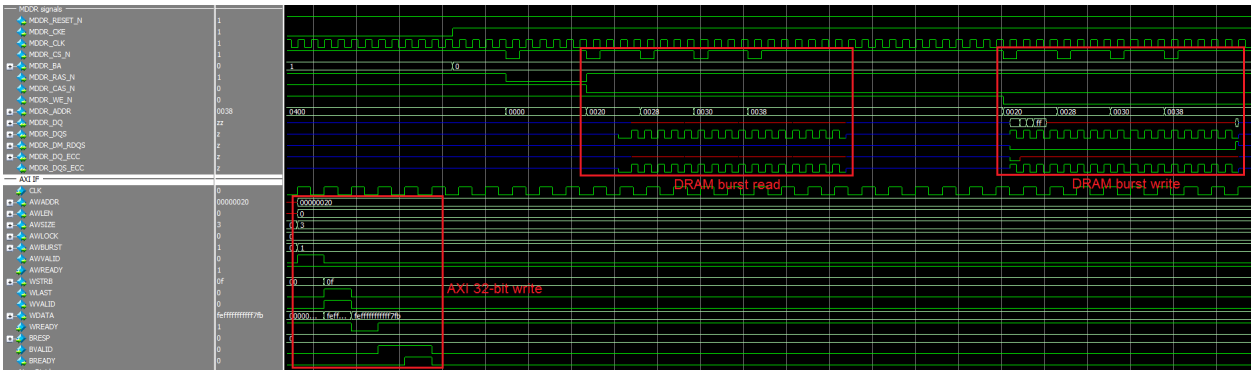
The following figure shows the DDR controller burst transactions to DRAM for unaligned 64 bit AXI write transaction. The DDR controller is configured for DDR3 memory, 16 bit bust width, and burst length 8.

Figure 1-10. DDR RMW Operation when DDR Bus Width is 16 bit and Burst Length is 8



The following figure shows the DDR controller burst transactions to DRAM for unaligned 64 bit AXI write transaction. The DDR controller is configured for DDR3 memory, 8 bit bust width, and burst length 8.

Figure 1-11. DDR RMW Operation when DDR Bus Width is 8 bit and Burst Length is 8



The SECDED bits are interlaced with the data bits. The following table lists the SECDED DQ lines at FDDR memory.

Table 1-1. SECDED DQ Lines at FDDR Memory

Mode	SECDED Data Pins
Full bus width mode	FDDR_DQ_ECC[3:0]
Half bus width mode	FDDR_DQ_ECC[1:0]
Quarter bus width mode	FDDR_DQ_ECC[0]

When the controller detects a correctable SECDED error, it does the following:

- Generates an interrupt signal which can be monitored by reading the interrupt status register, DDRC_ECC_INT_SR. The FDDR subsystem also generates the ECCINT interrupt signal that can be monitored from the FPGA fabric.
- Sends the corrected data to the read requested FPGA fabric initiator as part of the read data.
- Sends the SECDED error information to the DDRC_LCE_SYNDROME_1_SR register.
- Performs a read-modify-write operation to correct the data present in the DRAM.

When the controller detects an uncorrectable error, it does the following:

- Generates an interrupt signal which can be monitored by reading the interrupt status register, DDRC_ECC_INT_SR. The FDDR subsystem also generates the ECCINT interrupt signal that can be monitored from the FPGA fabric.
- Sends the data with error to the read requested FPGA fabric initiator as part of the read data.
- Sends the SECDED error information to the DDRC_LUE_SYNDROME_1_SR register.

The following SECDED registers can be monitored for identifying the exact location of an error in the DDR SDRAM:

- DDRC_LUE_ADDRESS_1_SR and DDRC_LUE_ADDRESS_2_SR give the row/bank/column information of the SECDED unrecoverable error.
- DDRC_LCE_ADDRESS_1_SR and DDRC_LCE_ADDRESS_2_SR give the row/bank/column information of the SECDED error correction.
- DDRC_LCB_NUMBER_SR indicates the location of the bit that caused the single-bit error in the SECDED case (encoded value).
- DDRC_ECC_INT_SR indicates whether the SECDED interrupt is because of a single-bit error or double-bit error. The SECDED interrupt and the ECC interrupt can be cleared by writing 1 to FDDR_INTERRUPT_SR (0x53C).

Important: Uncorrectable 2-bit errors trigger a HRESP error exception to the master (system/CPU/Mi-V). The HRESP error exception prevents the corrupt data from triggering an unexpected event or response at the master. Therefore, the master must implement a bus exception handler routine to recover or reset from a memory error.

1.5.5 1T or 2T Timing (Ask a Question)

To use the DRAM in 1T or 2T Timing modes, configure the DDRC_PERF_PARAM_3_CR register. To clock the address bus, use the 1T or 2T clocking. In 1T, the DDR controller can issue a new command on every clock cycle. In 2T timing, the DDR controller holds the address and command bus valid for two clock cycles. This reduces the efficiency of the bus to one command per two clocks, but it doubles the amount of setup and hold time. The data bus remains the same for all of the variations in the address bus. Default configuration is 1T timing mode.

1.5.6 ODT Controls (Ask a Question)

To enable or disable the ODT for a specific rank of memory, configure the DDRC_ODT_PARAM_1_CR and DDRC_ODT_PARAM_2_CR registers. These registers must be configured before taking the controller out of soft reset. They are applied to every read or write issued by the controller.

1.5.7 Performance (Ask a Question)

The DDR controller has several performance registers that can be used to increase the speed of the read and write transactions to the DDR memory.

The DDR controller has a transaction store, shared for low and high priority transactions. The DDRC_PERF_PARAM_1_CR register can be configured for allocating the transaction store between the low and high priority transactions. For example, if the REG_DDRC_LPR_NUM_ENTRIES field is configured to 0, the controller allocates more time to high priority transactions. The ratio for LPR: HPR is 1:7 as the transaction store depth is 8. For more information, see Table 2-40.

The DDRC_HPR_QUEUE_PARAM_1_CR, DDRC_LPR_QUEUE_PARAM_1_CR, and DDRC_WR_QUEUE_PARAM_CR registers can be configured for the minimum clock values for treating the transactions in the HPR, LPR, and WR queues as critical and non-critical.

To force all incoming transactions to low priority, configure the DDRC_PERF_PARAM_2_CR register. By default, it is configured to force all the incoming transactions to low priority.

1.5.8 Dynamic DRAM Constraints (Ask a Question)

Timing parameters for DDR memories must be configured according to the DDR memory specification. Dynamic DRAM constraints are divided into three basic categories:

- **Bank Constraints**: Affect the transactions that are scheduled to a given bank
- **Rank Constraints**: Affect the transactions that are scheduled to a given rank
- **Global Constraints**: Affect all transactions

1.5.8.1 Dynamic DRAM Bank Constraints (Ask a Question)

The following table lists the timing constraints that affect the transactions to a bank. The control bit field must be configured according to the DDR memory vendor specification.

Table 1-2. Dynamically Enforced Bank Constraints

Timing Constraint of DDR Memory	Control Bit	Description
Row cycle time (t_{RC})	REG_DDRC_T_RC	Minimum time between two successive cycles activates to a given bank. For more information, see Table 2-21.
Row precharge command period (t_{RP})	REG_DDRC_T_RP	Minimum time from a precharge command to the next command affecting that bank. For more information, see Table 2-28.
Minimum bank active time ($t_{RAS(min)}$)	REG_DDRC_T_RAS_MIN	Minimum time from an activate command to a precharge command to the same bank. For more information, see Table 2-25.
Maximum bank active time ($t_{RAS(max)}$)	REG_DDRC_T_RAS_MAX	Maximum time from an activate command to a precharge command to the same bank. For more information, see Table 2-25.
RAS-to-CAS delay (t_{RCD})	REG_DDRC_T_RCD	Minimum time from an activate command to a Read or Write command to the same bank. For more information, see Table 2-28.
Write command period (t_{WR})	REG_DDRC_WR2PRE	Minimum time from a Write command to a precharge command to the same bank. For more information, see Table 2-23.
Read-to-precharge delay (t_{RTP})	REG_DDRC_RD2PRE	Minimum time from a Read command to a precharge command to the same bank. Set this to the current value of additive latency plus half of the burst length. For more information, see Table 2-23.

1.5.8.2 Dynamic DRAM Rank Constraints (Ask a Question)

The following table shows the timing constraints that affect the transactions to a rank. The control bit field must be configured according to the DDR memory vendor specification.

Table 1-3. Dynamically Enforced Rank Constraints

Timing Constraints of DDR Memory	Control Bit	Description
Nominal refresh cycle time (t_{RFC}(nom) or t_{REFI})	REG_DDRC_T_RFC_NOM_X32	Average time between refreshes for a given rank. The actual time between any two refresh commands could be longer or shorter than this; this represents the maximum time allowed between refresh commands to a given rank when averaged over a large period of time. For more information, see Table 2-6.
Minimum refresh cycle time $t_{RFC(min)}$	REG_DDRC_T_RFC_MIN	Minimum time between refreshes or activations. For more information, see Table 2-5.

.....continued		
Timing Constraints of DDR Memory	**Control Bit**	**Description**
RAS-to-RAS delay (t_{RRD})	REG_DDRC_T_RRD	Minimum time between activations from bank A to bank B. For more information, see Table 2-28.
RAS-to-CAS delay (t_{CCD})	REG_DDRC_T_CCD	Minimum time between two reads or two writes (from bank A to bank B). For more information, see Table 2-28.
Four active window (t_{FAW})	REG_DDRC_T_FAW	Sliding time window in which a maximum of 4 bank activations are allowed in an 8-bank design. In a 4-bank design, set this register to 0x1. For more information, see Table 2-21.

1.5.8.3 Dynamic DRAM Global Constraints (Ask a Question)

The following table lists the timing constraints that affect the global transactions. The control bit field must be configured according to the DDR memory vendor specification.

Table 1-4. Dynamic DRAM Global Constraints

Timing Constraint	Control Bit	Description
Read-to-write turnaround time	REG_DDRC_RD2WR	Minimum time to allow between issuing any Read command and WRITE command. For more information, see Table 2-26.
Write-to-read turnaround time	REG_DDRC_WR2RD	Minimum time to allow between issuing any Write command and issuing any Read command. For more information, see Table 2-26.
Write latency	REG_DDRC_WRITE_LATENCY	Time between write command and write data to DRAM. For more information, see Table 2-22.

The DDR memories require delays after initializing the mode registers. The following registers must be configured for delay requirements for the DDR memories. The DDR controller uses these delay values while initializing the DDR memories.

- DDRC_CKE_RSTN_CYCLES_1_CR. Recommended value is 0x4242. For more information, see Table 2-15.
- DDRC_ CKE_RSTN_CYCLES_2_CR. Recommended value is 0x8. For more information, see Table 2-16.

1.6 DDR PHY (Ask a Question)

DDR PHY provides a physical interface to DDR2, DDR3, and LPDDR1 SDRAM devices. It receives commands from the DDR controller and generates DDR memory signals required to access the external DDR memory.

The PHY adjusts the delay between the data (DQ) lines and DQS signals by using a separate DLL for each data slice. DDR has 36 DQ lines split into the following data slices:

- **Data slice0**: DQ0 – DQ7
- **Data slice1**: DQ8 – DQ15
- **Data slice2**: DQ16 – DQ23
- **Data slice3**: DQ24 – DQ31
- **Data slice4**: DQ32 – DQ35

1.7 Initialization (Ask a Question)

Before the FDDR subsystem becomes active, it goes through an initialization phase, which includes reset sequence, ZQ calibration, DRAM training, and memory initialization.

The FDDR controller registers must also be initialized via the APB configuration interface. When using the FDDR Controller (FDDRC) with Initialization, the FDDR core is automatically initialized with the desired user settings via an auto-instantiated CoreABC IP core.

1.7.1 CoreABC Initialization Sequence (Ask a Question)

The FDDR with initialization IP (RTG4FDDRC_INIT) includes a built-in CoreABC for configuring the FDDRC registers and initializing FDDRC.

During initialization, CoreABC implements the following:

- Performs APB writes to set the FDDR PLL (FPLL) VCO to high frequency range.
- Issues the PLL reset to allow the VCO to calibrate to a high gain setting.
- Performs APB writes to restore the required FPLL frequency.
- Waits for the FPLL to Lock.
- Resumes initialization of FDDRC registers as per the user settings.

For more information about the FDDR initialization state machine, see RTG4 DDR Memory Controller with Initialization Configuration User Guide.

1.7.2 Guidelines for FDDR with Built-In CoreABC (Ask a Question)

The following points must be considered for proper FDDR initialization:

- If the reference clock (CLK_BASE) to the FDDR FPLL is sourced from an RTG4 Fabric CCC (FCCC) with enhanced PLL calibration, then the user must keep the FDDR with Initialization INIT_RESET_N input asserted until the fabric PLL lock has been acquired. This will prevent the FPLL calibration sequence and FDDR initialization from proceeding prior to achieving a stable reference clock. Figure 1-13 shows the FDDR and CoreABC SmartDesign connections.
- If the release of INIT_RESET_N can't be directly connected to the Fabric PLL Lock output (when the FCCC supplies the FDDR CLK_BASE), the CoreABC program instructions can be manually updated to delay the FDDR initialization process. This delay can be achieved using any one of the following methods.
  - Assert the FPLL power-down at the start of the CoreABC code sequence (using APBWRT DAT16 0 0x1508 0x32e4). Then, using an additional IO input to CoreABC, poll for the fabric PLL Lock to go high (using WAIT UNTIL INPUT#). Then, deassert the FPLL POWERDOWN (using APBWRT DAT16 0 0x1508 0x12e4).
  - Add a wait state at the start of the CoreABC init code to wait for the max FCCC PLL lock acquisition time (500 µs) + ~160 µs fabric PLL calibration time, before allowing the FDDR CoreABC initialization sequence to run.

Tip: The CoreABC program instructions can be updated on the **Program** tab of the CoreABC Configurator, as shown in Figure 1-12.

Figure 1-12. CoreABC Program Tab

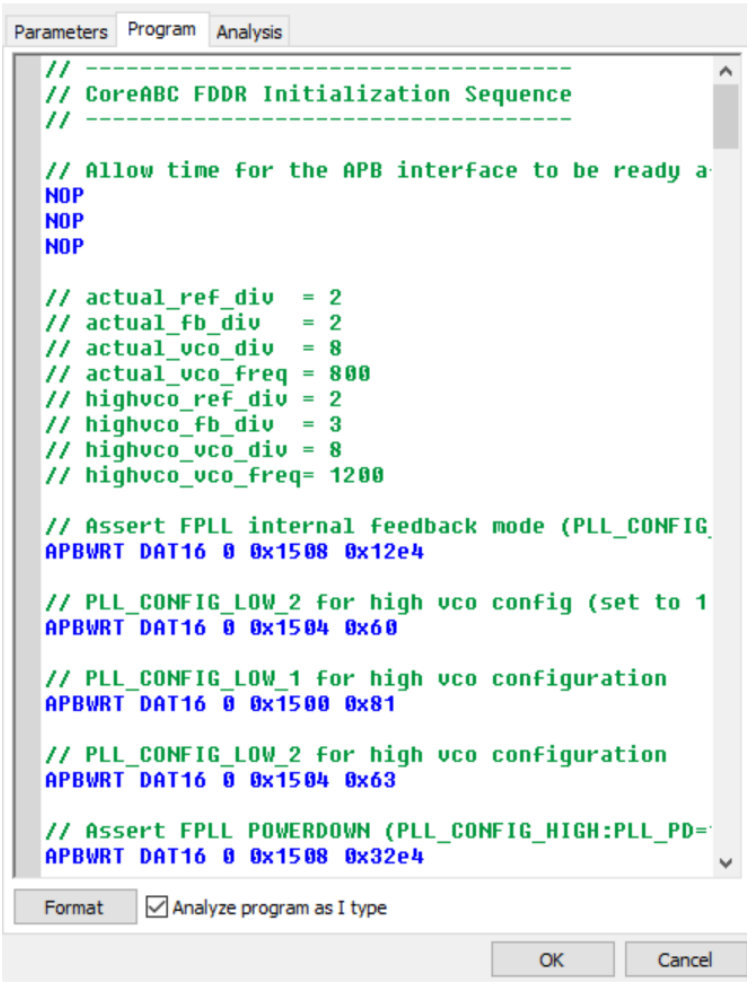
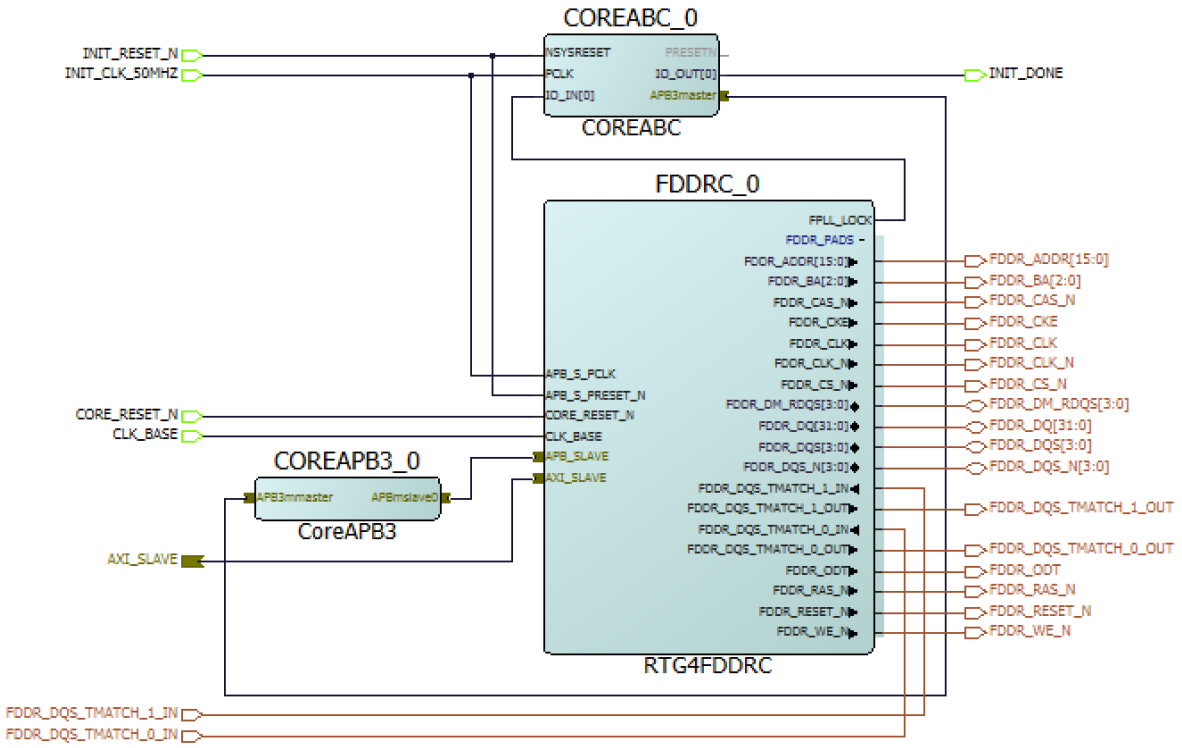


Figure 1-13. FDDR Initialization Connection



1.7.3 Reset Sequence (Ask a Question)

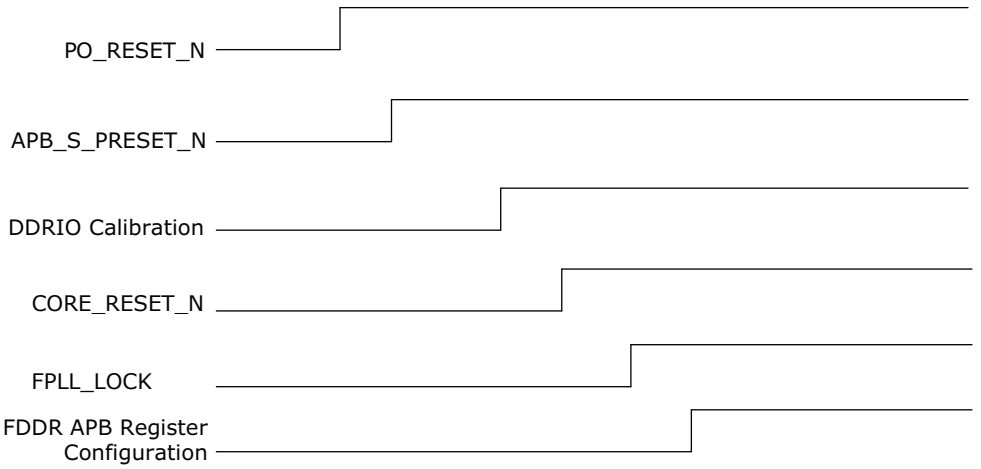
The following figure shows the reset sequence required for the FDDR subsystem from the power-on-reset stage. Once the CORE_RESET_N signal of the FDDR subsystem is asserted, the FPLL_LOCK signal is asserted and the APB register configuration is complete. Assertion of CORE_RESET_N denotes the end of the reset sequence. The DDR controller performs external DRAM memory reset and initialization according to the JEDEC specification, including reset, refresh, and mode registers. For more information about Reset sequence see AN4752: Interfacing RTG4 with External DDR3 Memory.

Important: The REG_DDRC_SOFT_RSTB bit of the DDRC_DYN_SOFT_RESET_CR register can be set to 0 to reset the DDR controller. To release the DDR controller from reset, set the REG_DDRC_SOFT_RSTB bit of the DDRC_DYN_SOFT_RESET_ALIAS_CR register to 1.

1.7.3.1 DDR I/O Calibration (Ask a Question)

Each DDR I/O has an ODT feature; the ODT is calibrated depending on the DDR I/O standard. DDR I/O calibration occurs after the DDR I/Os are enabled. I/O calibration is always enabled when the DDR subsystem is configured for DDR2 and DDR3 memories. I/O calibration can be optionally disabled when controller is configured to LPDDR. If I/O calibration is enabled, all I/Os in the DDR bank are calibrated even if the DDR controller does not use all I/Os in the bank. For more information on DDR I/O calibration, see the UG0741: RTG4 FPGA I/O User Guide.

Figure 1-14. Reset Sequence



1.7.4 ZQ Calibration (Ask a Question)

ZQ calibration is applicable only for DDR3. It is used to calibrate the DRAM output drivers (R_{ON}) and ODT values. DDR3 SDRAM requires a longer time to calibrate R_{ON} and ODT at initialization and a relatively lesser time to perform periodic calibrations.

The DDR controller performs ZQ calibration by issuing a ZQ Calibration Long (ZQCL) command and ZQ Calibration Short (ZQCS) command.

ZQCL is used to perform initial calibration during the power-up initialization sequence. This command is allowed for a period of t_{ZQinit}, as specified by a memory vendor that you use. The value of t_{ZQinit} can be modified using the register bits, REG_DDRC_T_ZQ_LONG_NOP. For more information, see Table 2-36.

The ZQCS command is used to perform periodic calibration to account for voltage and temperature variations. A shorter timing window is provided to perform calibration and transfer of values as defined by timing parameter t_{ZQCS}. The t_{ZQCS} parameter can be modified using the register bits, REG_DDRC_T_ZQ_SHORT_NOP. For more information, see Table 2-37. Other activities are not performed by the controller for the duration of t_{ZQinit} and t_{ZQCS}. All DRAM banks are precharged and t_{RP} is met before the ZQCL or ZQCS command is issued by the DDR controller.

1.7.5 DRAM Training (Ask a Question)

High-Speed DDR3 memories typically require the DDR controller to implement Write-Leveling, Read DQS Gate Training, and Read Data Eye Training. However, the RTG4 FDDR only supports a maximum data rate of 333 MHz/667 Mbps, which means the clock period and data window are relatively large compared to high-speed DDR3 memory interfaces. Therefore, dynamic write-leveling and read training are not performed. The following sections describe how write-leveling and read training are addressed by RTG4 FDDR.

1.7.5.1 Write-Leveling (Ask a Question)

Dynamic write-leveling is not required for the RTG4 FDDR controller. The board layout need to follow the guidelines as per the AC439: RTG4 FPGAs Board Design and Layout Guidelines to keep the skew between DQS and CK within the JEDEC DDR3 t_{DQSS} limit of ±750 ps at each memory device.

For board layouts, which do not meet the RTG4 Board Design Guidelines, the RTG4 FDDR controller allows static delay ratios which delays DQS for each byte lane so that the skew between DQS and CK is kept within JEDEC limits. REG_PHY_WR_DQS_SLAVE_RATIO [49:0] is split across four registers called PHY_WR_DQS_SLAVE_RATIO_#_CR, where # ranges from 1 to 4. The 50 bit ratio value is allocated to five write DQS delay ratios so that each data slice can have its DQS delayed independently as follows:

- [9:0] for DQS0
- [19:10] for DQS1
- [29:20] for DQS2
- [39:30] for DQS3
- [49:40] for DQS4

Each 10 bit ratio value is calculated using the following formula to delay DQS by the DELAY value in picoseconds (ps):
Ratio = (DELAY (in ps) * 256) / DDR3_CLK_PERIOD (in ps)

1.7.5.2 Read-Leveling (Ask a Question)

RTG4 FDDR does not perform the dynamic Read DQS Gate Training and Data Eye Training. Instead, these functions are achieved by using built-in static delay values automatically generated by Libero SoC for the FDDR automatic register initialization.

1.7.5.3 Read Gate (Ask a Question)

The DQS gate is aligned by the Libero SoC auto-generated FDDR initialization code containing fixed delay ratios to account for board round-trip time between RTG4 and the DDR3 memory. The TMATCH_OUT and TMATCH_IN signals are shortened close to the FPGA balls to remove the RTG4 output and input delays from the round-trip delay time. Therefore, the fixed delay ratios represent only the board delay.

The fixed delay ratios work in combination with board layouts which follows the AC439: RTG4 FPGAs Board Design and Layout Guidelines.

1.7.5.4 DQS Alignment within Data Eye (Ask a Question)

The incoming read DQS is internally centered within the read DQ data window using a static delay ratio. This static delay is applied by the Libero SoC auto-generated FDDR initialization code. The fixed delay ratios work in combination with board layouts, which follow the AC439: RTG4 FPGAs Board Design and Layout Guidelines.

Tip: The Libero SOC auto-generated delay ratio for read DQS data eye centering is written to the required register.

1.7.6 DDR Memory Initialization Time (Ask a Question)

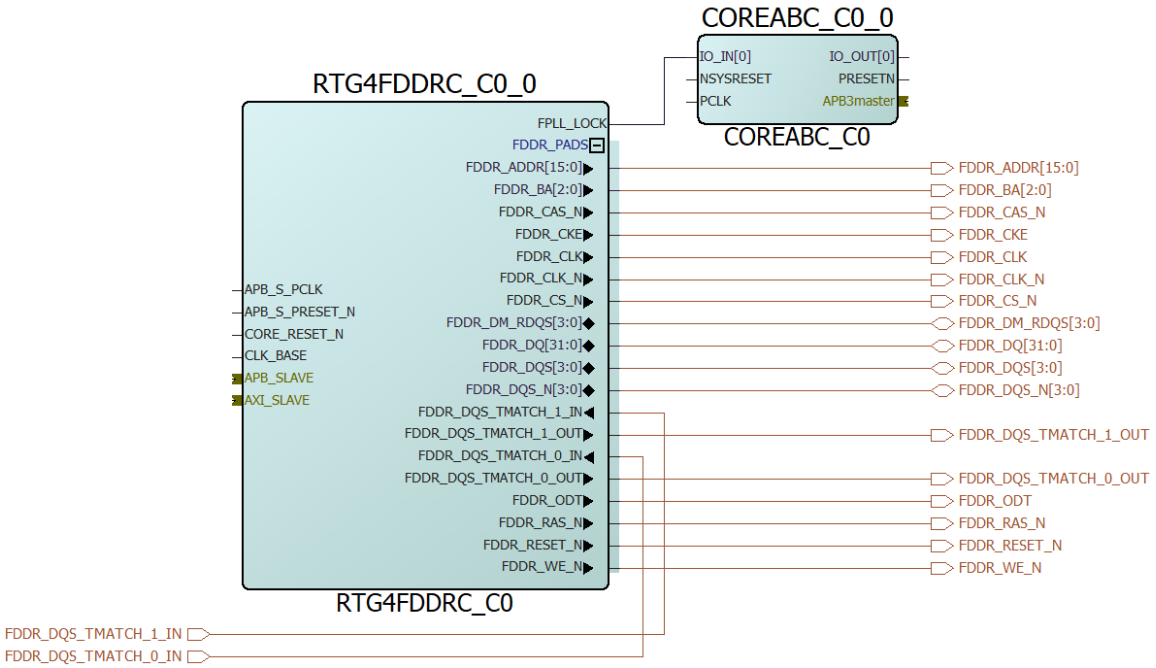
The time to initialize the DDR memory depends on the following factors:

- Power-up and register initialization by System Controller
- DDR controller and PHY configuration registers initialization by CoreABC
- DDR memory initialization by DDR controller according to JEDEC standard (mode register configuration and training)
- DDR memory settling time configured in the DDR memory controller configuration window

Important:

- If the reference clock (CLK_BASE) to the FDDR FPLL is sourced from RTG4 FCCC with an enhanced PLL Calibration, you must stagger the release of the FDDR initialization with INIT_RESET_N until the fabric PLL lock is asserted. This ensures that the reference clock is stabilized before the FPLL calibration and FDDR initialization.
- If you cannot stagger the INIT_RESET_N release until the fabric PLL is locked, the FDDR initialization with underlying SmartDesign connections and CoreABC code can be modified to any one of the following:
  - Assert FPLL Power Down at the start of the CoreABC code sequence. Release the FPLL from Power Down and let the generated initialization sequence run after the fabric PLL lock is signaled by an added I/O Input to CoreABC connected to the fabric PLL lock output. For further details, see Example.
  - Add a wait state at the start of the CoreABC init code to wait for the maximum PLL lock acquisition time (500 µs + ~150 µs fabric PLL calibration time) before allowing the FDDR CoreABC initialization sequence to run. For further details, see Figure 1-15

Figure 1-15. FDDR Initialization Using CoreABC



For example
CoreABC sequence to add at the start of the init code:

Some CoreABC init code at the start of the sequence to initially put the FPLL into POWERDOWN mode (APBWRT DAT16 0 0x1508 0x32e4).

- Poll for the fabric PLL Lock to go high (WAIT UNTIL INPUT#)
- De-assert the FPLL POWERDOWN (APBWRT DAT16 0 0x1508 0x12e4)

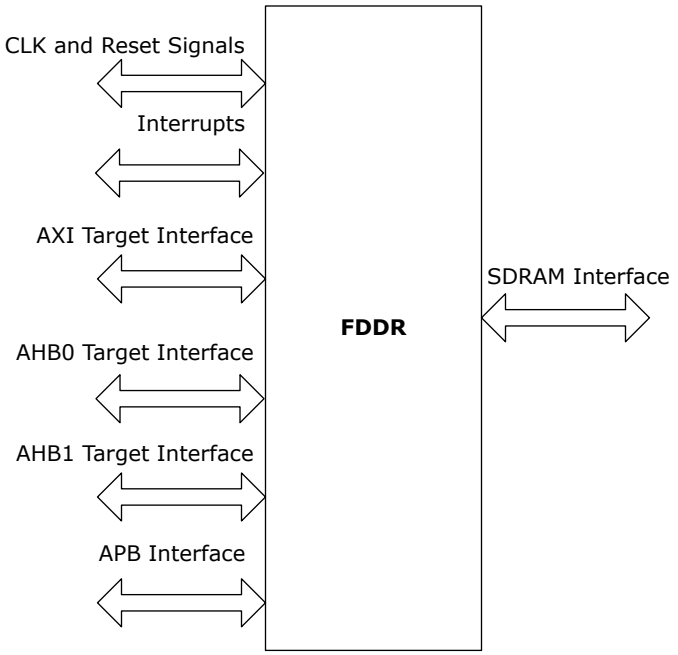
Follow the enhanced CoreABC init code generated by v12.4 FDDR with the Init core.

1.8 FDDR Subsystem Ports (Ask a Question)

The following figure shows the DDR controller ports categorized into the following sub-categories:

- **CLK and Reset Signal**: Clock and reset signals required for the FDDR.
- **Interrupts**: FDDR has five interrupts to interrupt you logic to indicate the ECC error, PLL lock, and so on.
- **AXI Target Interface**: Signals required when the DDR controller is configured in AXI mode. To perform DDR memory transactions, an AXI initiator is implemented in the FPGA fabric logic interfaces with the DDR controller.
- **AHB0/AHB1 Target Interface**: Signals required when the DDR controller is configured in AHB mode. To perform DDR memory transactions, an AHB initiator is implemented in the FPGA fabric logic interfaces with the DDR Controller.
- **APB Interface**: Allows user logic or software to configure the DDR controller.
- **SDRAM Interface**: Signals are connected to the SDRAM.

Figure 1-16. DDR Controller Port List



The following tables list the signals for each of the interfaces.

Table 1-5. Clock and Reset Signals

Signal Name	Type	Polarity	Description
APB_S_PCLK	In	—	APB clock. This clock drives all the registers of the APB interface.
APB_S_PRESET_N	In	Low	APB reset signal. This is an Active-Low signal. This drives the APB interface and is used to generate the soft reset for the DDR controller as well.
CORE_RESET_N	In	Low	Global reset. This resets the following blocks: DDR_FIC DDRC PHY DDRAXI
CLK_BASE	In	—	Base clock to the FDDR clock controller.
CLK_BASE_PLL_LOCK	In	High	Fabric PLL lock input.
FPLL_LOCK	Out	High	PLL lock status of DDR PLL.

Table 1-6. Interrupt Signals

Signal Name	Type	Polarity	Description
PLL_LOCK_INT	Out	High	PLL lock interrupt.
PLL_LOCKLOST_INT	Out	High	PLL lock lost interrupt.
ECC_INT	Out	High	Sticky interrupt on APB clock. Generated on ECC errors from the DDR controller.
IO_CALIB_INT	Out	High	Sticky Interrupt on APB clock. Generated on code lock from the I/O calibration block.
FIC_INT	Out	High	Sticky interrupt on APB clock. Generated on error conditions from DDR_FIC.
PLL_LOCK_INT	Out	High	PLL lock interrupt

The following are the limitations of AXI:

- Data must be provided on the same order as the write commands. The block ignores the WID input. Data is assumed to be following the last write address loaded.
- Each AXI channel supports up to four outstanding read and one outstanding write transactions.
- Fixed type burst is not supported.

The AXI target interface signals are available only if the FDDR interface is configured for AXI mode. For more details on AXI protocol, see the AMBA AXI v1.0 Protocol Specification (ARM Documentation).

Table 1-7. AXI Target Interface Signals

Signal Name	Direction	Polarity	Description
AXI_S_ARREADY	Output	High	Indicates whether the target is ready to accept an address and associated control signals. 1: Target ready 0: Target not ready
AXI_S_AWREADY	Output	High	Indicates that the target is ready to accept an address and associated control signals. 1: Target ready 0: Target not ready
AXI_S_BID[3:0]	Output	—	Indicates response ID. The identification tag of the write response.
AXI_S_BRESP[1:0]	Output	—	Indicates write response and the status of the write transaction. 00: Normal access okay 01: Exclusive access okay 10: Target error 11: Decode error
AXI_S_BVALID	Output	High	Indicates whether a valid write response is available. 1: Write response available. 0: Write response not available.
AXI_S_RDATA[63:0]	Output	—	Indicates a read data.
AXI_S_RID[3:0]	Output	—	Read an ID tag. ID tag of the read data group of signals.
AXI_S_RLAST	Output	High	Indicates the last transfer in a read burst.
AXI_S_RRESP[1:0]	Output	—	Indicates a read response and the status of the read transfer. 00: Normal access okay 01: Exclusive access okay 10: Target error 11: Decode error
AXI_S_RVALID	Output	—	Indicates whether the required read data is available and the read transfer can complete. 1: Read data available 0: Read data not available

.....continued

Signal Name	Direction	Polarity	Description
AXI_S_WREADY	Output	High	Indicates whether the target can accept the write data. 1: Target ready 0: Target not ready
AXI_S_ARADDR[31:0]	Input	—	Indicates the initial address of a read burst transaction.
AXI_S_ARBURST[1:0]	Input	—	Indicates burst type. 00: FIXED. Fixed-address burst FIFO type 01: INCR. Incrementing-address burst normal sequential memory 10: WRAP. Incrementing-address burst that wraps to a lower address at the wrap boundary 11: Reserved
AXI_S_ARID[3:0]	Input	—	Indicates the identification tag for the read address group of signals.
AXI_S_ARLEN[3:0]	Input	—	Indicates the burst length. The burst length gives the exact number of transfers in a burst. 0000: 1 0001: 2 ... 1111: 16
AXI_S_ARLOCK[1:0][1]	Input	—	Indicates the lock type. This signal provides an additional information about the atomic characteristics of the read transfer. 00: Normal access 01: Exclusive access 10: Locked access 11: Reserved
AXI_S_ARSIZE[1:0]	Input	—	Indicates the maximum number of data bytes to transfer in each data transfer, within a burst. 00: 1 01: 2 10: 4 11: 8
AXI_S_ARVALID	Input	High	Indicates the validity of read address and control information. 1: Address and control information valid 0: Address and control information not valid
AXI_S_AWADDR[31:0]	Input	—	Indicates write address. The write address bus gives the address of the first transfer in a write burst transaction.
AXI_S_AWBURST[1:0]	Input	—	Indicates burst type. 00: FIXED. Fixed-address burst FIFO-type 01: INCR. Incrementing-address burst normal sequential memory 10: WRAP. Incrementing-address burst that wraps to a lower address at the wrap boundary 11: Reserved
AXI_S_AWID[3:0]	Input	—	Indicates the identification tag for the write address group of signals.
AXI_S_AWLEN[3:0]	Input	—	Indicates the burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address. 0000: 1 0001: 2 ... 1111: 16

.....continued			
Signal Name	**Direction**	**Polarity**	**Description**
AXI_S_AWLOCK[1:0][2]	Input	—	Indicates the lock type. This signal provides an additional information about the atomic characteristics of the write transfer. 00: Normal access 01: Exclusive access 10: Locked access 11: Reserved
AXI_S_AWSIZE[1:0]	Input	—	Indicates the maximum number of data bytes to transfer in each data transfer, within a burst. 00: 1 01: 2 10: 4 11: 8
AXI_S_AWVALID	Input	High	Indicates whether a valid write address and control information are available. 1: Address and control information available 0: Address and control information not available
AXI_S_BREADY	Input	High	Indicates whether the initiator can accept the response information. 1: Initiator ready 0: Initiator not ready
AXI_S_RREADY	Input	High	Indicates whether the initiator can accept the read data and response information. 1: Initiator ready 0: Initiator not ready
AXI_S_WDATA[63:0]	Input	—	Indicates the write data.
AXI_S_WID[3:0]	Input	—	Indicates the response ID, the identification tag of the write response.
AXI_S_WLAST	Input	High	Indicates the last transfer in a write burst.
AXI_S_WSTRB[7:0]	Input	—	Indicates which byte lanes to update in memory.
AXI_S_WVALID	Input	High	Indicates whether a valid write data and strobes are available. 1: Write data and strobes available 0: Write data and strobes not available

Important:

1. Only four initiator IDs configured in DDRC_AXI_FABRIC_PRI_ID_CR register support exclusive access. For more information, see Table 2-52.
2. Only four initiator IDs configured in DDRC_AXI_FABRIC_PRI_ID_CR register support exclusive access. For more information, see Table 2-52.

The AHB target interface signals are available only if the FDDR interface is configured for a single or dual AHB mode. For more details on AHB protocol, see the AMBA AHB v3.0 Protocol Specification (ARM Documentation). AHBx indicates AHB0 or AHB1.

Table 1-8. FDDR AHB Target Interface Signals

Signal Name	Direction	Polarity	Description
AHBx_S_HREADYOUT	Output	High	Indicates that a transfer is completed on the bus. The signal is asserted low to extend a transfer. Input to Fabric initiator.
AHBx_S_HRESP	Output	High	Indicates the AHB transfer response to the Fabric initiator.
AHBx_S_HRDATA[31:0]	Output	—	Indicates the AHB read data to the Fabric initiator.
AHBx_S_HSEL	Input	High	Indicates the AHB target select signal from the Fabric initiator.

.....continued			
Signal Name	**Direction**	**Polarity**	**Description**
AHBx_S_HADDR[31:0]	Input	—	Indicates the AHB address initiated by the Fabric initiator.
AHBx_S_HBURST[2:0]	Input	—	Indicates the AHB burst type from the Fabric initiator. 000: Single burst 001: Incrementing burst of undefined length 010: 4-beat wrapping burst 011: 4-beat incrementing burst 100: 8-beat wrapping burst 101: 8-beat incrementing burst 110: 16-beat wrapping burst 111: 16-beat incrementing burst
AHBx_S_HSIZE[1:0]	Input	—	Indicates the AHB transfer size from the Fabric initiator. 00: 8-byte 01: 16-Half word 10: 32-Word
AHBx_S_HTRANS[1:0]	Input	—	Indicates the AHB transfer type from the Fabric initiator. 00: IDLE 01: BUSY 10: NONSEQUENTIAL 11: SEQUENTIAL
AHBx_S_HMASTLOCK	Input	High	Indicates the AHB initiator lock signal from the Fabric initiator.
AHBx_S_HWRITE	Input	High	Indicates the AHB write-control signal from the Fabric initiator.
AHBx_S_HREADY	Input	High	Indicates that a transfer is completed on the bus. The Fabric initiator can drive this signal to Low to extend a transfer.
AHBx_S_HWDATA[31:0]	Input	—	Indicates the AHB write data from the Fabric initiator.

For more information about the APB protocol, see the AMBA APB v3.0 Protocol Specification (ARM Documentation).

Table 1-9. FDDR APB Target Interface Signals

Signal Name	Direction	Polarity	Description
APB_S_PREADY	Output	High	Indicates the APB Ready signal to the Fabric initiator.
APB_S_PSLVERR	Output	High	Indicates the error condition on an APB transfer to the Fabric initiator.
APB_S_PRDATA[15:0]	Output	—	Indicates the APB read data to the Fabric initiator.
APB_S_PENABLE	Input	High	Indicates the APB enable from the Fabric initiator. The enable signal is used to indicate the second cycle of an APB transfer.
APB_S_PSEL	Input	High	Indicates the APB target select signal from the Fabric initiator.
APB_S_PWRITE	Input	High	Indicates the APB write-control signal form the Fabric initiator.
APB_S_PADDR[10:2]	Input	—	Indicates the APB address initiated by the Fabric initiator.
APB_S_PWDATA[15:0]	Input	—	Indicates the APB write data from the Fabric initiator.

TMATCH_IN and TMATCH_OUT pins must be connected together outside the device. They are used for gate training as part of the read data capture operation. These two pins create an internal DQS Enable signal that is used to calibrate the data path. DQS must be gated to prevent false triggering of the FIFO write clock. This DQS Enable signal is derived from the system clock and physically matches the clock output buffer and DQS input buffer to compensate for I/O buffer uncertainty due to Process-Voltage-Temperature (PVT) changes. Without this connection, the circuit is not operable.

Table 1-10. SDRAM Signals

Signal Name	Type	Polarity	Description
FDDR_CAS_N	Out	Low	DRAM CASN
FDDR_CKE	Out	High	DRAM CKE
FDDR_CLK	Out	—	DRAM single-ended clock— for differential pads
FDDR_CLK_N	Out	—	DRAM single-ended clock— for differential pads
FDDR_CS_N	Out	Low	DRAM CSN
FDDR_ODT	Out	High	DRAM ODT 0: Termination OFF 1: Termination ON
FDDR_RAS_N	Out	Low	DRAM RASN
FDDR_ RESET_N	Out	Low	DRAM reset for DDR3
FDDR_WE_N	Out	Low	DRAM WEN
FDDR_ADDR[15:0]	Out	—	DRAM address bits
FDDR_BA[2:0]	Out	—	DRAM Bank Address
FDDR_DM_RDQS[3:0]	In/out	—	DRAM data mask— from bidirectional pads
FDDR_DQS[3:0]	In/out	—	DRAM single-ended data strobe output—for bidirectional pads.
FDDR_DQS_N[3:0]	In/out	—	DRAM single-ended data strobe output— for bidirectional pads.
FDDR_DQ[31:0]	In/out	—	DRAM data input or output— for bidirectional pads.
FDDR_DQ_ECC[3:0]	In/out	—	DRAM data input or output for SECDED.
FDDR_DM_RDQS_ECC	In/out	High	DRAM single-ended data strobe output— for bidirectional pads.
FDDR_DQS_ECC	In/out	High	DRAM single-ended data strobe output— for bidirectional pads.
FDDR_DQS_ECC_N	In/out	Low	DRAM data input or output— for bidirectional pads.
FDDR_DQS_TMATCH_0_IN	In	High	DQS enable input for timing match between DQS and system clock. For simulations, tie to FDDR_DQS_TMATCH_0_OUT.
FDDR_DQS_TMATCH_1_IN	In	High	DQS enable input for timing match between DQS and system clock. For simulations, tie to FDDR_DQS_TMATCH_1_OUT.
FDDR_DQS_TMATCH_0_OUT	Out	High	DQS enable output for timing match between DQS and system clock. For simulations, tie to FDDR_DQS_TMATCH_0_IN.
FDDR_DQS_TMATCH_1_OUT	Out	High	DQS enable output for timing match between DQS and system clock. For simulations, tie to FDDR_DQS_TMATCH_1_IN.
FDDR_DQS_TMATCH_ECC_IN	In	High	DQS enable input for timing match between DQS and system clock. For simulations, tie to FDDR_DQS_TMATCH_ECC_OUT.
FDDR_DQS_TMATCH_ECC_OUT	Out	High	DQS enable output for timing match between DQS and system clock. For simulations, tie to FDDR_DQS_TMATCH_ECC_IN.

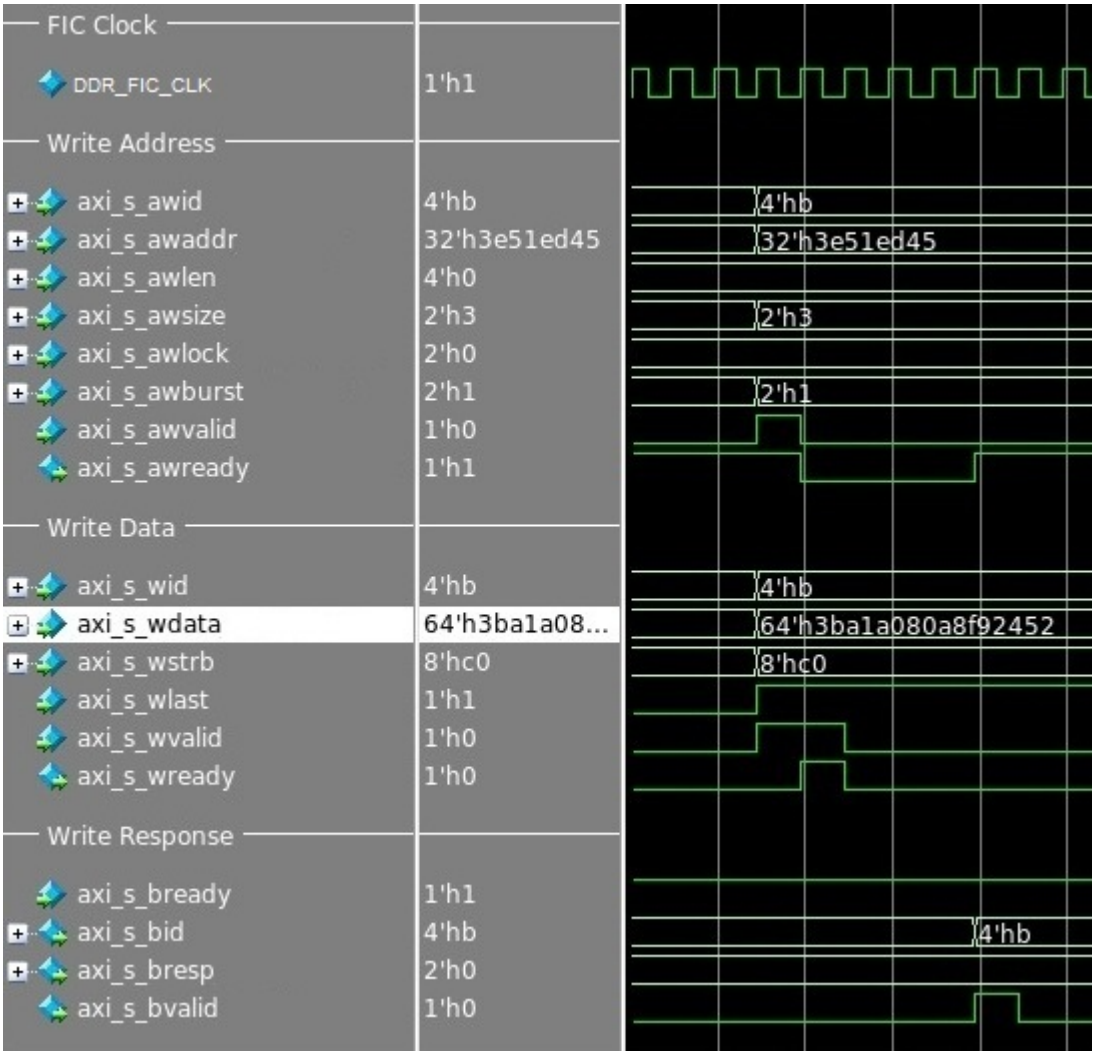
1.9 Functional Timing Diagrams (Ask a Question)

This section describes the operation of the DDR controller using the AXI and APB interfaces with timing diagrams. The DDR3 16 bit micron memory model is used to perform the read and write transactions from FDDR Fabric Interface (DDR_FIC). The AXI or AHB clock is configured for 166 MHz; the FDDR clock is configured at 332 MHz, that is, the FIC clock to the FDDR clock ratio is 1:2.

1.9.1 AXI-Single Write Transaction (Ask a Question)

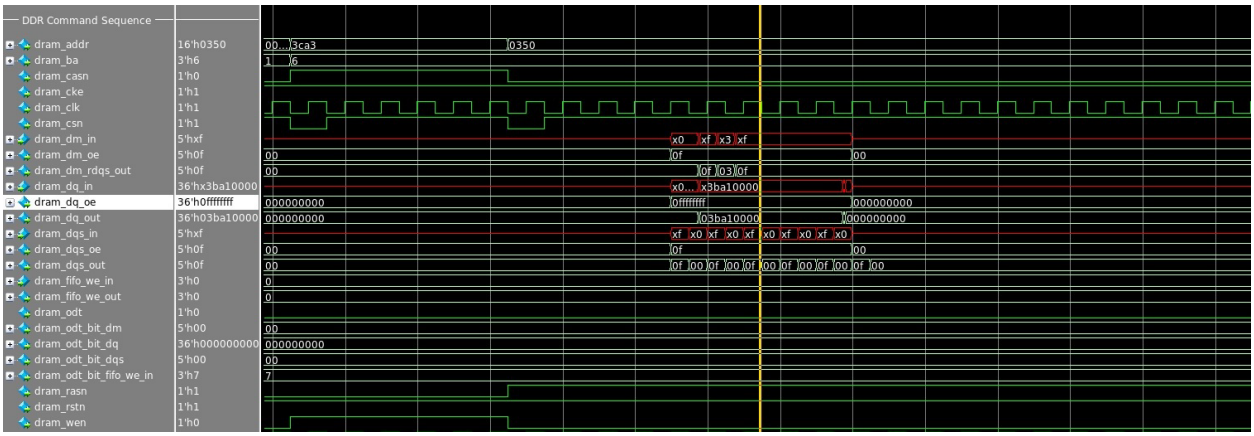
The following figure shows the AXI-single write transaction.

Figure 1-17. AXI - Single Write Transaction



The following figure shows the DDR controller command sequence for a single AXI write transaction.

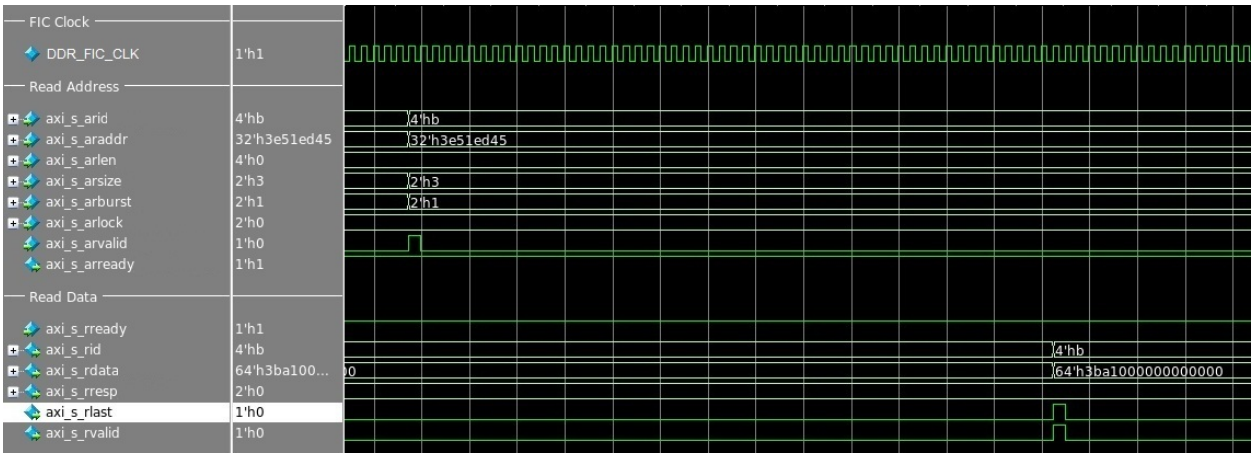
Figure 1-18. DDR Controller Command Sequence for a Single AXI Write Transaction



1.9.2 AXI Single Read Transaction (Ask a Question)

The following figure shows the AXI single read transaction.

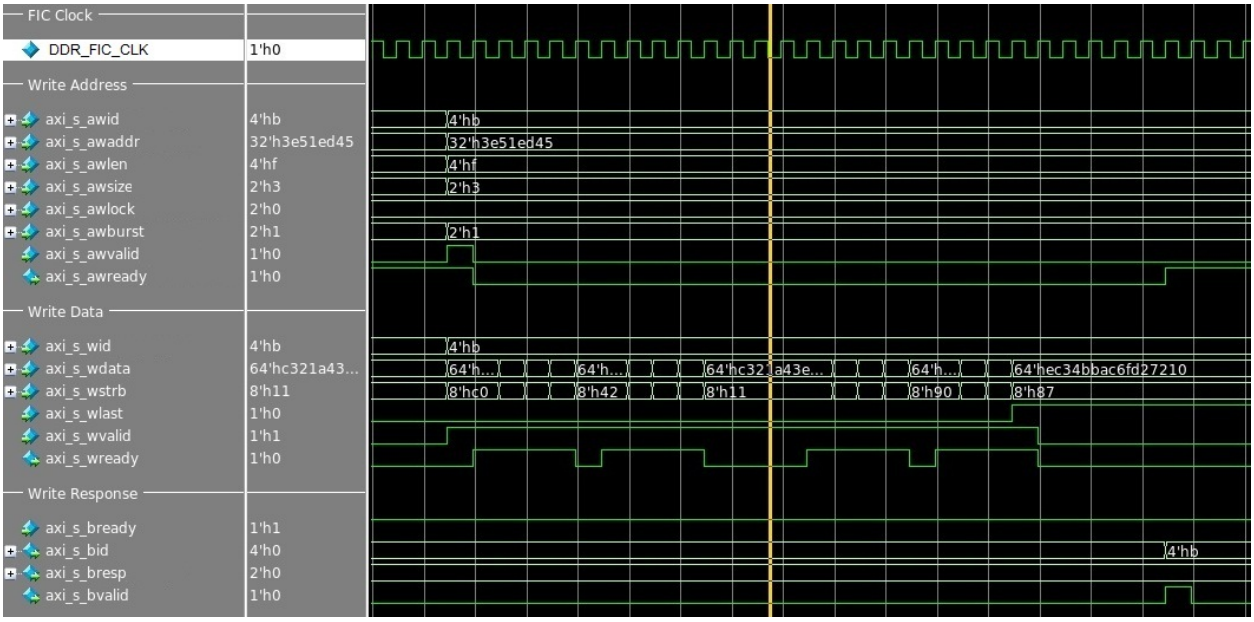
Figure 1-19. AXI Single Read Transaction



1.9.3 AXI Burst Write Transaction (INCR–16) (Ask a Question)

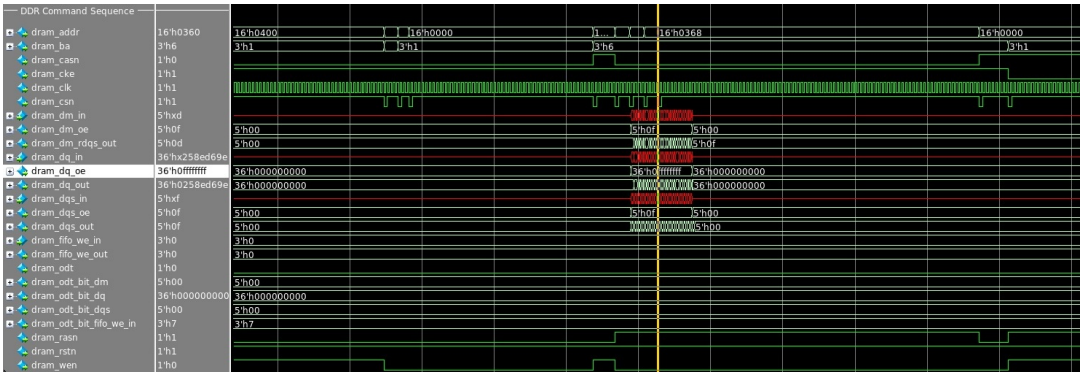
The following figure shows the AXI INCR-16 write transaction.

Figure 1-20. AXI INCR-16 Write Transaction



The following figure shows the DDR controller command sequence for AXI INCR-16 write transaction.

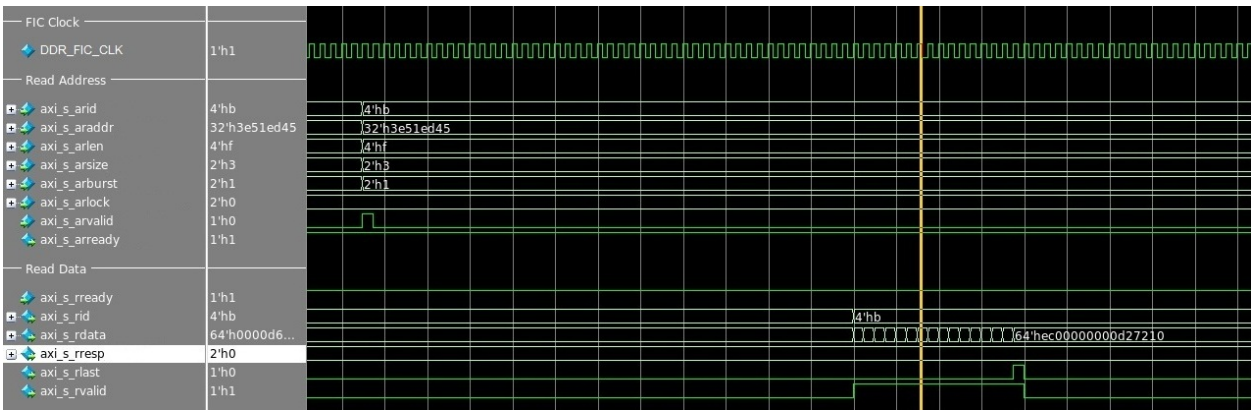
Figure 1-21. DDR Controller Command Sequence for AXI INCR-16 Write Transaction



1.9.4 AXI Burst Read Transaction (INCR–16) (Ask a Question)

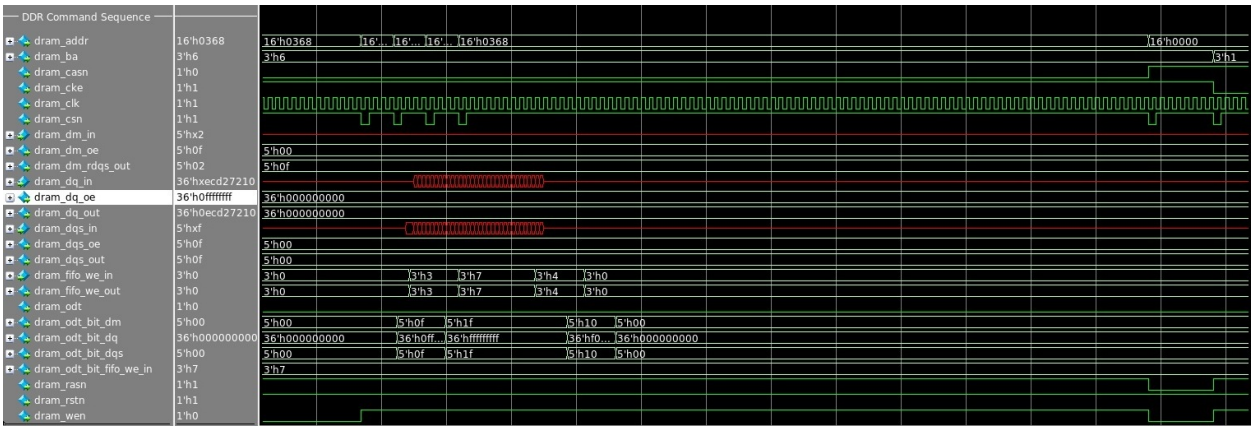
The following figure shows the AXI INCR-16 read transaction.

Figure 1-22. AXI INCR-16 Read Transaction



The following figure shows the DDR controller command sequence for AXI INCR-16 read transaction.

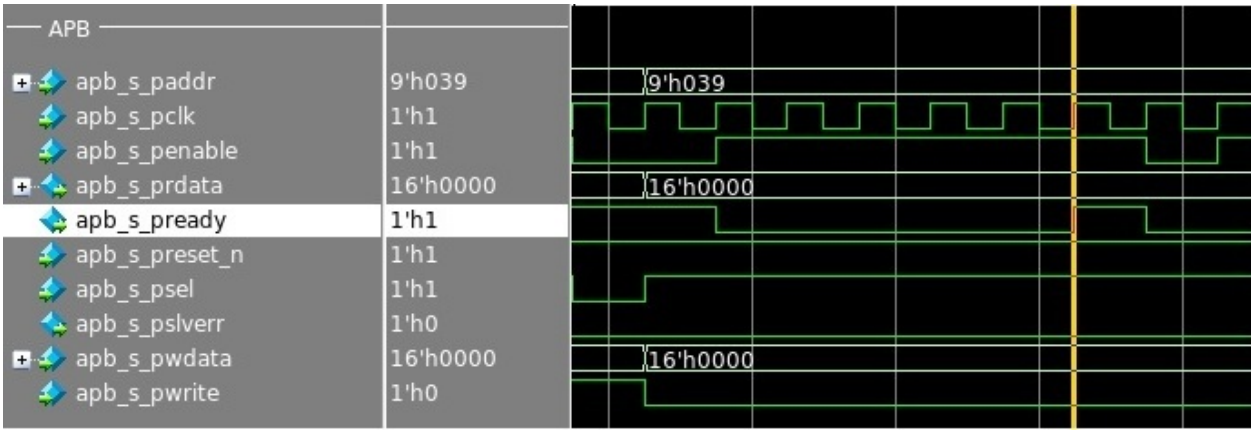
Figure 1-23. DDR Controller Command Sequence for AXI INCR-16 Read Transaction



1.9.5 APB Read Transaction (Ask a Question)

The following figure shows the APB read transaction.

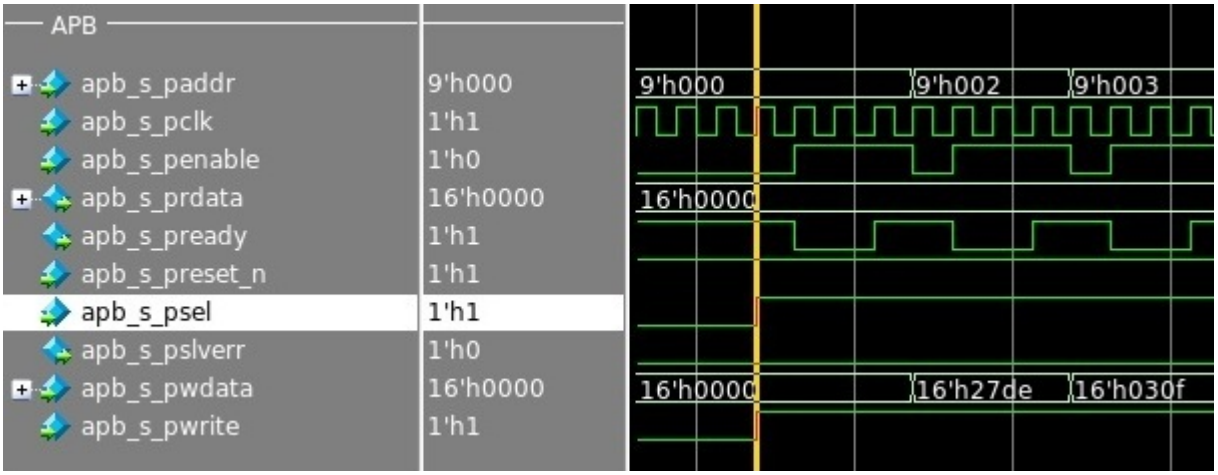
Figure 1-24. APB Read Transaction



1.9.6 APB Write Transaction (Ask a Question)

The following figure shows the APB write transaction.

Figure 1-25. APB Write Transaction



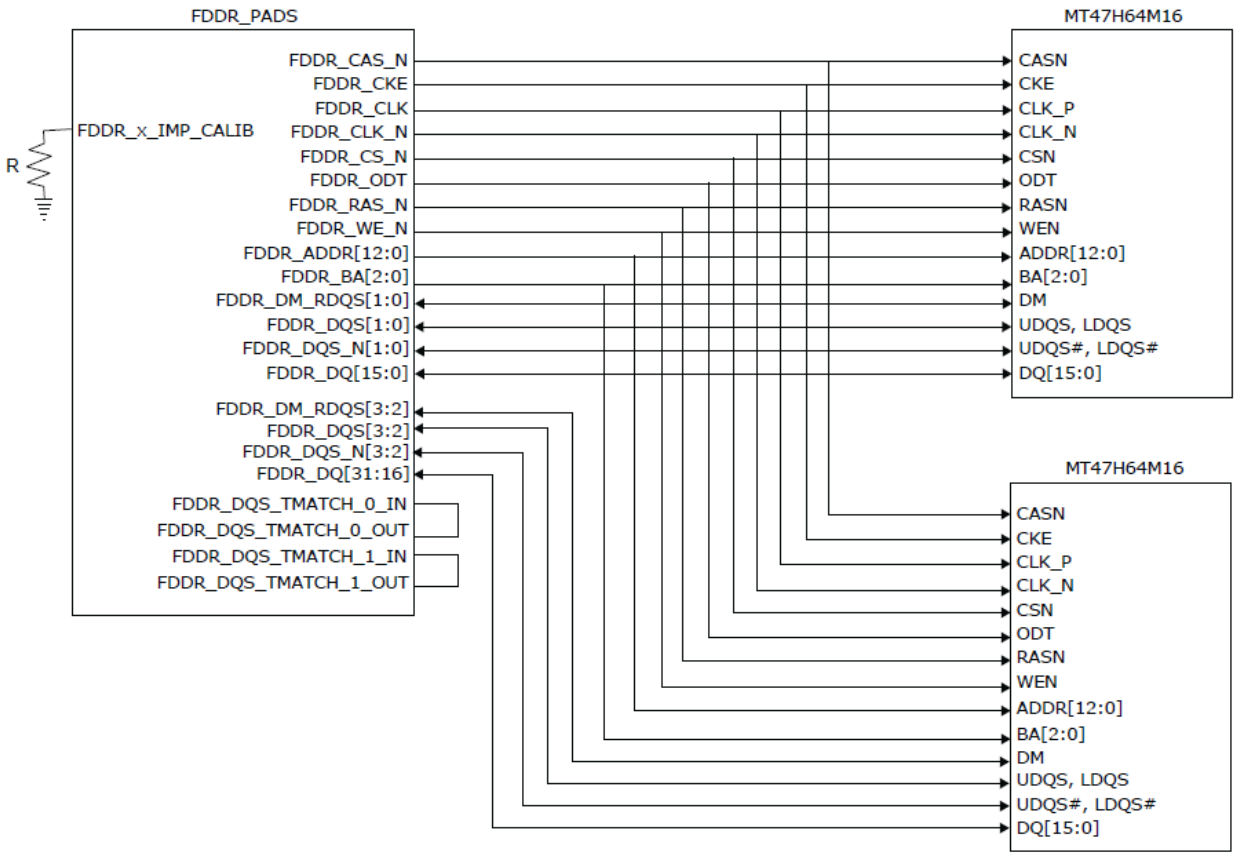
1.10 DDR Memory Device Examples (Ask a Question)

This section describes how to connect DDR memories to RTG4 FDDR_PADs with examples.

1.10.1 Example 1: Connecting 32 Bit DDR2 to FDDR_PADs (Ask a Question)

The following figure shows the DDR2 SDRAM connected to the FDDR subsystem of an RTG4 device. Micron's MT47H64M16 is a 128 MB density device with x16 data width. The FDDR subsystem is configured in Full Bus Width mode and without SECDED. The total DDR2 memory connected to the FDDR subsystem is 256 MB.

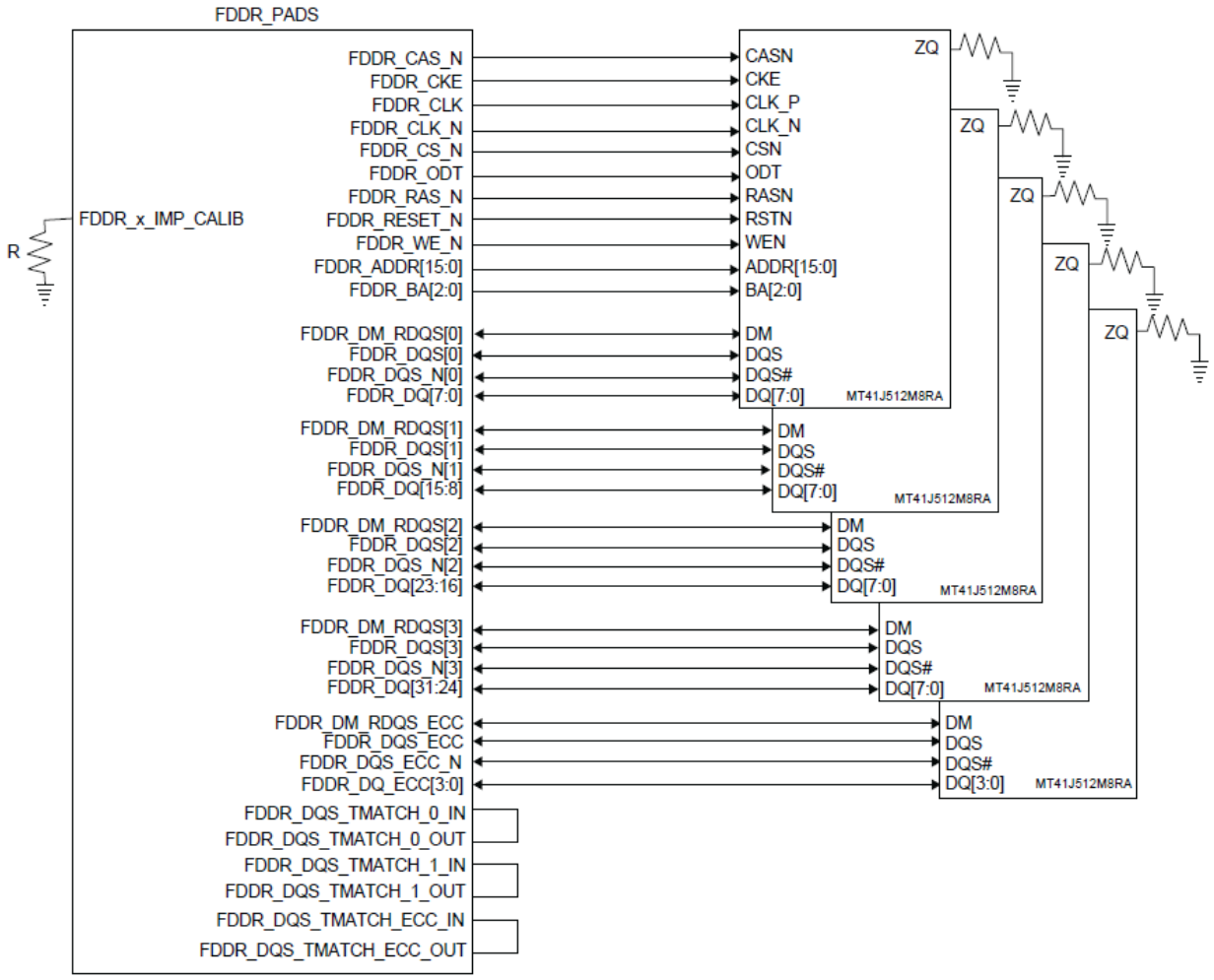
Figure 1-26. 32 Bit DDR2 SDRAM Connected to the FDDR Subsystem



1.10.2 Example 2: Connecting 32 Bit DDR3 to FDDR_PADs with SECDED (Ask a Question)

The following figure shows the DDR3 SDRAM connected to the FDDR subsystem of an RTG4 device. Micron's MT41J512M8RA is a 512 MB density device with x8 data width. The FDDR subsystem is configured in Full Bus Width mode with SECDED enabled. The SDRAM connected to FDDR_DQ_ECC[3:0] is used to store the SECDED bits. The total DDR3 memory, excluding memory for SECDED, connected to the FDDR subsystem is 2 GB.

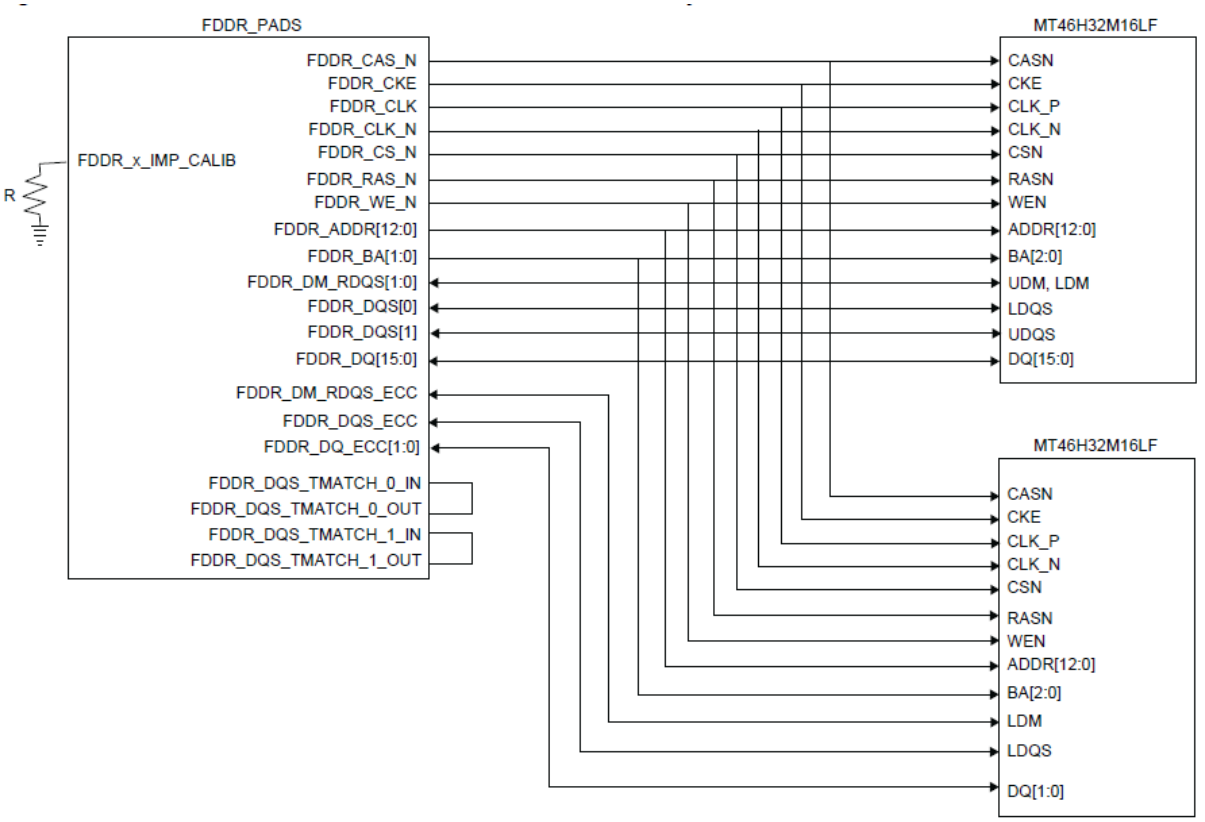
Figure 1-27. 32 Bit DDR3 SDRAM Connection to the FDDR Subsystem with SECDED



1.10.3 Example 3: Connecting 16 Bit LPDDR to FDDR_PADs with SECDED (Ask a Question)

The following figure shows the LPDDR1 SDRAM connected to the FDDR subsystem of an RTG4 device. Micron's MT46H32M16LF is a 64 MB density device with x16 data width. The FDDR subsystem is configured in Full Bus Width mode with SECDED enabled. The SDRAM connected to FDDR_DQ_ECC[1:0] is used to store the SECDED bits. The total amount of LPDDR1 memory, excluding memory for SECDED, connected to the FDDR subsystem is 64 MB.

Figure 1-28. x16 LPDR1 SDRAM Connection to the FDDR Subsystem



1.11 Implementing and Configuring FDDR Subsystem (Ask a Question)

This section describes how to use the DDR subsystem in the design, select memory devices, and configure the subsystem to interface with specific devices.

1.11.1 Selecting the Memory Device (Ask a Question)

The RTG4 FDDR subsystem supports DDR2, DDR3, and LPDDR memory devices. The memory that is used with the system depends on the application requirements. No single memory device can fulfill all system requirements; therefore, the trade-offs between each memory types must be considered for a balanced design.

Consider the following factors while selecting the memory device:

- Cost
- Bandwidth and speed
- Power consumption
- Bus width
- Latency
- Capacity

The following table lists the memory parameters for DDR2, DDR3, and LPDDR memory devices. Based on the system requirements (that is, bandwidth and speed, bus width, density, and cost), the approximate memory device is selected.

Table 1-11. Memory Selection Parameters

Parameter	DDR2	DDR3	LPDDR
Microchip supported Maximum data-rate	333 MHz 667 Mbps	333 MHz 667 Mbps	133 MHz 266 Mbps
Data width	8, 16, 32	8, 16, 32	8, 16, 32
Burst length	4, 8	4, 8	4, 8, 16
Number of banks	8	8	4
Density	512 Mb - 8192 Mb	512 Mb - 8192 Mb	64 Mb - 8192 Mb
I/O standard	SSTL-18 Class I, II	SSTL-15 Class I, II	LVCMOS18

1.11.2 Configuring the DDR Subsystem (Ask a Question)

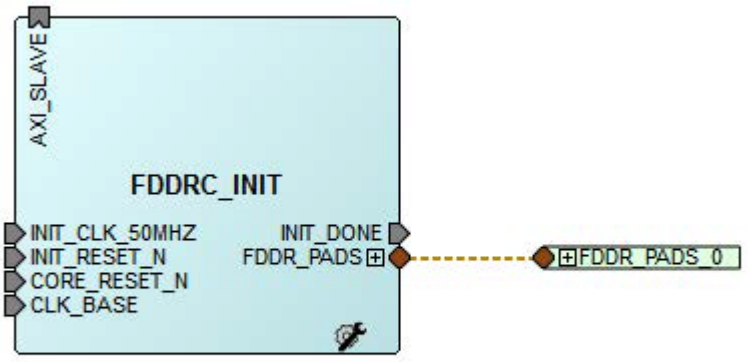
This section describes the FDDR Controller Configurator settings and how to use the FDDR controller to access the external DDR memory in RTG4 devices.

The core RTG4 DDR memory controller with initialization is used to initialize and configure the external DDR memory parameters for the fabric DDR controller.

To configure the RTG4 DDR memory controller with an initialization core:

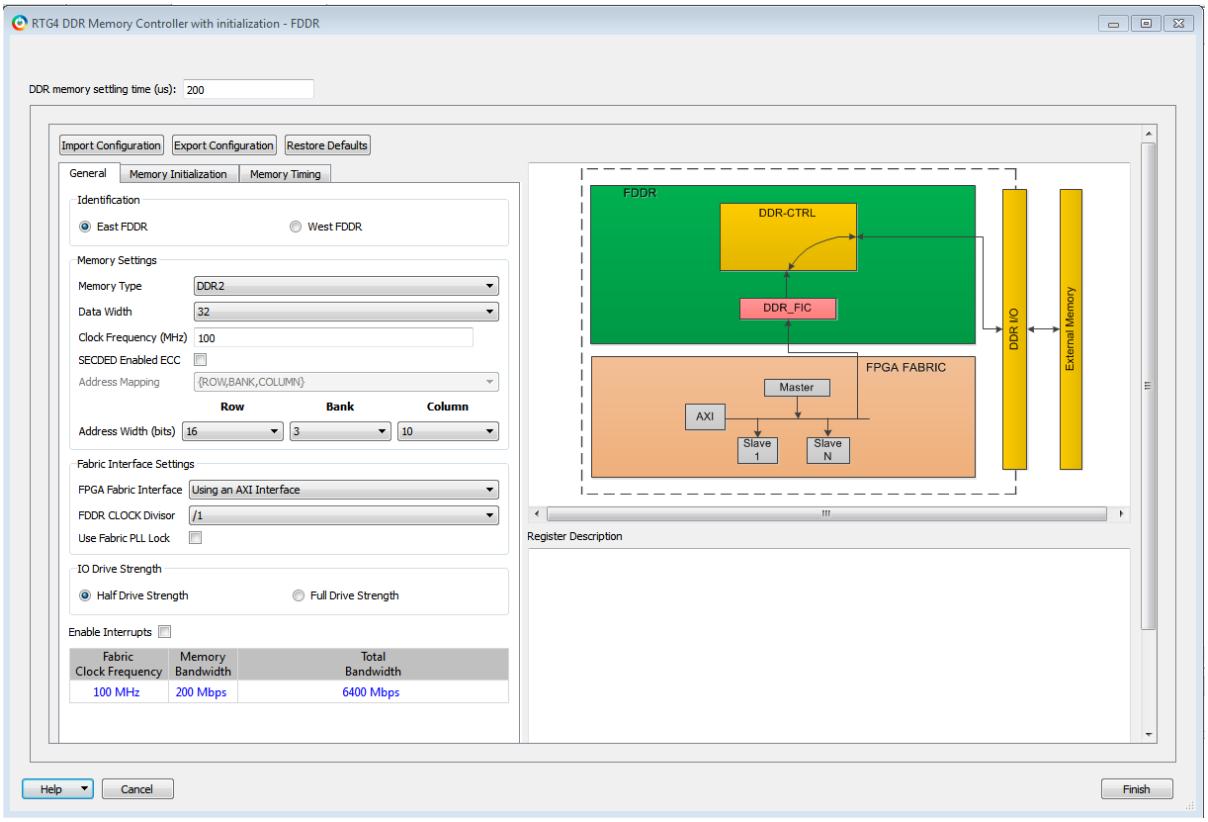
1. Double-click **Create SmartDesign** in the design flow tab of Libero and drag **RTG4 DDR Memory Controller with initialization core** from the **IP Catalog** onto the SmartDesign canvas.
2. Enter the component name when it prompts and click **OK**. The following figure shows the SmartDesign canvas with the instantiated component.

Figure 1-29. RTG4 DDR Memory Controller with Initialization Core



3. Double-click the **IP core** and open the **RTG4 DDR Memory Controller with initialization - FDDR** window.

Figure 1-30. FDDR Configurator Overview



4. In the **General** tab, select the **Memory Settings**, as shown in the preceding figure.
   - **Identification**: East FDDR or West FDDR
   - **Memory Type**: LPDDR, DDR2, or DDR3.
   - **Data Width**: 32, 16, or 8
   - **Clock Frequency**: Any value (decimal or fraction) between 20 MHz to 333 MHz
   - **SECDED Enabled ECC**: Select or Clear
   - **Address Width (bits)**: Register settings to map to system address bits for various row, bank, and column combinations are automatically computed by the Configurator using address mapping. The following table shows the software supported range for row, bank, and columns.

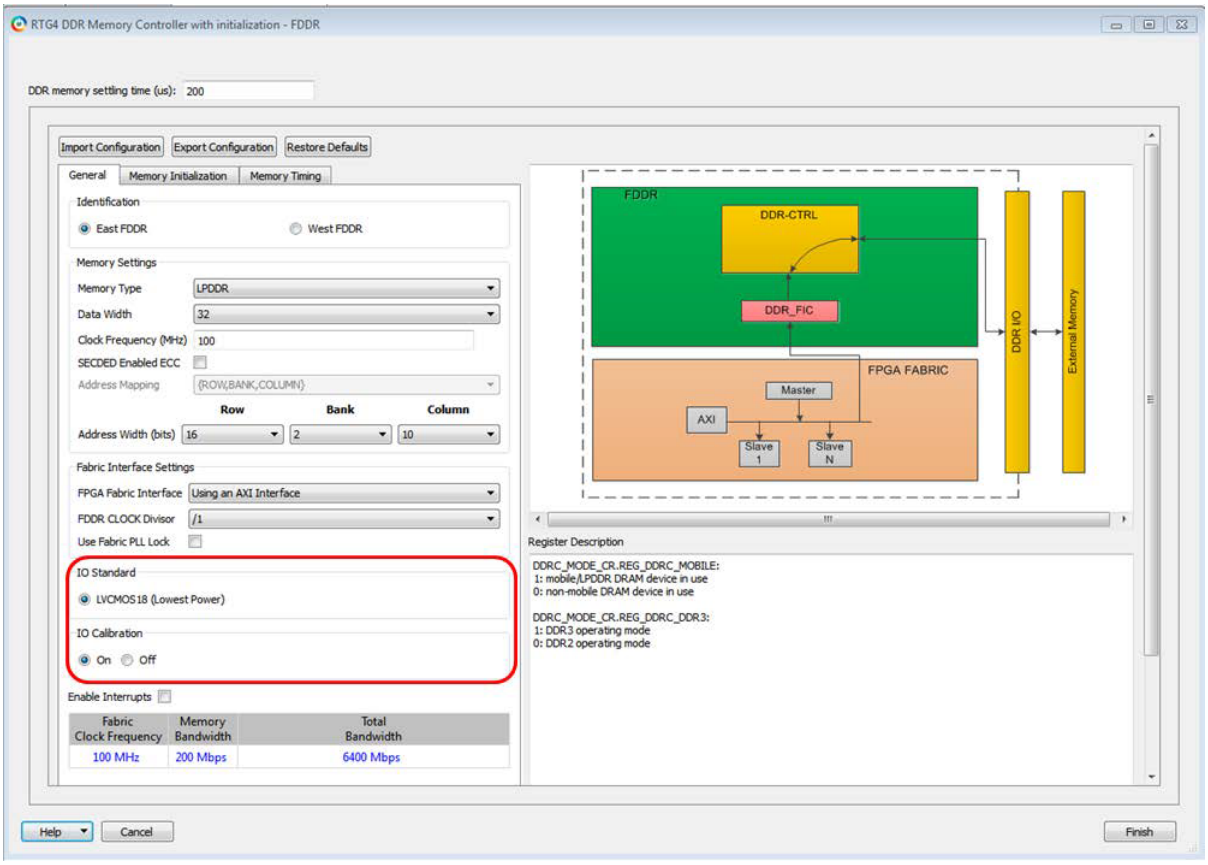
Table 1-12. Supported Address Width Range for Row, Bank, and Column Addressing in DDR/LPDDR

Address Width For	DDR2	DDR3	LPDDR
Row	12-16	12-16	12-16
Bank	2-3	2-3	2-3
Column	9-12	9-12	9-12

Use the settings in the following figure when you select LPDDR as the **Memory Type**:

- **I/O Standard**: LVCMOS18 (Lowest power), which is for LVCMOS1.8V I/O standard
- **I/O Calibration**: ON or OFF

Figure 1-31. Selecting I/O Standard and I/O Calibration for LPDDR



Important: Calibration ON and OFF provide different values for PCODE and NCODE registers. The I/O calibration block calibrates the I/O drivers to an external resistor. The impedance control is used to identify the digital values PCODE<5:0> and NCODE<5:0>. These values are fed to the pull-up or pull-down reference network to match the impedance with an external resistor. After it matches the PCODE and NCODE register values, they are latched and sent to the drivers. This feature can be turned ON or OFF as required.

5. Select the following to configure the **Fabric Interface Settings**:
   - **FPGA Fabric Interface**: Data interface between the FDDR controller and the FPGA design. As FDDR is a memory controller, it is intended to be a target on an AXI or AHB bus. The initiator of the bus initiates bus transactions, which are interpreted by FDDR as memory transactions and communicated to the off-chip DDR memory.
     - **Using an AXI-64 Interface**: One initiator accesses the FDDR controller through a 64 bit AXI interface.
     - **Using a Single AHB-32 Interface**: One initiator accesses the FDDR controller through a single 32 bit AHB interface.
     - **Using Two AHB-32 Interfaces**: Two initiators access the FDDR controller using two 32 bit AHB interfaces.
   - **FPGA CLOCK Divisor**: Specifies the frequency ratio between the DDR controller clock (CLK_FDDR) and the clock controlling the fabric interface (CLK_DDR_FIC). The CLK_DDR_FIC frequency must not exceed 167 MHz and must be equal to that of the AHB/AXI subsystem

that is connected to the FDDR AHB/AXI bus interface. For example, if a DDR RAM runs at 200 MHz and the fabric/AXI subsystem runs at 100 MHz, select a divisor of 2.

- **Use Fabric PLL Lock**: If CLK_BASE is sourced from a fabric CCC, connect the fabric CCC LOCK output to the FDDR FAB_PLL_LOCK input. CLK_BASE is not stable until the fabric CCC locks. Microchip recommends holding the FDDR controller in reset until CLK_BASE is stable, that is, assert the CORE_RESET_N input. The LOCK output of the fabric CCC indicates that the fabric CCC output clocks are stable. By selecting Use FAB_PLL_LOCK input port of the FDDR can be exposed, and then the LOCK output of the fabric CCC can be connected to the FAB_PLL_LOCK input of the FDDR controller.

6. Select one of the following to set the **IO Drive Strength** (for DDR2 and DDR3 only):
   - Half Drive Strength
   - Full Drive Strength

Tip: Selecting reduced drive strength reduces all outputs to approximately 45% to 60%. This is intended to support lighter loads.

Depending on the DDR memory type and the I/O strength selected, the Libero SoC software sets the DDR I/O standard for the FDDR system. The following table lists the I/O standards.

Table 1-13. I/O Standards

DDR Memory Type	Half Drive Strength	Full Drive Strength
DDR3	SSTL 15I	SSTL 15II
DDR2	SSTL 18I	SSTL 18II

7. Select/clear **Enable Interrupts**.
   The FDDR controller raises interrupts when predefined conditions are satisfied. This exposes the interrupt signals on the FDDR instance. Connect these interrupt signals according to the design requirement. The following are the available interrupt signals and preconditions:
   - **FIC_INT**: Generated when there is an error in the transaction between the initiator and the FDDR controller.
   - **IO_CAL_INT**: DDR controller allows you to recalibrate DDR I/O by writing into the FDDR_INTERRUPT_ENABLE SYSREG Configuration register through the APB configuration interface. When calibration is complete, this interrupt is raised.
   - **PLL_LOCK_INT**: Indicates that the FDDR FPLL is locked.
   - **PLL_LOCKLOST_INT**: Indicates that the FDDR FPLL has lost lock.
   - **FDDR_ECC_INT**: Indicates that a single or two-bit error is detected.
8. Enter **Fabric Clock Frequency**.
   Fabric clock frequency (CLK_BASE) is calculated using the DDR controller clock (CLK_FDDR) frequency and the FDDR clock divisor value.
   Fabric clock (CLK_BASE) Frequency (in MHz) = (CLK_FDDR clock frequency)/(FDDR clock divisor)
9. Enter **Memory Bandwidth**.
   Memory bandwidth is calculated using current clock frequency value in Mbps.
   Memory bandwidth (Mbps) = 2 × Clock Frequency
10. Enter **Total Bandwidth**.
    Total bandwidth calculation is based on current fabric clock frequency (CLK_BASE), DDR, data width, and FDDR clock divisor, in Mbps.
    Total Bandwidth (in Mbps) = (2 × Fabric Clock Frequency × DDR Data Width)/(FDDR CLOCK Divisor)

The FDDRC_INIT block is hard coded with a DDR memory settling time of 200 µs, assuming the clock period of INIT_CLK is 20 ns (frequency 50 MHz), as shown in Figure 1-29.

Tip: Microchip recommends that the initialization frequency must be kept at 50 MHz.

11. In the **Memory Initialization** tab, select the required memory initialization settings, as shown in the following figure.
    - Select the following to configure the **Performance Settings**:
      - **Burst Length**: 4, 8, or 16. For supported burst lengths, see the following table.
      - **Burst Order**: Sequential or Interleaved.
      - **Timing Mode**: 1T or 2T. For more information, see 1.5.5. 1T or 2T Timing.
      - **CAS Latency**: Delay in clock cycles between the internal READ command and the availability of the first bit of output data. Select according to the DDR memory (mode register) datasheet.

      The following table lists the supported burst modes for DDR SDRAM types and PHY widths.

Table 1-14. Supported Burst Modes

Bus Width	Memory Type	Sequential and Interleaving Modes		
		4	8	16
32	LPDDR1	Applicable	Applicable	—
	DDR2			
	DDR3	—		
16	LPDDR1			Applicable
	DDR2			—
	DDR3			
8	LPDDR1			
	DDR3			
	DDR2			

    - Select the following to configure **Power-Saving Mode** settings:
      - Self refresh enabled
      - Auto refresh burst count
      - Powerdown enabled
      - Stop the clock (supports LPDDR only)
      - Deep powerdown enabled (supports LPDDR only)
      - Powerdown entry time

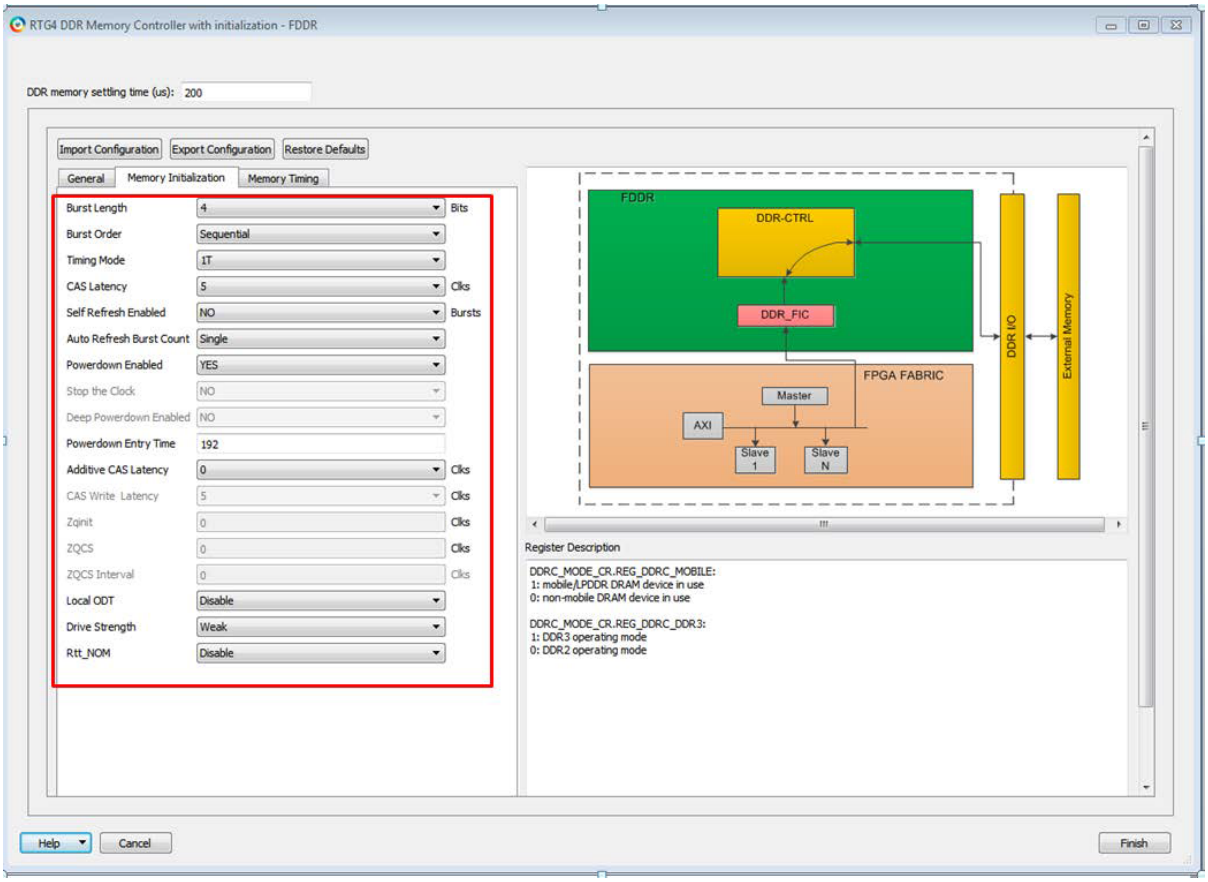
      For more information, see 1.15. Low Power Operating Options.
    - Select the following to configure **Additional Performance** settings:
      - **Additive CAS Latency**: Defined by the EMR[5:3] register of the DDR2 memory and the MR1[4:3] register of the DDR3 memory. It enables the DDR2 or DDR3 SDRAM to allow READ or WRITE command from the DDR controller after the ACTIVATE command for the same bank prior to $t_{RCD\ (MIN)}$. It is part of the DDR2 extended mode register and the DDR3 mode register1.

- **CAS Write Latency**: Defined by the DDR3 MR2[5:3] register. It is the delay in number of clock cycles from the release of the internal write to the latching of the first data in. The overall WRITE latency (WL) is equal to CWL + AL.
  Select the ZQ calibration settings (Zqinit, ZQCS, and ZQCS Interval) for the DDR3 memory. For more information, see 1.7.4. ZQ Calibration.
- The local ODT setting is not supported for LPDDR memory. For the DDR2 or DDR3 memory type, you can choose any option for Local ODT. You can enable or disable LOCAL ODT during read transaction.
- **Drive Strength**: Defined by the EMR[7:5] register bits of the LPDDR memory with full, half, quarter, and three by fourth drive strength options. The EMR[1] register bit of the DDR2 memory with full and reduced drive strength options, and MR1 register bits M5 and M1 of DDR3 memory with RZQ/6 and RZQ/7 options.
  Partial array self refresh coverage setting is defined by the EMR[2:0] register bits of LPDDR memory with full, quarter, one-eighth, and one-sixteenth options. This feature helps in improving power
  savings during self refresh by selecting the amount of memory to be refreshed during self refresh.
- **RTT_NOM**: Defined by the EMR[6] and EMR[2] register bits of the DDR2 memory, which determines what ODT resistance is enabled with RTT disabled, 50Ω, 75Ω, and 150Ω options. It is defined by MR1[9], MR1[6], and MR1[2] register bits of the DDR3 memory. In the DDR3 memory, RTT nominal termination is allowed during standby conditions and WRITE operations, and not during READ operations with RZQ/2, RZQ/4, and RZQ/6 options.
- **RTT_WR**: Dynamic ODT defined by the MR2[10:9] register bits of the DDR3 memory. This is
  applicable only during WRITE operations. If dynamic ODT (Rtt_WR) is enabled, DRAM switches from normal ODT (RTT_nom) to dynamic ODT (Rtt_WR) when beginning WRITE burst and
  subsequently switches back to normal ODT at the end of WRITE burst. Off, RZQ/4, and RZQ/2 options are available.

Auto self refresh setting is defined by the MR2[6] register bit of the DDR3 memory with manual and auto options.

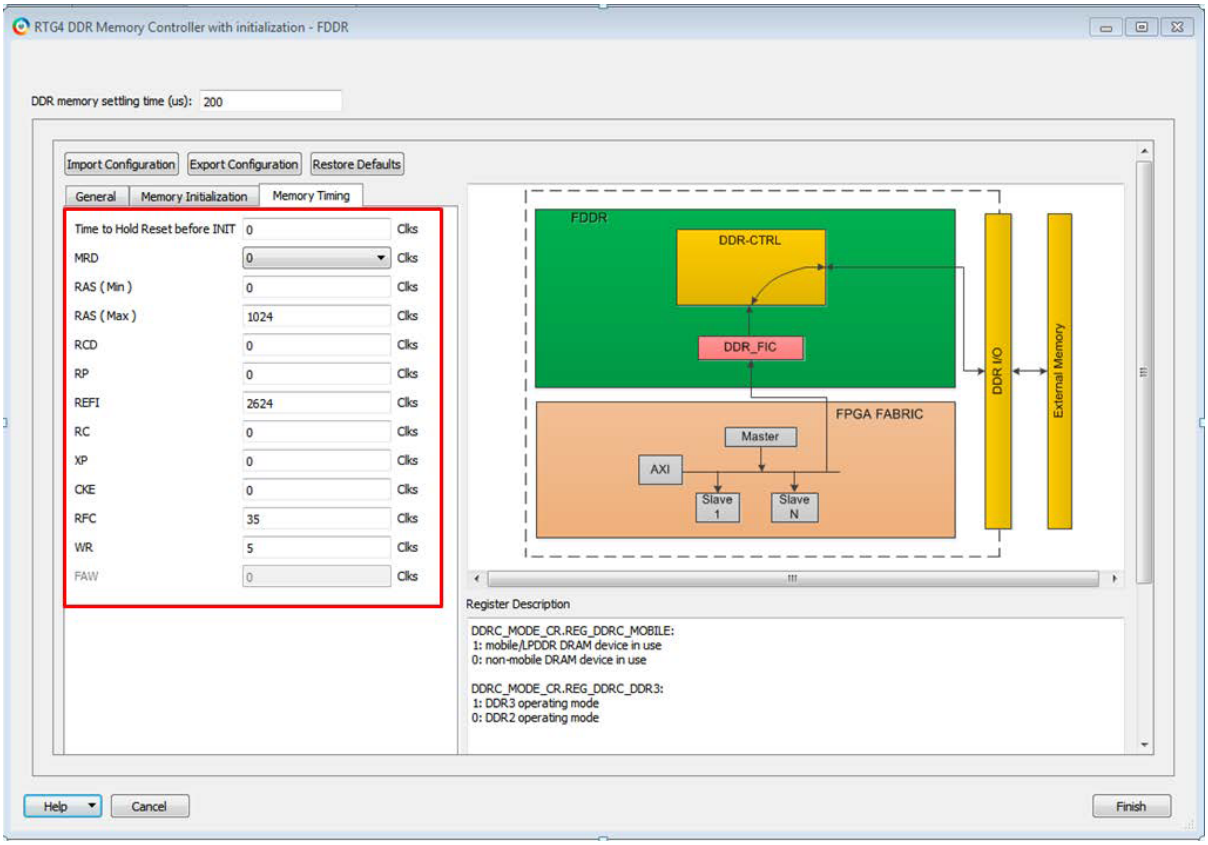
Self refresh temperature setting is defined by the MR2[7] register bit of the DDR2 memory with
normal and extended options.

Figure 1-32. Memory Initialization Setting Window



12. In the **Memory Timing** tab, select the memory timing settings according to the DDR memory vendor datasheet, as shown in the following figure. For more information, see 1.9. Functional Timing Diagrams.

Figure 1-33. Memory Timing Configuration



13. Click **Finish** to save the selected settings.
14. Click **OK** to save and close the **FDDR Controller Configurator** window.

The Configurator also provides the option to import and export the register configurations.

The following is a an example FDDR register configuration for operating the LPDDR memory (MT46H32M16LF) with clock 166 MHz:

- General
  - **Memory Type**: Select LPDDR
  - **Data Width**: 16
- Memory Initialization
  - **Burst length**: 8
  - **Burst Order**: Interleaved
  - **Timing Mode**: 1T
  - **CAS Latency**: 3
  - **Self Refresh Enabled**: No
  - **Auto Refresh Burst Count**: 8
  - **Power Down Enabled**: Yes
  - **Stop the clock**: No
  - **Deep Power Down enabled**: No
  - **Power down entry time**: 320
  - **Drive strength**: Full

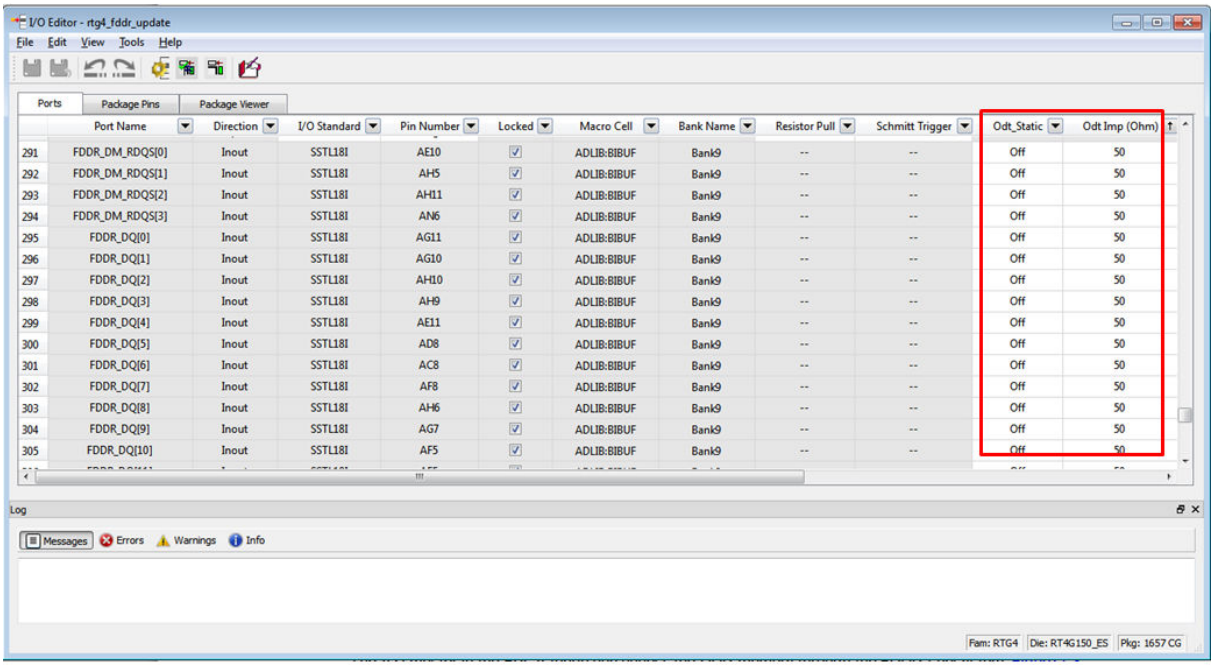
  - **Partial array self refresh**: Quarter array
- Memory Timing
  - **MRD**: 4 Clks
  - **RAS (Min)**: 8 Clks
  - **RAS (Max)**: 8192 Clks
  - **RCD**: 6 Clks
  - **RP**: 7 Clks
  - **REFI**: 3104 Clks
  - **RC**: 3 Clks
  - **XP**: 3 Clks
  - **CKE**: 3 Clks
  - **RFC**: 79 Clks
  - **FAW**: 0 Clks

1.11.2.1 I/O Configuration (Ask a Question)

The following steps describe how to configure I/O:

1. Double-click **Edit IO constraints** to open the **I/O Editor window**, as shown in the following figure.
2. Configure the **I/O settings** such as **ODT** and **drive strength**.

Figure 1-34. I/O Editor Window



1.12 Simulating FDDR Subsystem (Ask a Question)

The Libero SoC provides a complete simulation model of the FDDR subsystem. For simulating the FDDR subsystem, connect the FDDR DRAM interface to the DDR memory simulation model in a testbench. There are two types of memory simulation models that can be used:

- Microchip has provided Verification Intellectual Property (VIP): The VIP model is attached to the pin side of the FDDR subsystem and simulates the functionality of a DDR memory device. It

can be configured for the DDR2, DDR3, and LPDDR SDRAM memories and used to complement vendor models or to act as a substitute in case a vendor model is not available.

- Vendor-specific memory model: Memory vendors such as Micron, Samsung, and Hynix™ provide downloadable simulation models for specific memory devices. The downloaded simulation model must be JEDEC compliant.

Important: You are advised to use a vendor-specific memory model to simulate the target memory device connected to FDDR.

1.13 Functional Examples (Ask a Question)

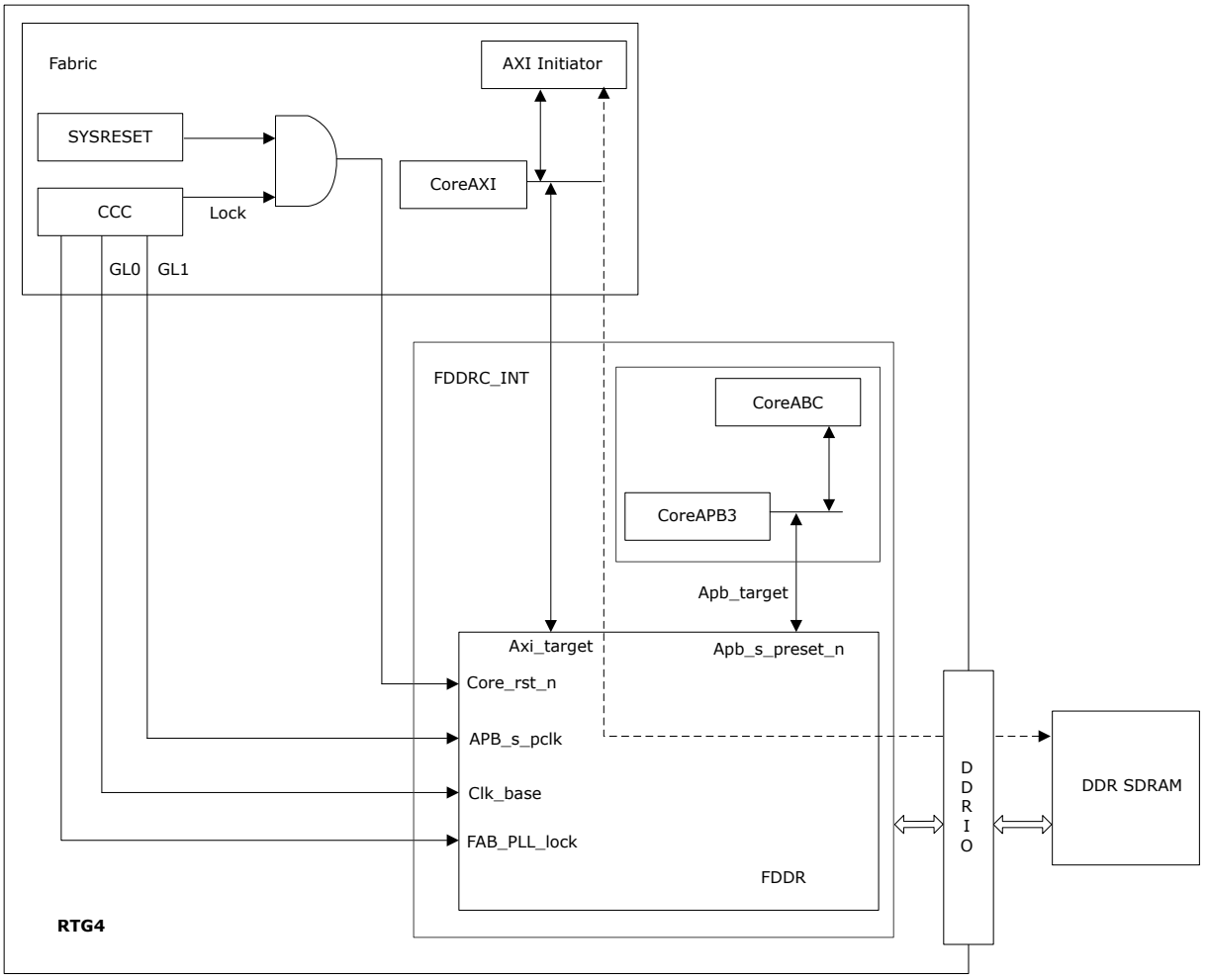
There are two functional examples:

- Accessing FDDR from FPGA Fabric through AXI Interface
- Accessing FDDR from FPGA Fabric through AHB Interface

1.13.1 Accessing FDDR from FPGA Fabric through AXI Interface (Ask a Question)

The AXI initiator in the FPGA fabric can access the DDR memory through the FDDR subsystem. The following figure shows the FDDR controller with the AXI interface. The FDDR registers are configured from the FPGA fabric using the CoreABC interface.

Figure 1-35. FDDR Subsystem with AXI Interface

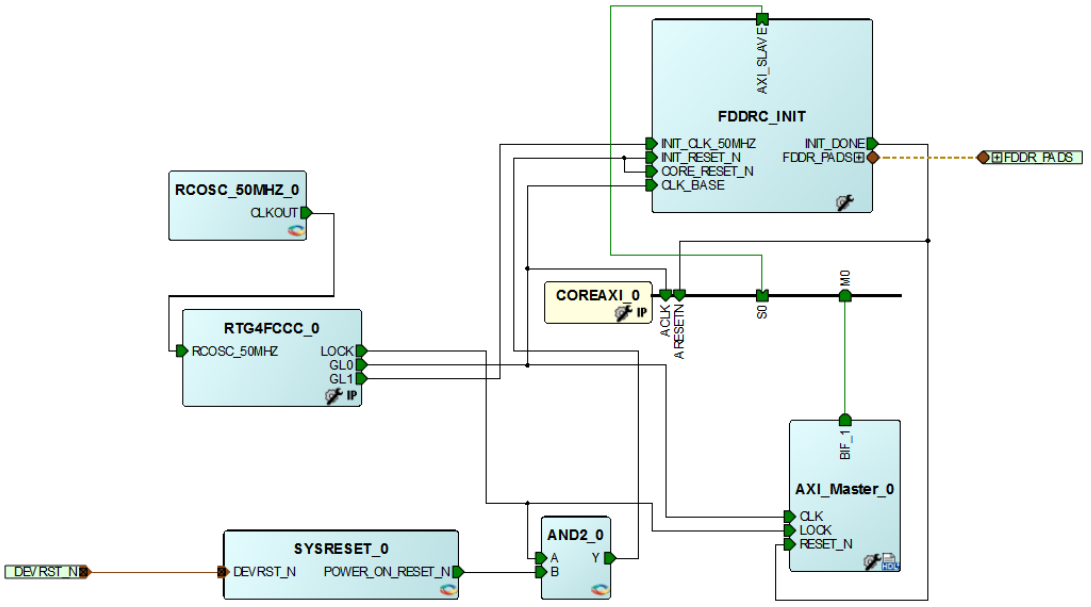


Read and write transactions are initiated by the AXI initiator to read from or write to the DDR memory after initializing the FDDR registers.

The following steps describe how to access the FDDR controller from the AXI initiator in the FPGA fabric:

- Create **SmartDesign** in the **Design Flow** tab of Libero and drag RTG4 DDR Memory Controller with initialization core from the **IP Catalog** onto the SmartDesign canvas.
- Enter the component name when it prompts and click **OK**. Figure 1-36 shows the SmartDesign canvas with the instantiated component.
- Double-click the core to open the **FDDR Configurator** window.
- Add the configuration settings and click **OK** to exit the **Configurator** window.
- Use on-chip 25/50 MHz RC oscillator as source for fabric CCC.
- Configure FDDR_CLK as 333 MHz and FDDR_SUBSYSTEM_CLK (CLK_BASE) as FDDR_CLK/3. The FDDR controller clock must be driven from the fabric CCC as an output.
- Instantiate the user AXI initiator logic in the SmartDesign canvas to access the FDDR through the AXI interface. Ensure that the AXI initiator logic accesses the FDDR after INIT_DONE is high.
- Instantiate CoreAXI and configure.
- Instantiate the CCC block in the SmartDesign canvas and configure it to generate 111 MHz for the FDDR_SUBSYSTEM_CLK (CLK_BASE).
- Connect the following AXI_Master logic signals:
  - CLK to GL0 of RTG4FCCC_0 and FDDRC_INIT CLK_BASE
  - LOCK of AXI_Master_0 to LOCK of RTG4_FCCC_0
  - RESET_N to INIT_DONE of FDDRC_INIT
  - AXI_SLAVE of FDDRC_INIT to target (mirrored) BIF of AXI bus when the FDDR is configured as an AXI target.

Figure 1-36. RTG4 SmartDesign Connections (Top-Level View)

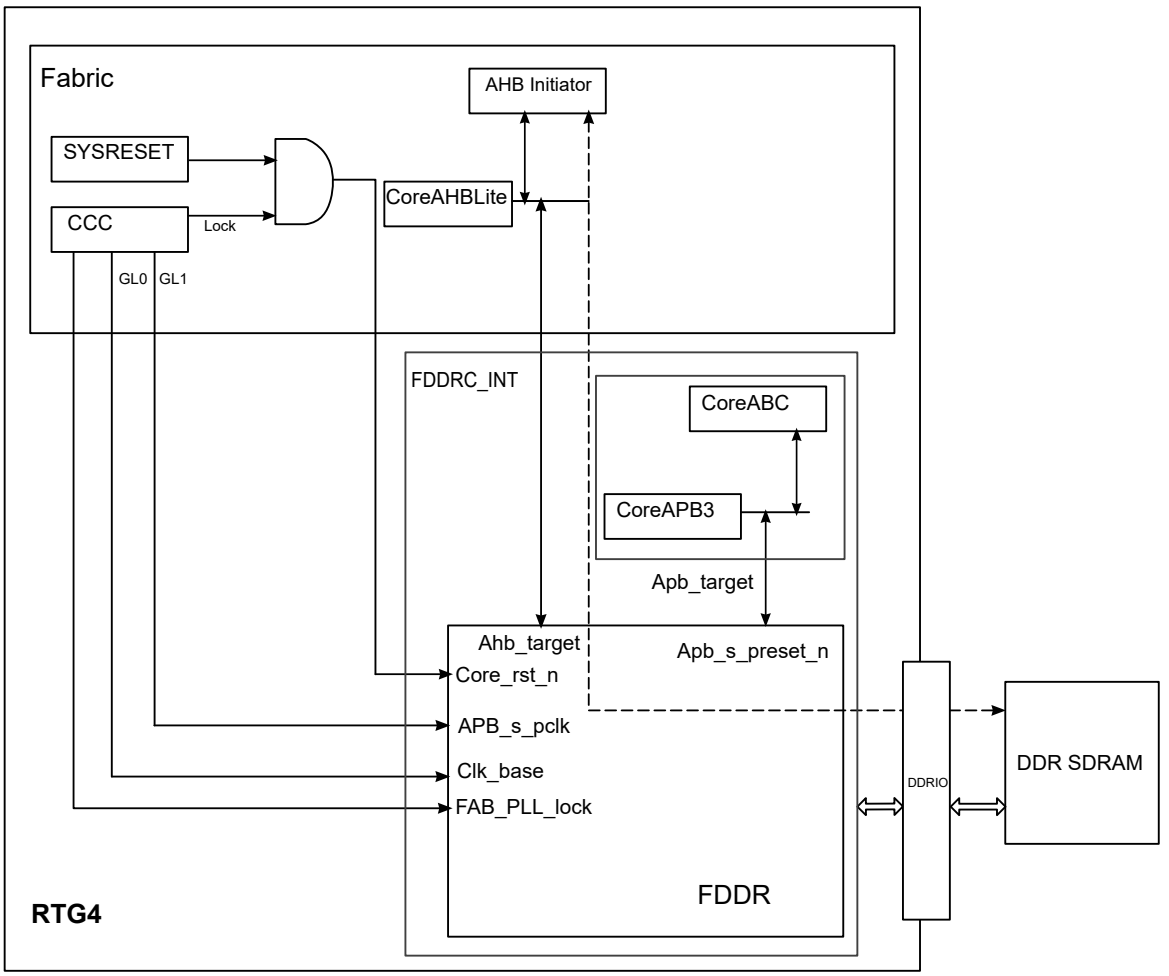


Tip: For FDDR AXI throughput, see AC446: Optimization Techniques to Improve DDR Throughput for RTG4 Devices Application Note.

1.13.2 Accessing FDDR from FPGA Fabric through AHB Interface (Ask a Question)

The FDDR subsystem can be used to access the DDR memory using the AHB-Lite interface. The following figure shows the FDDR with AHB-Lite interface.

Figure 1-37. FDDR Subsystem with AHB Interface



The steps for accessing the FDDR controller from the AHB initiator in the FPGA fabric are same as in 1.13.1. Accessing FDDR from FPGA Fabric through AXI Interface, except using the AHB initiator user interface and Core AHB-Lite soft IP for AHB bus interface.

The following table lists the FDDR throughput for the following FDDR configuration:

- Fabric Interface: AHB
- FDDR Mode: DDR3
- Fabric Clock to FDDR Clock Ratio: 1:4
- PHY Width: 16 and 32

- Clock Frequency: 80 MHz

The other parameters are configured similar to the FDDR configuration in AC446: Optimization Techniques to Improve DDR Throughput for RTG4 Devices Application Note.

Table 1-15. FDDR Throughput (for AHB)

FDDR-Fabric Interface-Memory	Frequency Ratio (CLK_BASE:FDDR_CLK)	PHY Width	Write Transaction BW (MB/sec)	Read Transaction BW (MB/sec)
FDDR_AHB-DDR3	1:4 80 MHz:320 MHz	PHY_16	80 MB	79 MB
		PHY_32	80 MB	79 MB

1.14 Synthesis Recommendations, Timing Constraints, and Floor Plan Constraints (Ask a Question)

The RTG4 FDDR subsystem is a hardened ASIC block and does not require any additional constraints.

1.15 Low Power Operating Options (Ask a Question)

The DDR controller can operate DDR memories in three power-saving modes:

- Precharge power-down (DDR2, DDR3, and LPDDR1)
- Self refresh (DDR2, DDR3, and LPDDR1)
- Deep power-down (LPDDR1)

1.15.1 Precharge Power-Down (Ask a Question)

The DDR2, DDR3, and LPDDR1 memories support precharge power-down mode. Entering precharge power-down deactivates the input buffers of DDR memory (excluding CK, CK#, CKE, RESET#, and ODT), for maximum power savings while in standby.

The DDR controller automatically keeps the DDR memory in precharge power-down mode when:

- REG_DDRC_POWERDOWN_EN = 1.
- The period specified by REG_DDRC_POWERDOWN_TO_X32 register has passed. During this time, the controller is in Idle state except for issuing refreshes.

While devices are in power-down mode, the DDR controller core continues to issue refresh commands to devices to maintain data integrity. During a refresh sequence, the controller takes all ranks out of power-down, issues a precharge-all command (if required), issues the refresh command, and then returns ranks to the power-down state.

The DDR controller automatically exits the precharge power-down on any of the following conditions:

- A refresh cycle is required to any rank in the system.
- The controller receives a new request from the core logic.
- REG_DDRC_POWERDOWN_EN register bit is set to 0.

1.15.2 Self Refresh (Ask a Question)

The DDR2, DDR3, and LPDDR1 memories support self refresh mode. The Self refresh mode is used to retain data in the DRAM, even if the rest of the system is powered down. When in self refresh mode, the DRAM retains data without external clocking. Entering self refresh deactivates the input buffers of DDR memory, excluding CKE and RESET#.

The DDR controller keeps the DDR memory devices in self refresh mode whenever the REG_DDRC_SELFREF_EN register bit is set and no reads or writes are pending in the controller.

The controller takes the DDR memory out of self refresh mode whenever the REG_DDRC_SELFREF_EN register bit input is deasserted or the controller receives new commands.

When the DDR self refresh is enabled, the DDR I/O bank might go into recalibration and a glitch might occur in the MDDR bank I/Os, which are being used for general purpose rather than for the DDR memory. The DDR I/Os ODT is periodically calibrated for PVT changes and are affected only when the I/Os are in tri-state (DDR I/Os are tri-stated only in self refresh mode).

1.15.3 Deep Power-Down (Ask a Question)

The LPDDR1 memories support deep power-down mode. Deep power-down mode achieves the maximum power reduction by eliminating the power of the memory array. Data is not retained after the device enters deep power-down mode.

The DDR controller puts the DDR SDRAM devices in deep power-down mode whenever the REG_DDRC_POWERDOWN_EN bit is set and no reads or writes are pending in the DDR controller.

The DDR controller automatically exits the deep power-down mode and reruns to the initialization sequence when the REG_DDRC_POWERDOWN_EN bit is reset to 0.

1.16 FDDR Subsystem Performance Optimization (Ask a Question)

The DDR memory connected to the FDDR subsystem is accessed by the AXI/AHB initiator logic implemented in the FPGA fabric. Throughput of a DDR depends on various parameters. In this section, optimization techniques that improve the efficiency and provide a better throughput are described.

Following are the optimization techniques:

- AXI initiator interface
- Maximum DDR memory burst length
- AXI initiator without write response state
- Read address queuing
- Series of writes or reads
- DDR configuration tuning

See the AC446: Optimization Techniques to Improve DDR Throughput for RTG4 Devices Application Note for more details on throughput optimization techniques.

1.17 Printed Circuit Board Guidelines (Ask a Question)

The DDR SDRAM memories are extensively used in networking, telecom, and embedded applications where high performance is an important design requirement. Although, it plays a major role to improve the overall system performance, good board design practice must be followed in the schematic and layout phases to achieve the expected performance. High quality and reliable results depend on minimizing noise levels, preserving signal integrity, and meeting impedance and power requirements.

It is important to understand the key factors that affect the DDR memory performances, which are as follows:

- Placement requirements
- Routing requirements
- Stack-up
- Power supply and decoupling
- Dynamic ODT
- Proper use of termination
- I/O drive strength setting

See the AC439: RTG4 FPGAs Board Design and Layout Guidelines to understand these key factors in detail. It is also important to understand the trade-offs between these key factors to select the most suitable settings for the design.

1.18 Debugging Techniques (Ask a Question)

Although, the external memory controller eliminates the design complexity associated with the DDR SDRAM interface design, debugging is a real challenge. Tracking down issues such as functional issues (at the FPGA or system level), functional system interaction problems, system timing issues, and signal fidelity issues between FPGA and memory devices (such as noise, crosstalk, or reflections) become much more complex. Therefore, care must be taken from the external memory controller configuration phase to the PCB layout phase to achieve the expected performance. This section describes the tools and process for the external memory controller interface.

This section contains the following sub-sections:

- Debug Tools
- Hardware Debug

1.18.1 Debug Tools (Ask a Question)

There are many tools available to debug external memory controller issues. This section provides an overview of each tool.

1.18.1.1 Identify RTL Debugger (Ask a Question)

Identify is an RTL Debugger that allows you to probe interface signals of the external memory controller AXI/AHB interface and view those signals directly from the programmed FPGA in the original RTL code or in a waveform viewer.

Identify consists of two separate tools:

- Instrumentor: The Instrumentor Compiles and inserts the necessary logic into the original RTL to create an instrumented design that allows probing of internal signals.
- Debugger: The Debugger Communicates to the FlashPro device via JTAG to get the values for the probed signals directly from the programmed FPGA and displays those signals.

1.18.1.2 Hardware Probes (Ask a Question)

RTG4 devices have built-in probe points that greatly enhance the ability to debug logic elements within the device through the following features:

- Live Probes: A dedicated probe can be configured to observe a probe point, which is any output of a register. After configuring, the probe data can be sent to the dedicated pin, and then to an oscilloscope.
- Active Probes: Active probes allow dynamic asynchronous read and write to a flip-flop or probe point. This capability enables quick observation of the output of the logic internally or quick experimenting on how the logic is affected by writing to a probe point.

The enhanced debug features implemented into the devices give access to any logic element and enable real-time monitoring of inputs and outputs, without the need to recompile the design. In many cases, the hardware probe capabilities such as live probe can be used in conjunction with the Identify RTL Debugger and the external test techniques to debug the issues in the subsystem.

1.18.2 Hardware Debug (Ask a Question)

Good board design practice must be followed in the schematic and layout phases to achieve the expected performance. High quality and reliable results depend on minimizing noise levels, preserving signal integrity, and meeting impedance, and power requirements. Therefore, ensure that all guidelines provided in AC439: RTG4 FPGAs Board Design and Layout Guidelines are strictly followed. This section details the list of general checks.

- Verify the parameters (timing, signal width, memory parts, and so on) entered in the Configurator are correct.
- Create a simple design that replicates the issue.
- Ensure that all voltages on the board are correctly set and noise is within the limit.
- Ensure that all terminations are adequate.
- Ensure that clock and reset signals are clean and correct.
- Perform the signal integrity analysis. The important things must be observed from the results are:
  - Setup and hold time between data signals (DQ) and the respective DQS over all corners.
  - Setup and hold time between Control/Command/Address signals and the clock over all corners.
  - Overshoot and undershoot of all signals with respect to JEDEC specifications over all corners.
  - DC threshold multi-crossing that is due to the excessive ringing.
- Cool and heat the device to understand whether the temperature variation is affecting the functionality. The device runs faster and makes timing easier to meet when it is cooled. Whereas, heating the device makes the device run slower and makes timing more difficult to meet. If this experiment resolves the issue, then the problem could be due to timing.
- Run at a lower speed. Ensure that DDR clock frequency is set within the minimal operating frequency of the device.
- Isolate the issue due to the Libero SoC software and the DDR Controller IP versions.
- Try it on multiple boards of same version.

2. FDDR Configuration Registers (Ask a Question)

This section provides FDDR subsystem registers along with the address offset, functionality, and bit definitions. The registers are categorized based on the controller blocks in the FDDR subsystem.

The following table lists the categories of registers and their offset addresses. If the address locations (between 0x000 to 0x3FC) not allocated for registers are accessed, then an error response is generated on APB PSLVERR. The reads to the address locations (between 0x400 to 0x7FC) not allocated for registers return an unknown value. The writes to the address locations (between 0x400 to 0x7FC) not allocated for registers are ignored.

Table 2-1. Address Table for Register Interfaces

Registers	Address Offset Space
DDR Controller Configuration Register	0x000:0x1FC
Reserved	0x200:0x3FC
DDR_FIC Configuration Register Summary	0x400:0x4FC
FDDR SYSREG Configuration Register Summary	0x500:0x5FC
Reserved	0x500:0x7FC

2.1 DDR Controller Configuration Register Summary (Ask a Question)

The following table lists the DDR controller configuration registers.

Table 2-2. DDR Controller Configuration Register

Register Name	Address Offset	Register Type	Reset Source	Description
DDRC_DYN_SOFT_RESET_CR	0x000	RW/RO	PRESET_N	DDRC Reset register
	0x004	RW	PRESET_N	Exclusive AXI initiator ID Configuration register. The configured four fabric AXI initiator IDS support exclusive accesses.
DDRC_DYN_REFRESH_1_CR	0x008	RW	PRESET_N	DDRC Refresh Control register
DDRC_DYN_REFRESH_2_CR	0x00C	RW	PRESET_N	DDRC Refresh Control register
DDRC_DYN_POWERDOWN_CR	0x010	RW	PRESET_N	DDRC Power-Down Control register
Reserved	0x014	—	—	—
DDRC_MODE_CR	0x018	RW	PRESET_N	DDRC Mode register
DDRC_ADDR_MAP_BANK_CR	0x01C	RW	PRESET_N	DDRC Bank Address Map register
Reserved	0x020	—	—	—
DDRC_ADDR_MAP_COL_1_CR	0x024	RW	PRESET_N	DDRC Column Address Map register
DDRC_ADDR_MAP_COL_2_CR	0x028	RW	PRESET_N	DDRC Column Address Map register
DDRC_ADDR_MAP_ROW_1_CR	0x02C	RW	PRESET_N	DDRC Row Address Map register
DDRC_ADDR_MAP_ROW_2_CR	0x030	RW	PRESET_N	DDRC Row Address Map register
DDRC_INIT_1_CR	0x034	RW	PRESET_N	DDRC Initialization Control register
DDRC_CKE_RSTN_CYCLES_1_CR	0x038	RW	PRESET_N	DDRC Initialization Control register
DDRC_ CKE_RSTN_CYCLES_2_CR	0x03C	RW	PRESET_N	DDRC Initialization Control register
DDRC_INIT_MR_CR	0x040	RW	PRESET_N	DDRC MR Initialization register
DDRC_INIT_EMR_CR	0x044	RW	PRESET_N	DDRC EMR Initialization register

.....continued

Register Name	Address Offset	Register Type	Reset Source	Description
DDRC_INIT_EMR2_CR	0x048	RW	PRESET_N	DDRC EMR2 Initialization register
DDRC_INIT_EMR3_CR	0x04C	RW	PRESET_N	DDRC EMR3 Initialization register
DDRC_DRAM_BANK_TIMING_PARAM_CR	0x050	RW	PRESET_N	DDRC DRAM Bank Timing Parameter register
DDRC_DRAM_RD_WR_LATENCY_CR	0x054	RW	PRESET_N	DDRC DRAM Write Latency register
DDRC_DRAM_RD_WR_PRE_CR	0x058	RW	PRESET_N	DDRC DRAM Read-Write Pre-charge Timing register
DDRC_DRAM_MR_TIMING_PARAM_CR	0x05C	RW	PRESET_N	DDRC DRAM Mode Register Timing Parameter register
DDRC_DRAM_RAS_TIMING_CR	0x060	RW	PRESET_N	DDRC DRAM RAS Timing Parameter register
DDRC_DRAM_RD_WR_TRNARND_TIME_CR	0x064	RW	PRESET_N	DDRC DRAM Read Write Turn-around Timing register
DDRC_DRAM_T_PD_CR	0x068	RW	PRESET_N	DDRC DRAM Power-Down Parameter register
DDRC_DRAM_BANK_ACT_TIMING_CR	0x06C	RW	PRESET_N	DDRC DRAM Bank Activate Timing Parameter register
DDRC_ODT_PARAM_1_CR	0x070	RW	PRESET_N	DDRC ODT Delay Control register
DDRC_ODT_PARAM_2_CR	0x074	RW	PRESET_N	DDRC ODT Hold/Block cycles register
DDRC_ADDR_MAP_COL_3_CR	0x078	RW	PRESET_N	Upper byte is DDRC Column Address Map register and lower byte controls debug features.
DDRC_MODE_REG_RD_WR_CR	0x07C	RW	PRESET_N	DDRC Mode Register Read/ Write Command register
DDRC_MODE_REG_DATA_CR	0x080	RW	PRESET_N	DDRC Mode Register Write Data Register
DDRC_PWR_SAVE_1_CR	0x084	RW	PRESET_N	DDRC Power Save register
DDRC_PWR_SAVE_2_CR	0x088	RW	PRESET_N	DDRC Power Save register
DDRC_ZQ_LONG_TIME_CR	0x08C	RW	PRESET_N	DDRC ZQ Long Time Calibration register
DDRC_ZQ_SHORT_TIME_CR	0x090	RW	PRESET_N	DDRC ZQ Short Time Calibration register
DDRC_ZQ_SHORT_INT_REFRESH_MARGIN_1_CR	0x094	RW	PRESET_N	DDRC ZQ Short Time Calibration register
DDRC_ZQ_SHORT_INT_REFRESH_MARGIN_2_CR	0x098	RW	PRESET_N	DDRC ZQ Short Time Calibration register
DDRC_PERF_PARAM_1_CR	0x09C	RW	PRESET_N	DDRC Performance Parameter register
DDRC_HPR_QUEUE_PARAM_1_CR	0x0A0	RW	PRESET_N	DDRC Performance Parameter register
DDRC_HPR_QUEUE_PARAM_2_CR	0x0A4	RW	PRESET_N	DDRC Performance Parameter register
DDRC_LPR_QUEUE_PARAM_1_CR	0x0A8	RW	PRESET_N	DDRC Performance Parameter register
DDRC_LPR_QUEUE_PARAM_2_CR	0x0AC	RW	PRESET_N	DDRC Performance Parameter register
DDRC_WR_QUEUE_PARAM_CR	0x0B0	RW	PRESET_N	DDRC Performance Parameter register

.....continued

Register Name	Address Offset	Register Type	Reset Source	Description
DDRC_PERF_PARAM_2_CR	0x0B4	RW	PRESET_N	DDRC Performance Parameter register
DDRC_PERF_PARAM_3_CR	0x0B8	RW	PRESET_N	DDRC Performance Parameter register
DDRC_DFI_RDDATA_EN_CR	0x0BC	RW	PRESET_N	DDRC DFI Read Command Timing register
DDRC_DFI_MIN_CTRLUPD_TIMING_CR	0x0C0	RW	PRESET_N	DDRC DFI Controller Update Min Time register
DDRC_DFI_MAX_CTRLUPD_TIMING_CR	0x0C4	RW	PRESET_N	DDRC DFI Controller Update Max Time register
Reserved	0x0C8 0x0CC 0x0D0 0x0D4	—	—	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
DDRC_DYN_SOFT_RESET_ALIAS_CR	0x0DC	—	PRESET_N	DDRC reset register
DDRC_AXI_FABRIC_PRI_ID_CR	0x0E0	RW	PRESET_N	DDRC AXI Interface Fabric Priority ID Register
DDRC_SR	0x0E4	RO	PRESET_N	DDRC STATUS Register
SECDED Registers				
DDRC_SINGLE_ERR_CNT_STATUS_SR	0x0E8	RO	PRESET_N	DDRC single error count STATUS Register
DDRC_DOUBLE_ERR_CNT_STATUS_SR	0x0EC	RO	PRESET_N	DDRC double error count status register
DDRC_LUE_SYNDROME_1_SR	0x0F0	RO	PRESET_N	DDRC last uncorrected error syndrome register
DDRC_LUE_SYNDROME_2_SR	0x0F4	RO	PRESET_N	DDRC last uncorrected error syndrome register
DDRC_LUE_SYNDROME_3_SR	0x0F8	RO	PRESET_N	DDRC last uncorrected error syndrome register
DDRC_LUE_SYNDROME_4_SR	0x0FC	RO	PRESET_N	DDRC last uncorrected error syndrome register
DDRC_LUE_SYNDROME_5_SR	0x100	RO	PRESET_N	DDRC last uncorrected error syndrome register
DDRC_LUE_ADDRESS_1_SR	0x104	RO	PRESET_N	DDRC last uncorrected error address register
DDRC_LUE_ADDRESS_2_SR	0x108	RO	PRESET_N	DDRC last uncorrected error address register
DDRC_LCE_SYNDROME_1_SR	0x10C	RO	PRESET_N	DDRC last corrected error syndrome register
DDRC_LCE_SYNDROME_2_SR	0x110	RO	PRESET_N	DDRC last corrected error syndrome register
DDRC_LCE_SYNDROME_3_SR	0x114	RO	PRESET_N	DDRC last corrected error syndrome register
DDRC_LCE_SYNDROME_4_SR	0x118	RO	PRESET_N	DDRC last corrected error syndrome register
DDRC_LCE_SYNDROME_5_SR	0x11C	RO	PRESET_N	DDRC last corrected error syndrome register
DDRC_LCE_ADDRESS_1_SR	0x120	RO	PRESET_N	DDRC last corrected error address register

.....continued				
Register Name	Address Offset	Register Type	Reset Source	Description
DDRC_LCE_ADDRESS_2_SR	0x124	RO	PRESET_N	DDRC last corrected error address register
DDRC_LCB_NUMBER_SR	0x128	RO	PRESET_N	DDRC last corrected bit number register
DDRC_LCB_MASK_1_SR	0x12C	RO	PRESET_N	DDRC last corrected bit mask status register
DDRC_LCB_MASK_2_SR	0x130	RO	PRESET_N	DDRC last corrected bit mask status register
DDRC_LCB_MASK_3_SR	0x134	RO	PRESET_N	DDRC last corrected bit mask status register
DDRC_LCB_MASK_4_SR	0x138	RO	PRESET_N	DDRC last corrected bit mask status register
DDRC_ECC_INT_SR	0x13C	RO	PRESET_N	DDRC SECDED interrupt status register
DDRC_ECC_INT_CLR_REG	0x140	RW	PRESET_N	DDRC SECDED interrupt clear register

2.2 DDR Controller Configuration Register Bit Definitions (Ask a Question)

The following tables provide the DDR controller Configuration register bit definitions.

Table 2-3. DDRC_DYN_SOFT_RESET_CR

Bit Number	Name	Reset Value	Description
[31:3]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
2	AXIRESET	0x1	Set when main AXI reset signal is asserted. Reads and writes to the dynamic registers must not be carried out. This is a read-only bit.
1	RESET_APB_REG	0x0	Full soft reset If this bit is set when the soft reset bit is written as 1, all APB registers reset to the power-up state.
0	REG_DDRC_SOFT_RSTB	0x0	This is a soft reset. 0: Puts the controller into reset. 1: Takes the controller out of reset. The controller must be taken out of reset only when all other registers have been programmed. Asserting this bit does not reset all the APB configuration registers. Once the soft reset bit is asserted, the APB register must be modified as required.

Table 2-4. DDRC_AXI_EXCLUSIVE_IDS_CR

Bit Number	Name	Reset Value	Description
[31:4]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:12]	EXCLUSIVE_ID3	0x3	Configures a fabric AXI initiator ID to support exclusive access.
[11:8]	EXCLUSIVE_ID2	0x2	Configures a fabric AXI initiator ID to support exclusive access.
[7:4]	EXCLUSIVE_ID1	0x1	Configures a fabric AXI initiator ID to support exclusive access.
[3:0]	EXCLUSIVE_ID0	0x0	Configures a fabric AXI initiator ID to support exclusive access.

Table 2-5. DDRC_DYN_REFRESH_1_CR

Bit Number	Name	Reset Value	Description
[31:15]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[14:7]	REG_DDRC_T_RFC_MIN	0x23	$t_{RFC\ (min)}$: Minimum time from refresh to refresh or activate (specification: 75 to 195 ns). Unit: clocks.
6	REG_DDRC_REFRESH_UPDATE_LEVEL	0x0	Toggle this signal to indicate that the refresh register(s) are updated. The value is automatically updated when exiting soft reset. Therefore, it does not need to be toggled initially.
5	REG_DDRC_SELFREF_EN	0x0	If 1, then the controller puts the DRAM into self refresh when the transaction store is empty.
[4:0]	REG_DDRC_REFRESH_TO_X32	0x8	When burst refresh is enabled (refresh_burst > 0), you might also make use of a feature called speculative refresh. Burst refresh is implemented by counting the number of times tREFI expires, and by issuing a group of refreshes when that number reaches the refresh burst number. If the refresh timer (t_{RFC_nom}, as known as t_{REFI}) has expired at least once, but it has not expired burst_of_N_refresh times yet, then a speculative refresh can be performed. A speculative refresh is a refresh performed at a time when refresh would be useful, but before it is absolutely required. When the DRAM bus is idle for a period of time determined by this refresh idle time-out and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed. Speculative refreshes continue successively until there are no refreshes pending or until new reads or writes are issued to the controller.

Table 2-6. DDRC_DYN_REFRESH_2_CR

Bit Number	Name	Reset Value	Description
[31:15]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[14:3]	REG_DDRC_T_RFC_NOM_X32	0x52	t_{REFI}: Average time between refreshes (specification: 7.8 µs). Unit: Multiples of 32 clocks.
[2:0]	REG_DDRC_REFRESH_BURST	0x0	The programmed value plus one is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute. Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes; therefore, performing refreshes in a burst reduces the per-refresh penalty of these page closings. Higher numbers for burst_of_N_refresh slightly increases utilization; lower numbers decreases the worst-case latency associated with refreshes. 0x0: Single refresh 0x1: Burst-of-2 0x7: Burst-of-8 refresh

Table 2-7. DDRC_DYN_POWERDOWN_CR

Bit Number	Name	Reset Value	Description
[31:2]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
1	REG_DDRC_POWERDOWN_EN	0x1	If true, the controller goes into power-down after a programmable number of cycles. This register bit might be reprogrammed during the course of normal operation.
0	REG_DDRC_DEEPPOWERDOWN_EN	0x0	1: Controller puts the DRAM into deep power-down mode when the transaction store is empty. 0: Brings controller out of deep power-down mode. Present only in designs that have mobile support.

Table 2-8. DDRC_MODE_CR

Bit Number	Name	Reset Value	Description
[31:9]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
8	REG_DDRC_DDR3	0x0	1: DDR3 operating mode 0: DDR2 operating mode
7	REG_DDRC_MOBILE	0x0	1: Mobile/LPDDR1 DRAM device in use 0: Non-mobile DRAM device in use
6	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify- write operation.
5	REG_DDRC_TEST_MODE	0x0	1: Controller is in test mode 0: Controller is in normal mode
[4:2]	REG_DDRC_MODE	0x0	DRAM SECDED mode 000: No SECDED 101: SECDED enabled All other selections are reserved.
[1:0]	REG_DDRC_DATA_BUS_WIDTH	0x0	00: Full DQ bus width to DRAM 01: Half DQ bus width to DRAM 10: Quarter DQ bus width to DRAM 11: Reserved

Table 2-9. DDRC_ADDR_MAP_BANK_CR

Bit Number	Name	Reset Value	Description
[31:12]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[11:8]	REG_DDRC_ADDRMAP_BANK_B0	0x0	Selects the address bits used as Bank Address bit 0. Valid Range: 0 to 14 Internal Base: 2 The selected address bit for each of the Bank Address bits is determined by adding the internal base to the value of this field.

.....continued			
Bit Number	Name	Reset Value	Description
[7:4]	REG_DDRC_ADDRMAP_BANK_B1	0x0	Selects the address bits used as Bank Address bit 1. Valid Range: 0 to 14 Internal Base: 3 The selected address bit for each of the Bank Address bits is determined by adding the internal base to the value of this field.
[3:0]	REG_DDRC_ADDRMAP_BANK_B2	0x0	Selects the address bits used as Bank Address bit 2. Valid Range: 0 to 14 and 15 Internal Base: 4 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, Bank Address bit 2 is set to 0.

Table 2-10. DDRC_ADDR_MAP_COL_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:12]	REG_DDRC_ADDRMAP_COL_B2	0x0	Full bus width mode: Selects column address bit 3. Half bus width mode: Selects column address bit 4. Quarter bus width mode: Selects column address bit 5. Valid range: 0 to 7 Internal base: 2 The selected address bit is determined by adding the internal base to the value of this field.
[11:8]	REG_DDRC_ADDRMAP_COL_B3	0x0	Full bus width mode: Selects column address bit 4 Half bus width mode: Selects column address bit 5 Quarter bus width mode: Selects column address bit 6 Valid range: 0 to 7 Internal base: 3 The selected address bit is determined by adding the internal base to the value of this field.
[7:4]	REG_DDRC_ADDRMAP_COL_B4	0x0	Full bus width mode: Selects column address bit 5 Half bus width mode: Selects column address bit 6 Quarter bus width mode: Selects column address bit 7 Valid Range: 0 to 7 Internal base: 4 The selected address bit for each of the column address bits is determined by adding the internal base to the value of this field.
[3:0]	REG_DDRC_ADDRMAP_COL_B7	0x0	Full bus width mode: Selects column address bit 8 Half bus width mode: Selects column address bit 9 Quarter bus width mode: Selects column address bit 11 Valid range: 0 to 7, and 15 Internal base: 7 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, column address bit 9 is set to 0.

Table 2-11. DDRC_ADDR_MAP_COL_2_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:12]	REG_DDRC_ADDRMAP_COL_B8	0x0	Full bus width mode: Selects column address bit 9 Half bus width mode: Selects column address bit 11 Quarter bus width mode: Selects column address bit 12 Valid range: 0 to 7, and 15 Internal base: 8 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, column address bit 9 is set to 0.
[11:8]	REG_DDRC_ADDRMAP_COL_B9	0x0	Full bus width mode: Selects column address bit 11 Half bus width mode: Selects column address bit 12 Quarter bus width mode: Selects column address bit 13 Valid range: 0 to 7, and 15 Internal base: 9 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, column address bit 9 is set to 0.
[7:4]	REG_DDRC_ADDRMAP_COL_B10	0x0	Full bus width mode: Selects column address bit 12 Half bus width mode: Selects column address bit 13 Quarter bus width mode: Unused. must be set to 15 Valid range: 0 to 7, and 15 Internal base: 10 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, column address bit 10 is set to 0.
[3:0]	REG_DDRC_ADDRMAP_COL_B11	0x0	Full bus width mode: Selects column address bit 13 Half bus width mode: Unused. To make it unused, this must be tied to 0xF. Quarter bus width mode: Unused. To make it unused, this must be tied to 0xF. Valid range: 0 to 7, and 15 Internal base: 11 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, column address bit 11 is set to 0.

Table 2-12. DDRC_ADDR_MAP_ROW_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:12]	REG_DDRC_ADDRMAP_ROW_B0	0x0	Select the address bits used as row address bit 0. Valid range: 0 to 11 Internal base: 6 The selected address bit for each of the row address bits is determined by adding the internal base to the value of this field.
[11:8]	REG_DDRC_ADDRMAP_ROW_B1	0x0	Select the address bits used as row address bit 1. Valid range: 0 to 11 Internal base: 7 The selected address bit for each of the row address bits is determined by adding the internal base to the value of this field.

.....continued			
Bit Number	**Name**	**Reset Value**	**Description**
[7:4]	REG_DDRC_ADDRMAP_ROW_B2_11	0x0	Select the address bits used as row address bits 2 to 11. Valid Range: 0 to 11 Internal Base: 8 for row address bit 2 9 for row address bit 3 10 for row address bit 4 .... 15 for row address bit 9 16 for row address bit 10 17 for row address bit 11 The selected address bit for each of the row address bits is determined by adding the internal base to the value of this field.
[3:0]	REG_DDRC_ADDRMAP_ROW_B12	0x0	Select the address bit used as row address bit 12. Valid Range: 0 to 11, and 15 Internal Base: 18 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 12 is set to 0.

Table 2-13. DDRC_ADDR_MAP_ROW_2_CR

Bit Number	Name	Reset Value	Description
[31:12]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[11:8]	REG_DDRC_ADDRMAP_ROW_B13	0x0	Select the address bits used as row address bit 13. Valid range: 0 to 11, and 15 Internal base: 19 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 13 is set to 0.
[7:4]	REG_DDRC_ADDRMAP_ROW_B14	0x0	Select the address bit used as row address bit 14. Valid range: 0 to 11, and 15 Internal base: 20 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 14 is set to 0.
[3:0]	REG_DDRC_ADDRMAP_ROW_B15	0x0	Select the address bit used as row address bit 15. Valid range: 0 to 11, and 15 Internal base: 21 The selected address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 15 is set to 0.

Table 2-14. DDRC_INIT_1_CR

Bit Number	Name	Reset Value	Description
[31:12]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued			
Bit Number	Name	Reset Value	Description
[11:8]	REG_DDRC_PRE_OCD_X32	0x0	Wait period before driving the on-chip driver calibration (OCD) Complete command to DRAM. Units are in counts of a global timer that pulses every 32 clock cycles. There is no known specific requirement for this. It might be set to 0.
[7:1]	REG_DDRC_FINAL_WAIT_X32	0x0	Cycles to wait after completing the DRAM initialization sequence before starting the dynamic scheduler. Units are in counts of a global timer that pulses every 32 clock cycles. There is known specific requirement for this; it might be set to 0.
0	REG_DDRC_SKIP_OCD	0x1	This register must be kept at 1. 1: Indicates the controller to skip the on-chip driver calibration (OCD) adjustment step during DDR2 initialization. OCD_Default and OCD_Exit are performed instead. 0: Not supported

Table 2-15. DDRC_CKE_RSTN_CYCLES_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:8]	REG_DDRC_PRE_CKE_X1024	0x0	The 10-bit REG_DDRC_PRE_CKE_X1024 [9:0] value is spit across the two registers DDRC_CKE_RSTN_CYCLES_1_CR and DDRC_CKE_RSTN_CYCLES_2_CR. [7:0] bits of REG_DDRC_PRE_CKE_X1024. Cycles to wait after reset before driving CKE High to start the DRAM initialization sequence. Units: 1024 clock cycles. DDR2 specifications typically require this to be programmed for a delay of >= 200 µs. For DDR3, an additional 500 µs delay must be set in this field.
[7:0]	REG_DDRC_DRAM_RSTN_X1024	0x0	Number of cycles to assert DRAM reset signal during initialization sequence. This is only present for implementations supporting DDR3 devices.

Table 2-16. DDRC_ CKE_RSTN_CYCLES_2_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[11:2]	REG_DDRC_POST_CKE_X1024	0x0	Cycles to wait after driving CKE High to start the DRAM initialization sequence. Units: 1024 clocks. DDR: Requires a 400 ns delay, requiring this value to be programmed to 2 at all clock speeds. SDR: Requires this to be programmed for a delay of 100 to 200 µs.

.....continued

Bit Number	Name	Reset Value	Description
[1:0]	REG_DDRC_PRE_CKE_X1024	0x0	This field represents the upper 2 bits of the 10-bit REG_DDRC_PRE_CKE_X1024 value split across the 2 registers DDRC_CKE_RSTN_CYCLES_1_CR and DDRC_CKE_RSTN_CYCLES_2_CR. [9:8] bits of REG_DDRC_PRE_CKE_X1024. Cycles to wait from the start of reset assertion before driving CKE High to start the DRAM initialization sequence. Units: 1024 clock cycles. DDR2 specifications typically require this to be programmed for a delay of >= 200 µs.

Table 2-17. DDRC_INIT_MR_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	REG_DDRC_MR	0x095A	Value to be loaded into the DRAM mode register. During the DRAM initialization procedure, the controller sends the mode register setting to the DRAM. The mode register sets the DRAM burst length, burst type, CAS latency (CL), and operating mode. Bit 8 is for the DLL and the DDR controller set them appropriately.

Table 2-18. DDRC_INIT_EMR_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	REG_DDRC_EMR	0x0402	Value is loaded into DRAM EMR registers. Bits [9:7] are for OCD and the setting in this bits is ignored. The controller sets those bits appropriately.

Table 2-19. DDRC_INIT_EMR2_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	REG_DDRC_EMR2	0x0	Value is loaded into DRAM EMR2 registers.

Table 2-20. DDRC_INIT_EMR3_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	REG_DDRC_EMR3	0x0	Value to be loaded into DRAM EMR3 registers.

Table 2-21. DDRC_DRAM_BANK_TIMING_PARAM_CR

Bit Number	Name	Reset Value	Description
[31:12]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued			
Bit Number	**Name**	**Reset Value**	**Description**
[11:6]	REG_DDRC_T_RC	0x0	t_{RC}: Minimum time between the cycles to activate the same bank (specification: 65 ns for DDR2-400 and smaller for faster parts). Unit: clocks.
[5:0]	REG_DDRC_T_FAW	0x0	t_{FAW}: Valid only in burst-of-8 mode. At most 4 banks must be activated in a rolling window of t_{FAW} cycles. Unit: clocks.

Table 2-22. DDRC_DRAM_RD_WR_LATENCY_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:5]	REG_DDRC_WRITE_LATENCY	0x0	Number of clocks between the write command to write data enable PHY.
[4:0]	REG_DDRC_READ_LATENCY	0x0	Time from read command to read data on DRAM interface. Set this to CL. Unit: clocks This signal is present for designs supporting LPDDR1 DRAM only. It is used to calculate when the DRAM clock is stopped.

Table 2-23. DDRC_DRAM_RD_WR_PRE_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:5]	REG_DDRC_WR2PRE	0x0	Minimum time between write and precharge to same bank (specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @ 400 MHz and less for lower frequencies). Unit: Clocks where: WL: Write latency BL: Burst length. This must match the value programmed in the BL bit of the mode register to the DRAM. t_{WR}: Write recovery time. This comes directly from the DRAM specs.
[4:0]	REG_DDRC_RD2PRE	0x0	t_{RTP}: Minimum time from read to precharge of same bank (specification: t_{RTP} for BL = 4 and t_{RTP} + 2 for BL = 8. t_{RTP} = 7.5 ns). Unit: clocks.

Table 2-24. DDRC_DRAM_MR_TIMING_PARAM_CR

Bit Number	Name	Reset Value	Description
[31:13]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[12:3]	REG_DDRC_T_MOD	0x0	Present for DDR3 only (replaces REG_DDRC_T_MRD functionality when used with DDR3 devices). The mode register set command updates delay in number of clock cycles. This is required to be programmed even when a design that supports DDR3 is running in DDR2 mode (minimum is the larger of 12 clock cycles or 15 ns).
[2:0]	REG_DDRC_T_MRD	0x0	t_{MRD}: Cycles between load mode commands. Not used in DDR3 mode.

Table 2-25. DDRC_DRAM_RAS_TIMING_CR

Bit Number	Name	Reset Value	Description
[31:11]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[10:5]	REG_DDRC_T_RAS_MAX	0x0	t_{RAS} (max): Maximum time between activate and precharge to same bank. Maximum time that a page can be kept open (specification: 70 µs). Minimum value of this register is 1. Zero is invalid. Unit: Multiples of 1024 clocks.
[4:0]	REG_DDRC_T_RAS_MIN	0x0	t_{RAS} (min): Minimum time between activate and precharge to the same bank (specification: 45 ns). Unit: clocks.

Table 2-26. DDRC_DRAM_RD_WR_TRNARND_TIME_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:5]	REG_DDRC_RD2WR	0x0	RL + BL/2 + 2 – WL Minimum time from READ command to WRITE command. Include time for bus turnaround and all per-bank, per-rank, and global constraints. Unit: clocks. where, WL: Write latency BL: Burst length. This must match the value programmed in the BL bit of the mode register to the DRAM. RL: Read latency: CAS latency.
[4:0]	REG_DDRC_WR2RD	0x0	WL + t_{WTR} + BL/2 Minimum time from WRITE command to READ command. Includes time for bus turnaround and recovery times and all per-bank, per- rank, and global constraints. Unit: clocks. where, WL: Write latency. BL: Burst length. This must match the value programmed in the BL bit of the mode register to the DRAM. t_{WTR}: Internal WRITE to READ command delay. This comes directly from the DRAM specifications.

Table 2-27. DDRC_DRAM_T_PD_CR

Bit Number	Name	Reset Value	Description
[31:9]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[8:4]	REG_DDRC_T_XP	0x0	t_{XP}: Minimum time after power-down exit to any operation. Units: clocks
[3:0]	REG_DDRC_T_CKE	0x0	Minimum number of cycles of CKE High/Low during power-down and self refresh. Unit: clocks

Table 2-28. DDRC_DRAM_BANK_ACT_TIMING_CR

Bit Number	Name	Reset Value	Description
[31:14]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued			
Bit Number	**Name**	**Reset Value**	**Description**
[13:10]	REG_DDRC_T_RCD	0x0	t_{RCD}: Minimum time from activate to READ or WRITE command to same bank (specification: 15 ns for DDR2-400 and lower for faster devices). Unit: clocks.
[9:7]	REG_DDRC_T_CCD	0x0	t_{CCD}: Minimum time between two reads or two writes (from bank A to bank B) (specification: 2 cycles) is this value + 1. Unit: clocks.
[6:4]	REG_DDRC_T_RRD	0x0	t_{RRD}: Minimum time between the cycle activates from bank A to bank B (specification: 10 ns or less). Unit: clocks.
[3:0]	REG_DDRC_T_RP	0x0	t_{RP}: Minimum time from precharge to activate of same bank. Unit: clocks.

Table 2-29. DDRC_ODT_PARAM_1_CR

Bit Number	Name	Reset Value	Description
[31:12]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[11:8]	REG_DDRC_RD_ODT_DELAY	0x0	The delay, in clock cycles, from issuing a READ command to setting ODT values is associated with that command. Recommended value for DDR2 is CL4.
[7:4]	REG_DDRC_WR_ODT_DELAY	0x0	The delay, in clock cycles, from issuing a WRITE command to setting ODT values is associated with that command. The recommended value for DDR2 is CL5, where CL is CAS latency. DDR ODT has a 2-cycle on-time delay and a 2.5-cycle off-time delay. ODT settings must remain constant for the entire time that DQS is driven by the controller.
[3:2]	REG_DDRC_RANK0_WR_ODT	0x0	0: Indicates which remote ODTs must be turned ON during a write to rank 0. Each rank has a remote ODT (in the DRAM) which can be turned ON by setting the appropriate bit here. Set this bit to 1 to enable its ODT. 1: Uppermost bit is unused.
[1:0]	REG_DDRC_RANK0_RD_ODT	0x0	0: Indicates which remote ODTs must be turned ON during a read to rank 0. Each rank has a remote ODT (in the DRAM) which can be turned ON by setting the appropriate bit here. Set this bit to 1 to enable its ODT. 1: Uppermost bit is unused.

Table 2-30. DDRC_ODT_PARAM_2_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:6]	REG_DDRC_RD_ODT_HOLD	0x0	Cycles to hold ODT for a READ command. 0: ODT signal is ON for 1 cycle. 1: ODT signal is ON for 2 cycles, and so on.
[5:2]	REG_DDRC_WR_ODT_HOLD	0x0	Cycles to hold ODT for a WRITE command. 0: ODT signal is ON for 1 cycle. 1: ODT signal is ON for 2 cycles, and so on.

.....continued			
Bit Number	Name	Reset Value	Description
[1:0]	REG_DDRC_WR_ODT_BLOCK	0x0	00: Block read/write scheduling for 1-cycle when write requires changing ODT settings. 01: Block read/write scheduling for 2 cycles when write requires changing ODT settings. 10: Block read/write scheduling for 3 cycles when write requires changing ODT settings. 11: Reserved

Table 2-31. DDRC_ADDR_MAP_COL_3_CR

Bit Number	Name	Reset Value	Description
[31:16] [7:6]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:12]	REG_DDRC_ADDRMAP_COL_B5	0x0	Full bus width mode: Selects column address bit 6. Half bus width mode: Selects column address bit 7. Quarter bus width mode: Selects column address bit 8. Valid range: 0 to 7 Internal base: 5 The selected address bit for each of the column address bits is determined by adding the internal base to the value of this field.
[11:8]	REG_DDRC_ADDRMAP_COL_B6	0x0	Full bus width mode: Selects column address bit 7. Half bus width mode: Selects column address bit 8. Quarter bus width mode: Selects column address bit 9. Valid range: 0 to 7 Internal base: 6 The selected address bit for each of the column address bits is determined by adding the internal base to the value of this field.
5	REG_DDRC_DIS_WC	0x0	When 1, disable write combine.
4	REG_DDRC_DIS_ACT_BYPASS	0x0	Only present in designs supporting activate bypass. When 1, disable bypass path for high priority read activates
3	REG_DDRC_DIS_RD_BYPASS	0x0	Only present in designs supporting read bypass. When 1, disable bypass path for high-priority read page hits.
2	REG_DDRC_DIS_PRE_BYPASS	0x0	Only present in designs supporting precharge bypass. When 1, disable bypass path for high-priority precharges
1	REG_DDRC_DIS_COLLISION_PAGE_OPT	0x0	When this is set to 0, auto-precharge is disabled for the flushed command in a collision case. Collision cases are write followed by read to same address, read followed by write to same address, or write followed by write to same address with REG_DDRC_DIS_WC bit = 1 (where same address comparisons exclude the two address bits representing the critical word).
0	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read- modify-write operation.

Table 2-32. DDRC_MODE_REG_RD_WR_CR

Bit Number	Name	Reset Value	Description
[31:4]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
3	REG_DDRC_MR_WR	0x0	When 1 is written and DDRC_REG_MR_WR_BUSY is low, a mode register write operation is started. It is not required to set this bit back to 0. This bit always reads as 0.
[2:1]	REG_DDRC_MR_ADDR	0x0	Address of the Mode register that is to be written to. 00: MR0 01: MR1 10: MR2 11: MR3
0	REG_DDRC_MR_TYPE	0x0	Indicates a write operation for the Mode register. Read operation is not supported and so 1 is not present. Only 0 is present and 0 indicates a write operation. 1: Not supported 0: Write

Table 2-33. DDRC_MODE_REG_DATA_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	REG_DDRC_MR_DATA	0x0	Mode register write data

Table 2-34. DDRC_PWR_SAVE_1_CR

Bit Number	Name	Reset Value	Description
[31:13]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[12:6]	REG_DDRC_POST_SELFREF_GAP_X32	0x10	Minimum time to wait after coming out of self refresh before doing anything. This must be larger than all the constraints that exist (Specifications: maximum of t_{XSNR} and t_{XSRD} and tXSDLL, which is 512 clocks). Unit: Multiples of 32 clocks.
[5:1]	REG_DDRC_POWERDOWN_TO_X32	0x06	After these many clocks of NOP or DESELECT, the controller puts the DRAM into power-down. This must be enabled in the Initiator Control register. Unit: Multiples of 32 clocks.
0	REG_DDRC_CLOCK_STOP_EN	0x0	1: Stops the clock to the PHY whenever a clock is not required by LPDDR1. 0: Clock is never stopped. This is only present for implementations supporting mobile/ LPDDR1 devices.

Table 2-35. DDRC_PWR_SAVE_2_CR

Bit Number	Name	Reset Value	Description
[31:0]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

Table 2-36. DDRC_ZQ_LONG_TIME_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:0]	REG_DDRC_T_ZQ_LONG_NOP	0x0	Number of cycles of NOP required after a ZQCL (ZQ calibration long) command is issued to DRAM. Units: Clock cycles. This is only present for implementations supporting DDR3 devices

Table 2-37. DDRC_ZQ_SHORT_TIME_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:0]	REG_DDRC_T_ZQ_SHORT_NOP	0x0	Number of cycles of NOP required after a ZQCS (ZQ calibration short) command is issued to DRAM. Units: Clock cycles. This is only present for implementations supporting DDR3 devices.

Table 2-38. DDRC_ZQ_SHORT_INT_REFRESH_MARGIN_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:4]	REG_DDRC_T_ZQ_SHORT_INTERVAL_X1024	0x0	20 bits are split into two registers. [11:0] bits of REG_DDRC_T_ZQ_SHORT_INTERVAL_ X1024. Average interval to wait between automatically issuing ZQ calibration short (ZQCS) commands to DDR3 devices. Units: 1024 clock cycles This is only present for implementations supporting DDR3 devices.
[3:0]	REG_DDRC_REFRESH_MARGIN	0x02	Threshold value in number of clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. **Note:** Microchip recommends using the default value. Unit: Multiples of 32 clocks.

Table 2-39. DDRC_ZQ_SHORT_INT_REFRESH_MARGIN_2_CR

Bit Number	Name	Reset Value	Description
[31:8]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued			
Bit Number	Name	Reset Value	Description
[7:0]	REG_DDRC_T_ZQ_SHORT_INTERVAL_X1024	0x0	20 bits are split into two registers. [19:12] bits of REG_DDRC_T_ZQ_SHORT_INTERVAL_X1024. Average interval to wait between automatically issuing ZQ calibration short (ZQCS) commands to DDR3 devices. Units: 1024 clock cycles This is only present for implementations supporting DDR3 devices.

Table 2-40. DDRC_PERF_PARAM_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:13]	REG_DDRC_BURST_RDWR	0x0	001: Burst length of 4 010: Burst length of 8 100: Burst length of 16 All other values are reserved. This controls the burst size used to access the DRAM. This must match the BL mode register setting in the DRAM. The DDRC and AXI controllers are optimized for a burst length of 8. The recommended setting is 8. A burst length of 16 is only supported for LPDDR1. Setting to 16 when using LPDDR1 in half/quarter bus mode might boost the performance. For systems that tend to do many single cycle random transactions, a burst length of 4 might slightly improve system performance.
12	Reserved	0x0	This bit must always be set to 0.
[11:5]	REG_DDRC_RDWR_IDLE_GAP	0x04	When the preferred transaction store is empty for this many clock cycles, switch to the alternate transaction store if it is non-empty. The read transaction store (both high and low priority) is the default preferred transaction store and the write transaction store is the alternate store. When Prefer write over read is set, this is reversed.
4	REG_DDRC_PAGECLOSE	0x0	1: Bank is closed and kept closed if no transactions are available for it. This is different from auto-precharge: • Explicit precharge commands are used, and not read/write with auto-precharge. • Page is not closed after a read/write if there is another read/write pending to the same page. 0: Bank remains open until there is a need to close it (to open a different page, or for page time-out or refresh time-out).

.....continued

Bit Number	Name	Reset Value	Description
[3:0]	REG_DDRC_LPR_NUM_ ENTRIES	0x03	Number of credits available for the low priority transaction stores (REG_DDRC_LPR_NUM_ENTRIES + 1). The ECC system (both when On or OFF) takes an LPR credit and so the actual number of credits is REG_DDRC_LPR_NUM_ENTRIES. Number of credits available for the high priority transaction store is READ_CAM_DEPTH - (REG_DDRC_LPR_NUM_ENTRIES + 1). READ_CAM_DEPTH = Depth of the read transaction store, that is 16. Setting this to the maximum value allocates all entries to low priority transaction store. Setting this to 0 allocates a single entry to low priority transaction store and the rest to high priority transaction store.

Table 2-41. DDRC_HPR_QUEUE_PARAM_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
15	REG_DDRC_HPR_MAX_STARVE_X32	0x0	Lower 1 bit of REG_DDRC_HPR_MAX_STARVE_X32. Number of clocks that the HPR queue can be starved before it goes critical. Unit: 32 clocks.
[14:4]	REG_DDRC_HPR_MIN_NON_CRITICAL	0x0	Number of clocks that the HPR queue is ensured to be non-critical. Unit: 32 clocks.
[3:0]	REG_DDRC_HPR_XACT_RUN_LENGTH	0x0	Number of transactions that are serviced once the HPR queue goes critical is the smaller of this value and the number of transactions is available. Units: Transactions.

Table 2-42. DDRC_HPR_QUEUE_PARAM_2_CR

Bit Number	Name	Reset Value	Description
[31:11]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[10:0]	REG_DDRC_HPR_MAX_STARVE_X32	0x0	[11:1] bits of REG_DDRC_HPR_MAX_STARVE_X32 Number of clocks that the HPR queue can be starved before it goes critical. Unit: 32 clocks.

Table 2-43. DDRC_LPR_QUEUE_PARAM_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
15	REG_DDRC_LPR_MAX_STARVE_X32	0x0	12 bits are split into two registers. Lower 1 bit of REG_DDRC_LPR_MAX_STARVE_X32. Number of clocks that the LPR queue can be starved before it goes critical. Unit: 32 clocks.

.....continued

Bit Number	Name	Reset Value	Description
[14:4]	REG_DDRC_LPR_MIN_NON_CRITICAL	0x0	Number of clocks that the LPR queue is ensured to be non-critical. Unit: 32 clocks.
[3:0]	REG_DDRC_LPR_XACT_RUN_LENGTH	0x0	Number of transactions that are serviced once the LPR queue goes critical is the smaller of this value and the number of transactions is available. Units: Transactions.

Table 2-44. DDRC_LPR_QUEUE_PARAM_2_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[10:0]	REG_DDRC_LPR_MAX_STARVE_X32	0x0	12 bits are split into two registers. [11:1] bits of REG_DDRC_HPR_MAX_STARVE_X32. Number of clocks that the LPR queue can be starved before it goes critical. Unit: 32 clocks.

Table 2-45. DDRC_WR_QUEUE_PARAM_CR

Bit Number	Name	Reset Value	Description
[31:15]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[14:4]	REG_DDRC_W_MIN_NON_CRITICAL	0x0	Number of clocks that the write queue is ensured to be non-critical. Unit: 32 clocks.
[3:0]	REG_DDRC_W_XACT_RUN_LENGTH	0x0	Number of transactions that are serviced once the WR queue goes critical is the smaller of this value and number of transactions available. Units: Transactions.

Table 2-46. DDRC_PERF_PARAM_2_CR

Bit Number	Name	Reset Value	Description
[31:11]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
10	REG_DDRC_BURST_MODE	0x0	1: Interleaved burst mode 0: Sequential burst mode The burst mode programmed in the DRAM mode register and the order of the input data to the controller must both match the value programmed in the REG_DDRC_BURST_MODE register.
[9:2]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
1	REG_DDRC_PREFER_WRITE	0x0	If set, the bank selector prefers writes over reads.
0	REG_DDRC_FORCE_LOW_PRI_N	0x0	Active-Low signal. When asserted 0, all incoming transactions are forced to low priority. Forcing the incoming transactions to low priority implicitly turns off bypass.

Table 2-47. DDRC_PERF_PARAM_3_CR

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	REG_DDRC_EN_2T_TIMING_MODE	0x0	1: DDRC uses 2T timing. 0: DDRC uses 1T timing.

Table 2-48. DDRC_DFI_RDDATA_EN_CR

Bit Number	Name	Reset Value	Description
[31:5]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[4:0]	REG_DDRC_DFI_T_RDDATA_EN	0x0	Time from the assertion of a READ command on the DFI interface to the assertion of the DDRC_DFI_RDDATA_EN signal. Program this to (RL – 1), where RL is the read latency of the DRAM. For LPDDR1 this must be set to RL. Units: Clocks.

Table 2-49. DDRC_DFI_MIN_CTRLUPD_TIMING_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:0]	REG_DDRC_DFI_T_CTRLUP_MIN	0x03	Specifies the minimum number of clock cycles that the DDRC_DFI_CTRLUPD_REQ signal must be asserted. Lowest value to assign to this variable is 0x3. Units: Clocks.

Table 2-50. DDRC_DFI_MAX_CTRLUPD_TIMING_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:0]	REG_DDRC_DFI_T_CTRLUP_MAX	0x40	Specifies the maximum number of clock cycles that the DDRC_DFI_CTRLUPD_REQ signal can assert. Lowest value to assign to this variable is 0x40. Units: Clocks.

Table 2-51. DDRC_DYN_SOFT_RESET_ALIAS_CR

Bit Number	Name	Reset Value	Description
[31:3]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
2	AXIRESET	0x1	Set when the main AXI reset signal is asserted. Reads and writes to the dynamic registers must not be carried out. This is a read-only bit.
1	RESET_APB_REG	0x0	Full soft reset If this bit is set when the soft reset bit is written as 1, all APB registers reset to the power-up state.

.....continued

Bit Number	Name	Reset Value	Description
0	REG_DDRC_SOFT_RSTB	0x0	This is a soft reset. 0: Puts the controller into reset. 1: Takes the controller out of reset. The controller must be taken out of reset only when all other registers have been programmed. Asserting this bit does not reset all the APB configuration registers. Once the soft reset bit is asserted, the APB register must be modified as required.

Table 2-52. DDRC_AXI_FABRIC_PRI_ID_CR

Bit Number	Name	Reset Value	Description
[31:9]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
8	BORROW_LPR	0x0	1: The DDR Controller can use LPR credits for HPR transactions if no HPR credits are available.
7	BORROW_HPR	0x0	1: The DDR Controller can use HPR credits for LPR transactions if no LPR credits are available
6	FABRIC_LOCK_PRIORITY	0x0	1: Fabric initiator lock transaction gets high priority
[5:4]	PRIORITY_ENABLE_BIT	0x0	This is to set the priority of the fabric initiator ID. 01/10/11: Indicates that the ID is higher priority. 00: None of the initiator IDs from the fabric has a higher priority.
[3:0]	PRIORITY_ID	0x0	If the Priority Enable bit is 1, this ID will has a higher priority over other IDs.

Table 2-53. DDRC_SR

Bit Number	Name	Reset Value	Description
[31:6]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[5:3]	DDRC_CORE_REG_OPERATING_MODE	0x0	Operating mode. This is 3 bits wide in designs with mobile support and 2 bits in all other designs. Non-mobile designs: • 000: Init • 001: Normal • 010: Power-down • 011: Self Refresh Mobile designs: • 000: Init • 001: Normal • 010: Power-down • 011: Self refresh • 1XX: Deep power-down
[2:1]	Reserved	–	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued

Bit Number	Name	Reset Value	Description
0	DDRC_REG_MR_WR_BUSY	0x0	1: Indicates that a mode register write operation is in progress. 0: Indicates that the core can initiate a mode register write operation. Core must initiate an MR write operation only if this signal is Low. This signal goes High in the clock after the controller accepts the write request. It goes Low when the MR write command is issued to the DRAM. Any MR write command that is received when DDRC_REG_MR_WR_BUSY is High, is not accepted.

Table 2-54. DDRC_SINGLE_ERR_CNT_STATUS_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	–	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_SINGLE_ERR_CNT_STATUS_REG	0x0	Single error count status. If the count reaches 0xFFFF, it is held and only cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-55. DDRC_DOUBLE_ERR_CNT_STATUS_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_DOUBLE_ERR_CNT_STATUS_REG	0x0	Double error count status. If the count reaches 0xFFFF, then it is held and only cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-56. DDRC_LUE_SYNDROME_1_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [15:0] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_ERR_READ_DONE_CR is written over by the system.

Table 2-57. DDRC_LUE_SYNDROME_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [31:16] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-58. DDRC_LUE_SYNDROME_3_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [47:32] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-59. DDRC_LUE_SYNDROME_4_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued

Bit Number	Name	Reset Value	Description
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [63:48] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-60. DDRC_LUE_SYNDROME_5_SR

Bit Number	Name	Reset Value	Description
[16:8]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[7:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [71:64] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-61. DDRC_LUE_ADDRESS_1_SR

Bit Number	Name	Reset Value	Description
[31:15]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[14:12]	DDRC_REG_ECC_BANK	0x0	Bank where the SECDED error occurred. Only present in designs that support SECDED.

.....continued			
Bit Number	Name	Reset Value	Description
[11:0]	DDRC_REG_ECC_COL	0x0	Column where the SECDED error occurred. Col[0] is always set to 0, coming out of the controller. This bit is overwritten by the register module and indicates whether the error has occurred from upper or lower lane. Only present in designs that support SECDED.

Table 2-62. DDRC_LUE_ADDRESS_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_ROW	0x0	Row where the SECDED error occurred. Only present in designs that support SECDED.

Table 2-63. DDRC_LCE_SYNDROME_1_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [15:0] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: SECDED: [71:64] – SECDED [63:00] – Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-64. DDRC_LCE_SYNDROME_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued

Bit Number	Name	Reset Value	Description
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [31:16] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: SECDED: [71:64] – SECDED [63:00] – Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, then the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-65. DDRC_LCE_SYNDROME_3_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [47:32] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-66. DDRC_LCE_SYNDROME_4_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued

Bit Number	Name	Reset Value	Description
[15:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [63:48] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following: [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-67. DDRC_LCE_SYNDROME_5_SR

Bit Number	Name	Reset Value	Description
[16:8]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[7:0]	DDRC_REG_ECC_SYNDROMES	0x0	72 bits are split into five registers. [71:64] bits of DDRC_REG_ECC_SYNDROMES. First data which has SECDED error in it. 72 bits consist of the following [71:64]: SECDED [63:00]: Data In the same clock cycle, if one lane has a correctable error and the other lane has an uncorrectable error, the syndrome for the uncorrectable error is sent on this bus. If more than one data lane has an error in it, the lower data lane is selected. The priority applied when there are multiple errors in the same cycle is as follows: Uncorrectable error, lower lane Uncorrectable error, upper lane Correctable error, lower lane Correctable error, upper lane Only present in designs that support SECDED. This is cleared after DDRC_ECC_INT_CLR_REG is written over by the system.

Table 2-68. DDRC_LCE_ADDRESS_1_SR

Bit Number	Name	Reset Value	Description
[31:15]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[14:12]	DDRC_REG_ECC_BANK	0x0	Bank where the SECDED error occurred.
[11:0]	DDRC_REG_ECC_COL	0x0	Column where the SECDED error occurred. Col[0] is always set to 0 coming out of the controller. This bit is overwritten by the register module and indicates whether the error has occurred from upper or lower lane.

Table 2-69. DDRC_LCE_ADDRESS_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_REG_ECC_ROW	0x0	Row where the SECDED error occurred.

Table 2-70. DDRC_LCB_NUMBER_SR

Bit Number	Name	Reset Value	Description
[31:7]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[6:0]	DDRC_LCB_BIT_NUM	0x0	Indicates the location of the bit that caused a single-bit error in SECDED case (encoded value). If more than one data lane has an error in it, the lower data lane is selected. This register is 7 bits wide to handle 72 bits of the data present in a single lane. This does not indicate CORRECTED_BIT_NUM in the case of device correction SECDED. The encoding is only present in designs that support SECDED.

Table 2-71. DDRC_LCB_MASK_1_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_LCB_MASK	0x0	64 bits are split into four registers. [15:0] bits of DDRC_LCB_MASK. Indicates the mask of the corrected data. 1: On any bit indicates that the bit has been corrected by the DRAM SECDED logic. 0: On any bit indicates that the bit has NOT been corrected by the DRAM SECDED logic. Valid when a correctable error occurs. This mask does not indicate any correction that has been made in the SECDED check bits. If there are errors in multiple lanes, this signal has the mask for the lowest lane.

Table 2-72. DDRC_LCB_MASK_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_LCB_MASK	0x0	64 bits are split into four registers. [31:16] bits of DDRC_LCB_MASK. Indicates the mask of the corrected data. 1: On any bit indicates that the bit has been corrected by the DRAM SECDED logic. 0: On any bit indicates that the bit has not been corrected by the DRAM SECDED logic. Valid when correctable error occurs. This mask does not indicate any correction that has been made in the SECDED check bits. If there are errors in multiple lanes, this signal has the mask for the lowest lane.

Table 2-73. DDRC_LCB_MASK_3_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_LCB_MASK	0x0	64 bits are split into four registers. [47:32] bits of DDRC_LCB_MASK. Indicates the mask of the corrected data. 1: On any bit indicates that the bit has been corrected by the DRAM SECDED logic. 0: On any bit indicates that the bit has not been corrected by the DRAM SECDED logic. Valid when correctable error occurs. This mask does not indicate any correction that has been made in the SECDED check bits. If there are errors in multiple lanes, this signal has the mask for the lowest lane.

Table 2-74. DDRC_LCB_MASK_4_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDRC_LCB_MASK	0x0	64 bits are split into four registers. [63:48] bits of DDRC_LCB_MASK. Indicates the mask of the corrected data. 1: On any bit indicates that the bit has been corrected by the DRAM SECDED logic. 0: On any bit indicates that the bit has not been corrected by the DRAM SECDED logic. Valid when correctable error occurs. This mask does not indicate any correction that has been made in the SECDED check bits. If there are errors in multiple lanes, this signal has the mask for the lowest lane.

Table 2-75. DDRC_ECC_INT_SR

Bit Number	Name	Reset Value	Description
[31:3]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[2:0]	DDRC_ECC_STATUS_SR	0x0	Bit 0: 1 Indicates the SECDED interrupt is due to a single error. Bit 1: 1 Indicates the SECDED interrupt is due to a double error. Bit 2: Always 1.

Table 2-76. DDRC_ECC_INT_CLR_REG

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	DDRC_ECC_INT_CLR_REG	0x0	This register must be written by the processor when it has read the SECDED error status information. This helps to clear all the SECDED status information, such as error counters and other SECDED registers. The read value of this register is always 0.

2.3 PHY Configuration Register Summary (Ask a Question)

The following table lists the PHY configuration registers.

Table 2-77. PHY Configuration Register Summary

Register Name	Offset	Type	Reset Source	Description
Reserved	0x200 to 0x22C	—	—	—
PHY_DATA_SLICE_IN_USE_CR	0x230	RW	PRESET_N	PHY data slice in use register
Reserved	0x234 to 0x3C8	—	—	—

2.4 PHY Configuration Register Bit Definitions (Ask a Question)

The following table lists the PHY Configuration register bit definitions.

Table 2-78. PHY_DATA_SLICE_IN_USE_CR

Bit Number	Name	Reset Value	Description
[31:5]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[4:0]	REG_PHY_DATA_SLICE_IN_USE	0x0	Data bus width selection for read FIFO RE generation. One bit for each data slice. 1: Data slice is valid. 0: Read data responses are ignored.

Tip: The PHY data slice 0 must always be enabled.

2.5 DDR_FIC Configuration Registers Summary (Ask a Question)

The following table lists the DDR_FIC configuration registers.

Table 2-79. DDR_FIC Configuration Register Summary

Register Name	Address Offset	R/W	Reset Source	Description
DDR_FIC_NB_ADDR_CR	0x400	RW	PRESET_N	Indicates the base address of the non-bufferable address region.
DDR_FIC_NBRWB_SIZE_CR	0x404	RW	PRESET_N	Indicates the size of the non-bufferable address region.
DDR_FIC_BUF_TIMER_CR	0x408	RW	PRESET_N	10-bit timer interface used to configure the time-out register.
DDR_FIC_HPD_SW_RW_EN_CR	0x40C	RW	PRESET_N	Enable write buffer and read buffer register for AHBL initiator1 and initiator2.
DDR_FIC_HPD_SW_RW_INVAL_CR	0x410	RW	PRESET_N	Invalidates write buffer and read buffer for AHBL initiator1 and initiator2.
DDR_FIC_SW_WR_ERCLR_CR	0x414	RW	PRESET_N	Clear bit for error status by AHBL initiator1 and initiator2 write buffer.
DDR_FIC_ERR_INT_ENABLE_CR	0x418	RW	PRESET_N	Used for Interrupt generation.
DDR_FIC_NUM_AHB_MASTERS_CR	0x41C	RW	PRESET_N	Defines whether one or two AHBL 32-bit initiators are implemented in fabric.
DDR_FIC_HPB_ERR_ADDR_1_SR	0x420	RO	PRESET_N	Tag of write buffer for which error response is received is placed in this register.
DDR_FIC_HPB_ERR_ADDR_2_SR	0x424	RO	PRESET_N	Tag of write buffer for which error response is received is placed in this register.
DDR_FIC_SW_ERR_ADDR_1_SR	0x428	RO	PRESET_N	Tag of write buffer for which error response is received is placed in this register.

.....continued				
Register Name	Address Offset	R/W	Reset Source	Description
DDR_FIC_SW_ERR_ADDR_2_SR	0x42C	RO	PRESET_N	Tag of write buffer for which error response is received is placed in this register.
DDR_FIC_HPD_SW_WRB_EMPTY_ SR	0x430	RO	PRESET_N	Indicates valid data in read and write buffer for AHBL initiator1 and initiator2.
DDR_FIC_SW_HPB_LOCKOUT_SR	0x434	RO	PRESET_N	Write and read buffer status register for AHBL initiator1 and initiator2.
DDR_FIC_SW_HPD_WERR_SR	0x438	RO	PRESET_N	Error response register for bufferable write request.
DDR_FIC_LOCK_TIMEOUTVAL_1_CR	0x440	RW	PRESET_N	Indicates the maximum number of cycles a initiator can hold the bus for locked transfer.
DDR_FIC_LOCK_TIMEOUTVAL_2_CR	0x444	RW	PRESET_N	Indicates the maximum number of cycles a initiator can hold the bus for locked transfer.
DDR_FIC_LOCK_TIMEOUT_EN_CR	0x448	RW	PRESET_N	Lock time-out feature enable register.
DDR_FIC_RDWR_ERR_SR	0x460	RO	PRESET_N	Indicates read address of math error register.

2.6 DDR_FIC Configuration Register Bit Definitions (Ask a Question)

The following tables provide the DDR_FIC Configuration register bit definitions.

Table 2-80. DDR_FIC_NB_ADDR_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDR_FIC_NB_ADD	0x0	This indicates the base address of the non-bufferable address region.

Table 2-81. DDR_FIC_NBRWB_SIZE_CR

Bit Number	Name	Reset Value	Description
[31:9]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
8	DDR_FIC_WCB_SZ	0x0	Configures write buffer and read buffer size as per DDR burst size. This port is common for all buffers. Buffers can be configured to 16 byte or 32 byte size. 0: Buffer size is configured to 16 bytes . 1: Buffer size is configured to 32 bytes.
[7:4]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued

Bit Number	Name	Reset Value	Description
[3:0]	DDR_FIC_NUBF_SZ	0x0	This signal indicates the size of the non-bufferable address region. The region sizes are as follows: 0000: None (default) 0001: 64 KB bufferable region 0010: 128 KB bufferable region 0011: 256 KB bufferable region 0100: 512 KB bufferable region 0101: 1 MB bufferable region 0110: 2 MB bufferable region 0111: 4 MB bufferable region 1000: 8 MB bufferable region 1001: 16 MB bufferable region 1010: 32 MB bufferable region 1011: 64 MB bufferable region 1100: 128 MB bufferable region 1101: 256 MB bufferable region 1110: 512 MB bufferable region 1111: 1 GB bufferable region

Table 2-82. DDR_FIC_BUF_TIMER_CR

Bit Number	Name	Reset Value	Description
[31:10]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[9:0]	DDR_FIC_TIMER	0x0	10-bit timer interface is used to configure time-out register. Once timer reaches the time-out value, a flush request is generated by the flush controller in the DDR_FIC. This port is common for all buffers.

Table 2-83. DDR_FIC_HPD_SW_RW_EN_CR

Bit Number	Name	Reset Value	Description
[31:7]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
6	DDR_FIC_M1_REN	0x0	1: Enable read buffer for AHBL initiator1. 0: Disable read buffer for AHBL initiator1.
5	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
4	DDR_FIC_M1_WEN	0x0	1: Enable write buffer for AHBL initiator1. 0: Disable write buffer for AHBL initiator1.
3	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
2	DDR_FIC_M2_REN	0x0	1: Enable read buffer for AHBL initiator2. 0: Disable read buffer for AHBL initiator2.
1	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued			
Bit Number	Name	Reset Value	Description
0	DDR_FIC_M2_WEN	0x0	1: Enable write buffer for AHBL initiator2. 0: Disable write buffer for AHBL initiator2.

Table 2-84. DDR_FIC_HPD_SW_RW_INVAL_CR

Bit Number	Name	Reset Value	Description
[31:7]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
6	DDR_FIC_flshM1	0x0	1: Flush read buffer for AHBL initiator1 0: Default
5	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
4	DDR_FIC_invalid_M1	0x0	1: Invalidate write buffer for AHBL initiator1 0: Default
3	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
2	DDR_FIC_flshM2	0x0	1: Flush write buffer for AHBL initiator2 0: Default
1	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	DDR_FIC_invalid_M2	0x0	1: Invalidate read buffer for AHBL initiator2 0: Default

Table 2-85. DDR_FIC_SW_WR_ERCLR_CR

Bit Number	Name	Reset Value	Description
[31:9]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
8	DDR_FIC_LTO_CLR	0x0	Clear signal to lock time-out interrupt.
[7:5]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
4	DDR_FIC_M2_WR_ERCLR	0x0	Clear bit for error status of AHBL initiator2 write buffer. After it goes High, error status is cleared.
[3:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	DDR_FIC_M1_WR_ERCLR	0x0	Clear bit for error status posted by AHBL initiator1 write buffer. After it goes High, error status is cleared.

Table 2-86. DDR_FIC_ERR_INT_ENABLE_CR

Bit Number	Name	Reset Value	Description
[31:2]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
1	SYR_SW_WR_ERR	0x0	Status bit Goes high when error response is received for a bufferable write request. Goes low when processor serves interrupt and makes clear bit for AHBL initiator1.
0	SYR_HPD_WR_ERR	0x0	Status bit Goes High when error response is received for a bufferable write request. Goes Low when processor serves the interrupt.

Table 2-87. DDR_FIC_NUM_AHB_MASTERS_CR

Bit Number	Name	Reset Value	Description
[31:5]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
4	CFG_NUM_AHB_MASTERS	0x0	Defines whether one or two AHBL 32-bit initiators are implemented in the fabric. 0: One 32-bit AHB initiator implemented in fabric. 1: Two 32-bit AHB initiators implemented in fabric.
[3:0]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

Table 2-88. DDR_FIC_HPB_ERR_ADDR_1_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDR_FIC_M1_ERR_ADD	0x0	32 bits are split into two registers. [15:0] bits of DDR_FIC_M1_ERR_ADD Tag of write buffer for which error response is received is placed in this register. The following values are updated in this register as per buffer size: 16 bytes: 28 bits tag value is loaded to [31:4] and 0000 to [3:0] 32 bytes: Upper 27 bits of tag is loaded to [31:5] and 00000 to [4:0].

Table 2-89. DDR_FIC_HPB_ERR_ADDR_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDR_FIC_M1_ERR_ADD	0x0	32 bits are split into two registers. [31:16] bits of DDR_FIC_M1_ERR_ADD Tag of write buffer for which error response is received is placed in this register. The following values are updated in this register as per buffer size: 16 bytes: 28 bits tag value is loaded to [31:4] and 0000 to [3:0] 32 bytes: Upper 27 bits of tag is loaded to [31:5] and 00000 to [4:0]

Table 2-90. DDR_FIC_SW_ERR_ADDR_1_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDR_FIC_M2_ERR_ADD	0x0	32 bits are split into two registers. Lower 16 bits. Tag of write buffer for which error response is received is placed in this register. The following values are updated in this register as per buffer size: Buffer size: DDR_FIC_M2_ERR_ADD[31:0] 16 bits: TAG, 0000 32 bits: TAG[27:1], 00000

Table 2-91. DDR_FIC_SW_ERR_ADDR_2_SR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	DDR_FIC_M2_ERR_ADD	0x0	32 bits are split into two registers. [31:16] bits of DDR_FIC_M2_ERR_ADD Tag of write buffer for which error response is received is placed in this register. The following values are updated in this register as per buffer size: Buffer size 16 bytes: 28 bit TAG value is loaded to [31:4] and 0000 to [3:0] 32 bytes: upper 27 bits of tag is loaded to [31:5] and 00000 to [4:0]

Table 2-92. DDR_FIC_HPD_SW_WRB_EMPTY_SR

Bit Number	Name	Reset Value	Description
[31:7]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
6	DDR_FIC_M1_RBEMPTY	0x0	1: Read buffer of AHBL initiator1 does not have valid data.
5	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
4	DDR_FIC_M1_WBEMPTY	0x0	1: Write buffer of AHBL initiator1 does not have valid data. 0: Default
3	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
2	DDR_FIC_M2_RBEMPTY	0x0	1: Read buffer of AHBL initiator2 does not have valid data. 0: Default
1	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.

.....continued			
Bit Number	Name	Reset Value	Description
0	DDR_FIC_M2_WBEMPTY	0x0	1: Write buffer of AHBL initiator2 does not have valid data. 0: Default

Table 2-93. DDR_FIC_SW_HPB_LOCKOUT_SR

Bit Number	Name	Reset Value	Description
[31:9] [7:5] [3:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
8	DDR_FIC_LCKTOUT	0x0	Indicates lock counter in arbiter reached its maximum value. Lock counter (20-bit) starts counting when a locked request gets access to a bus and is cleared when the lock signal becomes logic 0.
6	DDR_FIC_M2_WDSBL_DN	0x0	High indicates AHBL initiator2 write buffer is disabled.
4	DDR_FIC_M2_RDSBL_DN	0x0	High indicates AHBL initiator2 read buffer is disabled.
2	DDR_FIC_M1_WDSBL_DN	0x0	High indicates AHBL initiator1 read buffer is disabled.
0	DDR_FIC_M1_RDSBL_DN	0x0	High indicates AHBL initiator1 write buffer is disabled.

Table 2-94. DDR_FIC_SW_HPD_WERR_SR

Bit Number	Name	Reset Value	Description
[31:9]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
8	DDR_FIC_M1_WR_ERR	0x0	Status bit. Goes High when error response is received for a bufferable write request. Goes Low when the processor serves an interrupt and makes a clear bit for AHBL initiator1.
[7:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	DDR_FIC_M2_WR_ERR	0x0	Status bit. Goes High when error response is received for a bufferable write request. Goes Low when processor serves the interrupt.

Table 2-95. DDR_FIC_LOCK_TIMEOUTVAL_1_CR

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:0]	CFGR_LOCK_TIMEOUT_REG	0x0	20 bits are split into two registers. [15:0] bits of CFGR_LOCK_TIMEOUT_REG Lock time-out 20-bit register. Indicates the maximum number of cycles a initiator can hold the bus for locked transfer. If an initiator holds the bus for locked transfer more than the required cycles, an interrupt is generated.

Table 2-96. DDR_FIC_LOCK_TIMEOUTVAL_2_CR

Bit Number	Name	Reset Value	Description
[31:4]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[3:0]	CFGR_LOCK_TIMEOUT_REG	0x0	20 bits are split into two registers. [19:16] bits of CFGR_LOCK_TIMEOUT_REG Lock time-out 20-bit register. Indicates the maximum number of cycles a initiator can hold the bus for locked transfer. If an initiator holds the bus for locked transfer more than the required cycles, an interrupt is generated.

Table 2-97. DDR_FIC_LOCK_TIMEOUT_EN_CR

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	CFGR_LOCK_TIMEOUT_EN	0x0	1: Lock time-out feature is enabled and interrupt is generated. 0: Lock time-out feature is disabled and interrupt is not generated.

Table 2-98. DDR_FIC_RDWR_ERR_SR

Bit Number	Name	Reset Value	Description
[31:6]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[5:0]	DDR_FIC_CFG_RDWR_ERR_SR	0x0	Read address of math error register.

2.7 FDDR SYSREG Configuration Register Summary (Ask a Question)

The following table lists the FDDR SYSREG configuration registers.

Table 2-99. FDDR SYSREG

Register Name	Address Offset	Register Type	Flash	Reset Source	Description
PLL_CONFIG_LOW_1	0x500	RW	P	PRESETN	Comes from SYSREG. Controls the corresponding configuration input of the FPLL.
PLL_CONFIG_LOW_2	0x504	RW	P	PRESETN	Comes from SYSREG. Controls the corresponding configuration input of the FPLL.
PLL_CONFIG_HIGH	0x508	RW	P	PRESETN	Comes from SYSREG. Controls the corresponding configuration input of the FPLL.
FDDR_FACC_CLK_EN	0x50C	RW	P	PRESETN	Enables the clock to the DDR memory controller.
FDDR_FACC_MUX_CONFIG	0x510	RW	P	PRESETN	Selects the standby glitch-free multiplexers within the fabric alignment clock controller (FACC).
FDDR_FACC_DIVISOR_RATIO	0x514	RW	P	PRESETN	Selects the ratio between CLK_A and CLK_DDR_FIC.
PLL_DELAY_LINE_SEL	0x518	RW	P	PRESETN	Selects the delay values to be added to the FPLL.
FDDR_SOFT_RESET	0x51C	RW	P	PRESETN	Soft reset register for FDDR

.....continued					
Register Name	Address Offset	Register Type	Flash	Reset Source	Description
FDDR_IO_CALIB	0x520	RW	P	PRESETN	Configurations register for DDRIO calibration block.
FDDR_INTERRUPT_ENABLE	0x524	RW	P	PRESETN	Interrupt enable register
F_AXI_AHB_MODE_SEL	0x528	RW	P	PRESETN	Selects AXI/AHB interface in the fabric.
PHY_SELF_REF_EN	0x52C	RW	P	PRESETN	Automatic calibration lock is enabled.
FDDR_FAB_PLL_CLK_SR	0x530	RO	—	PRESETN	Indicates the lock status of the FDDR PLL.
FDDR_FPLL_CLK_SR	0x534	RO	—	PRESETN	Indicates the lock status of the Fabric PLL (CLK_BASE_PLL_LOCK).
FDDR_INTERRUPT_SR	0x53C	RO	—	PRESETN	Interrupt status register
FDDR_IO_CALIB_SR	0x544	RO	—	PRESETN	I/O calibration status register
FDDR_FATC_RESET	0x548	RW	P	PRESETN	Reset to fabric portion of the fabric alignment test circuit.
FDDR_DEFAULTS	0x54C	W1P	—	PRESETN	FDDR register configuration reset

2.8 FDDR SYSREG Configuration Register Bit Definitions (Ask a Question)

The following tables provide FDDR SYSREG Configuration register bit definitions.

Table 2-100. PLL_CONFIG_LOW_1

Bit Number	Name	Reset Value	Description
[31:16]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[15:6]	PLL_FEEDBACK_DIVISOR	0x2	Can be configured to control the corresponding configuration input of the FPLL. Feedback divider value (SSE = 0) (binary value + 1: 00000000 = ÷1, ... 1111111111 = ÷ 1,024) Feedback divider value (SSE = 1) (binary value + 1: 0000000 = ÷1, ... 1111111 = ÷ 128)
[5:0]	PLL_REF_DIVISOR	0x1	Can be configured to control the corresponding configuration input of the FPLL. Reference divider value (binary value + 1: 000000 = ÷ 1)

Table 2-101. PLL_CONFIG_LOW_2

Bit Number	Name	Reset Value	Description
[31:4]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
3	PLL_RESET	0x0	0: Releases FDDR PLL from reset 1: Keeps FDDR PLL in reset

.....continued			
Bit Number	Name	Reset Value	Description
[2:0]	PLL_OUTPUT_DIVISOR	0x2	Configures the amount of division to be performed on the internal (multiplied) PLL clock, to generate the DDR clock. Output divider value 000: ÷1 001: ÷2 010: ÷4 011: ÷8 100: ÷16 101: ÷32 It is possible to configure the PLL output divider as ÷1; this setting must not be used when the DDR is operational.

Table 2-102. PLL_CONFIG_HIGH

Bit Number	Name	Reset Value	Description
[31:14]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
13	PLL_PD	0x0	When PD is asserted, the PLL does power-down and outputs are Low. PD has precedence over all other functions.
12	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
11	PLL_BYPASS	0x1	If 1, powers down the PLL core and bypasses it such that PLLOUT tracks REFCK. BYPASS has precedence over RESET. **Note:** Microchip recommends that either BYPASS or RESET are asserted until all configuration controls are set in the desired working value, and the power supply and reference clock are stable within operating range.
[10:7]	PLL_LOCKCNT	0xF	Configured to control the corresponding configuration input of the FPLL. LOCK counter Value 2 ^ (binary value + 5) 0000: 32 1111: 1048576 For the number of reference cycles before LOCK is asserted from LOCK being detected.
[6:4]	PLL_LOCKWIN	0x0	100: 8000 ppm 101: 16000 ppm 010: 2000 ppm 110: 32000 ppm 011: 4000 ppm 111: 64000 ppm Phase error window for Lock assertion as a fraction of divided reference period. Values are at typical PVT only and are not PVT compensated.
[3:0]	PLL_FILTER_RANGE	0x9	PLL filter ranges are follows: x000: BYPASS x001: 10 - 16.8 MHz x010: 16.8 - 26.8 MHz x011: 26.8 - 43 MHz x100: 43 - 69 MHz x101: 69 - 110 MHz x110: 110 - 175 MHz x111: 175 - 200 MHz

Table 2-103. FDDR_FACC_CLK_EN

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	DDR_CLK_EN	0x1	0: Disables the clock to the DDR memory controller. 1: Enables the clock to the DDR memory controller.

Table 2-104. FDDR_FACC_MUX_CONFIG

Bit Number	Name	Reset Value	Description
[31:3]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
2	SRCCLOCK_REF_SEL	0x0	Selects the source of the reference clock to be supplied to the FDDR PLL. 0: 50 MHz RC oscillator selected as the reference clock for the FPLL 1: Fabric clock (CLK_BASE) selected as the reference clock for the FPLL
[1:0]	CLOCK_SRC_SEL	0x0	Selects the standby glitch-free multiplexers within the FACC. This is used to allow one of four possible clocks to proceed through the FACC divider during FACC PLL initialization time. 00: Source clock is driven to ground 01: Source clock is driven from 50 MHz oscillator 10: Source clock is driven from FPLL output clock 11: Source clock is driven from Fabric CCC

Table 2-105. FDDR_FACC_DIVISOR_RATIO

Bit Number	Name	Reset Value	Description
[31:4]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[3:0]	DDR_FIC_DIVISOR	0x0	Selects the ratio between DDR CLOCK and CLK_DDR_FIC. Divider value = (DDR_FIC_DIVISOR + 1) Range is 0x0 (÷1) to 0xF (÷16).

Table 2-106. PLL_DELAY_LINE_SEL

Bit Number	Name	Reset Value	Description
[31:6]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
[5:3]	PLL_FB_DEL_SEL	0x0	Selects the delay values that are added to the FPLL feedback clock before being output to the FPLL. 000: None (Bypass Delay Block) 001: 550 ps 010: 780 ps 011: 1015 ps 100: 1250 ps 101: 1490 ps 110: 1720 ps 111: 1840 ps

.....continued			
Bit Number	Name	Reset Value	Description
[2:0]	PLL_REF_DEL_SEL	0x0	Selects the delay values that are added to the FPLL reference clock before being output to the FPLL. 000: None (Bypass Delay Block) 001: 550 ps 010: 780 ps 011: 1015 ps 100: 1250 ps 101: 1490 ps 110: 1720 ps 111: 1840 ps

Table 2-107. FDDR_SOFT_RESET

Bit Number	Name	Reset Value	Description
[31:2]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
1	FDDR_DDR_FIC_SOFTRESET	0x1	0: Releases DDR_FIC from reset 1: Keeps DDR_FIC in reset
0	FDDR_CTLR_SOFTRESET	0x1	0: Releases FDDR subsystem from reset 1: Keeps FDDR subsystem in reset

Table 2-108. FDDR_IO_CALIB

Bit Number	Name	Reset Value	Description
[31:15]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
14	CALIB_TRIM	0x0	Indicates override of the calibration value from the pc code/programmed code values in the DDRIO calibration block.
13	CALIB_LOCK	0x0	Used in the DDRIO calibration block as an override to lock the codes during intermediate runs. When the logic receives CALIB_INTRPT, it might choose to assert this signal by prior knowledge of the traffic without going through the process of putting the DDR into self refresh.
12	CALIB_START	0x0	Indicates that rerun of the calibration state machine is required in the DDRIO calibration block.
[11:6]	NCODE	0x0	Indicates the DPC override NCODE from Flash in DDRIO calibration. This can be overwritten from the APB interface.
[5:0]	PCODE	0x0	Indicates the PC override PCODE from Flash in the DDRIO calibration block. This can be overwritten from the APB interface.

Table 2-109. FDDR_INTERRUPT_ENABLE

Bit Number	Name	Reset Value	Description
[31:7]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
6	DDR_FIC_INT_ENABLE	0x0	Masking bit to enable DDR_FIC interrupt
5	IO_CALIB_INT_ENABLE	0x0	Masking bit to enable DDR I/O calibration interrupt
4	FDDR_ECC_INT_ENABLE	0x0	Masking bit to enable ECC error interrupt

.....continued			
Bit Number	**Name**	**Reset Value**	**Description**
3	FABRIC_PLL_LOCKLOST_INT_ENABLE	0x0	Masking bit to enable FAB_PLL_LOCK_LOST interrupt
2	FABRIC_PLL_LOCK_INT_ENABLE	0x0	Masking bit to enable FAB_PLL_LOCK interrupt
1	FPLL_LOCKLOST_INT_ENABLE	0x0	Masking bit to enable FPLL_LOCK_LOST interrupt
0	FPLL_LOCK_INT_ENABLE	0x0	Masking bit to enable FPLL_LOCK interrupt

Table 2-110. F_AXI_AHB_MODE_SEL

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	F_AXI_AHB_MODE	0x0	1: AXI interface in the fabric is selected. 0: AHB interface in the fabric is selected.

Table 2-111. PHY_SELF_REF_EN

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	PHY_SELF_REF_EN	0x0	If 1, automatic calibration lock is enabled.

Table 2-112. FDDR_FAB_PLL_CLK_SR

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	FAB_PLL_LOCK	0x0	Indicates the lock status of the FDDR PLL.

Table 2-113. FDDR_FPLL_CLK_SR

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	FPLL_LOCK	0x0	Indicates the lock status of the Fabric PLL (CLK_BASE_PLL_LOCK).

Table 2-114. FDDR_INTERRUPT_SR

Bit Number	Name	Reset Value	Description
[31:7]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
6	DDR_FIC_INT	0x0	Indicates interrupt from DDR_FIC.
5	IO_CALIB_INT	0x0	The interrupt is generated when the calibration is finished. For the calibration after reset, this typically would be followed by locking the codes directly. For in-between runs during functional operation of DDR, the assertion of an interrupt does not ensure lock because the state machine would wait for the ideal time (DRAM self refresh) for locking. This can be used by firmware to insert the ideal time and provides an indication that locked codes are available.
4	FDDR_ECC_INT	0x0	Indicates when the ECC interrupt from the FDDR subsystem is asserted.
3	FABRIC_PLL_LOCKLOST_INT	0x0	This bit indicates that a falling edge event occurred on the fabric PLL LOCK signal. This indicates that the fabric PLL lost lock.

.....continued			
Bit Number	**Name**	**Reset Value**	**Description**
2	FABRIC_PLL_LOCK_INT	0x0	This bit indicates that a rising edge event occurred on the fabric PLL LOCK signal. This indicates that the fabric PLL came into lock.
1	FDDR_PLL_LOCKLOST_INT	0x0	This bit indicates that a falling edge event occurred on the FPLL_LOCK signal. This indicates that the FPLL lost lock.
0	FDDR_PLL_LOCK_INT	0x0	This bit indicates that a rising edge event occurred on the FPLL_LOCK signal. This indicates that the FPLL came into lock.

Table 2-115. FDDR_IO_CALIB_SR

Bit Number	Name	Reset Value	Description
31	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
14	CALIB_PCOMP	0x01	The state of the P analog comparator
13	CALIB_NCOMP	0x01	The state of the N analog comparator
[12:7]	CALIB_PCODE	0x3F	The current PCODE value set on the FDDR DDR I/O bank
[6:1]	CALIB_NCODE	0x3F	The current NCODE value set on the FDDR DDR I/O bank
0	CALIB_STATUS	0x0	This is 1 when the codes are actually locked. For the first run after reset, this would be asserted 1 cycle after CALIB_INTRPT. For in-between runs, this would be asserted only when the DRAM is put into self refresh or there is an override from the firmware (CALIB_LOCK).

Table 2-116. FDDR_FATC_RESET

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	FATC_RESET	0x1	Reset to the fabric portion of the fabric alignment test circuit. 1: Reset active

Table 2-117. FDDR_DEFAULTS

Bit Number	Name	Reset Value	Description
[31:1]	Reserved	0x0	Software must not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit must be preserved across a read-modify-write operation.
0	FDDR_DEFAULTS	0x1	1: The FDDR subsystem registers are reset to Flash configuration values.

3. Appendix: Register Lock Bits Configuration (Ask a Question)

The Register Lock Bits Configuration tool is used to lock SERDES and FDDR configuration registers of RTG4 devices to prevent them from being overwritten by initiators that have access to these registers. Register lock bits are set in a text `(*.txt)` file, which is then imported into the RTG4 Libero project.

3.1 Lock Bit File (Ask a Question)

An initial default lock bit file can be generated by clicking **Generate FPGA Array Data** in the **Design Flow** window.

The default file located at `<proj_location>/designer/<root>/<root>_init_config_lock_bits.txt` can be used to make the required changes.

Tip: Save the file using a different name if you modify the text file to set the lock bits.

3.2 Lock Bit File Syntax (Ask a Question)

A valid entry in the lock bit configuration file is defined as a `<lock_parameters>< lock bit value>` pair format.

The lock parameters are structured as follows:

- Lock bits syntax for a register: `<Physical block name>_<register name>_LOCK`
- Lock bits syntax for a specific field: `<Physical block name>_<register name>_<field name>_LOCK`

Set the lock bit value to 1 to indicate that the register can be written to that is unlocked and to 0 to indicate that the register cannot be written to that is locked. Lines starting with # or; are comments. Empty lines are allowed in the lock bit configuration file.

The following figure shows the lock bit configuration file.

Figure 3-1. Lock Bit Configuration File

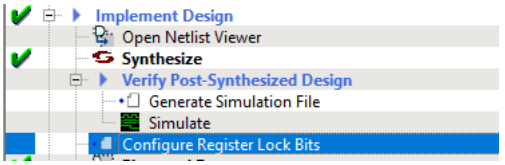
```
File  Edit  Format  View  Help
# Register Lock Bits Configuration File for SERDES(s) and Fabric DDR
# Microchip Technology Inc. - Microchip Libero Software Release v2022.2 (Version 2022.2.0.10)
# Date: Tue Sep 20 13:06:32 2022

# RTG4FDDRC_INIT_C0_0/FDDRC_0/U0/INST_FDDR_IP
 FDDR_E_PLL_CONFIG_LOW_1_FDDR_PLL_DIVR_LOCK          1
 FDDR_E_PLL_CONFIG_LOW_1_FDDR_PLL_DIVF_LOCK          1
 FDDR_E_PLL_CONFIG_LOW_2_FDDR_PLL_DIVQ_LOCK          1
 FDDR_E_PLL_CONFIG_LOW_2_FDDR_PLL_RESET_LOCK         1
 FDDR_E_PLL_CONFIG_LOW_2_FDDR_PLL_TEST_LOCK          1
 FDDR_E_PLL_CONFIG_HIGH_FDDR_PLL_RANGE_LOCK          1
 FDDR_E_PLL_CONFIG_HIGH_FDDR_PLL_LOCKWIN_LOCK        1
 FDDR_E_PLL_CONFIG_HIGH_FDDR_PLL_LOCKCNT_LOCK        1
 FDDR_E_PLL_CONFIG_HIGH_FDDR_PLL_BYPASS_LOCK         1
 FDDR_E_PLL_CONFIG_HIGH_FDDR_PLL_FSE_LOCK            1
 FDDR_E_PLL_CONFIG_HIGH_FDDR_PLL_PD_LOCK             1
 FDDR_E_DDR_CLK_EN_DDR_CLK_EN_LOCK                   1
 FDDR_E_CLOCK_SOURCE_CLOCK_SRC_SEL_LOCK              1
 FDDR_E_CLOCK_SOURCE_CLOCK_REF_SEL_LOCK              1
 FDDR_E_FDDR_FDDR_DIVISOR_RATIO_FIC64_DIVISOR_LOCK   1
 FDDR_E_PLL_DELAY_LINE_SEL_LOCK                      1
```

3.3 Locking and Unlocking a Register (Ask a Question)

A register can be locked or unlocked by setting the appropriate lock bit value in the lock bit configuration `.txt` file.

1. Browse to locate the lock bit configuration `.txt` file.
2. Do one or both of the following:
   - Set the lock bit value to 0 for the registers you want to lock.
   - Set the lock bit value to 1 for the registers you want to unlock.
3. Save the file, and import the file into the project (`Design Flow window > Configure Register Lock Bits`).
4. Regenerate the bit stream.

Figure 3-2. Register Lock Bit Settings Window



4. Revision History (Ask a Question)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Table 4-1. Revision History

Revision	Date	Description
B	08/2023	The following is the list of changes in revision B of the document: • Added a note in 1.5.4. SECDED to describe how to handle uncorrectable 2-bit errors.
A	12/2022	The following is the list of changes in revision A of the document: • The document was migrated to the Microchip template. • The document number was updated to DS50003412 from 50200573. • Following sections are updated: – 1.7.3. Reset Sequence: Updated the section. – 1.7.3.1. DDR I/O Calibration: Updated the section. – Figure 1-14: Updated the figure.
9	—	The following is the list of changes in revision 9 of the document: • Following sections are updated: – 1.7.5.1. Write-Leveling: Updated the section. – 1.12. Simulating FDDR Subsystem: Added note.
8	—	The following is the list of changes in revision 8 of the document: • Decreased RTG4 memory density support of Fabric Double-Data Rate (FDDR) controller from 4 GB to 2 GB. This limits the address access to 2 GB support for the East and West FDDR controller. For more information, see Customer Notification (CN). • 1.5.4. SECDED: Updated. • 1.7.3.1. DDR I/O Calibration: Updated.
7	—	The following is the list of changes in revision 7 of the document: • AC435 is merged into AC439. Removed all AC435 links throughout the document and replaced them with AC439. • Most of the PHY configuration registers have been reserved. • 1.7. Initialization: Updated. • 1.7.6. DDR Memory Initialization Time: Updated. • Table 2-5: Added more information about the REG_DDRC_REFRESH_TO_X32 field. • Table 2-15: Added the delay value for DDR3 in the description of the REG_DDRC_PRE_CKE_X1024 field. • Table 2-102: Removed the invalid settings 500 ppm and 1000 ppm for the PLL_LOCKWIN field of the PLL_CONFIG_HIGH Configuration register. • 3. Appendix: Register Lock Bits Configuration: Added.

.....continued

Revision	Date	Description
6	—	The following is the list of changes in revision 6 of the document: • The following registers are removed: – REG_PHY_FIFO_WE_SLAVE_RATIO – DDRC_DFI_CTRLUPD_TIME_INTERVAL_CR – PHY_DLL_LOCK_DIFF_CR • Description for register FDDR_FAB_PLL_CLK_SR was updated in Table 2-99 and Table 2-112. • Description for register FDDR_FPLL_CLK_SR was updated in Table 2-99 and Table 2-113.
5	—	The following is the list of changes in revision 5 of the document: • Table 1-11: Updated. • Table 1-15: Updated.
4	—	The following is the list of changes in revision 4 of the document: • 1.7.5.1. Write-Leveling: Updated. • 1.7.5.2. Read-Leveling: Updated.
3	—	The following is the list of changes in revision 3 of the document: • Updated the document to the new template. • 4. I/O Utilization: Updated. • 1.7.3.1. DDR I/O Calibration: Updated. • 1.15.2. Self Refresh: Updated.
2	—	The following is the list of changes in revision 2 of the document: • 1.1.1. FPLL: Updated section. • Table 2-104: Updated table.
1	—	Initial Release.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic

ISBN: 978-1-6683-2949-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Australia - Sydney**	**India - Bangalore**	**Austria - Wels**
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	**China - Beijing**	**India - New Delhi**	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	**Denmark - Copenhagen**
Fax: 480-792-7277	**China - Chengdu**	**India - Pune**	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	**China - Chongqing**	**Japan - Osaka**	**Finland - Espoo**
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	**China - Dongguan**	**Japan - Tokyo**	**France - Paris**
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	**China - Guangzhou**	**Korea - Daegu**	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	**Germany - Garching**
Fax: 678-957-1455	**China - Hangzhou**	**Korea - Seoul**	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Haan**
Tel: 512-257-3370	**China - Hong Kong SAR**	**Malaysia - Kuala Lumpur**	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	**Germany - Heilbronn**
Westborough, MA	**China - Nanjing**	**Malaysia - Penang**	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	**Germany - Karlsruhe**
Fax: 774-760-0088	**China - Qingdao**	**Philippines - Manila**	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	**Germany - Munich**
Itasca, IL	**China - Shanghai**	**Singapore**	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	**China - Shenyang**	**Taiwan - Hsin Chu**	**Germany - Rosenheim**
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	**China - Shenzhen**	**Taiwan - Kaohsiung**	**Israel - Ra'anana**
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Fax: 972-818-2924	**China - Suzhou**	**Taiwan - Taipei**	**Italy - Milan**
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	**China - Wuhan**	**Thailand - Bangkok**	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	**Italy - Padova**
Houston, TX	**China - Xian**	**Vietnam - Ho Chi Minh**	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	**Netherlands - Drunen**
Indianapolis	**China - Xiamen**		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	**China - Zhuhai**		**Norway - Trondheim**
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			**Poland - Warsaw**
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			**Romania - Bucharest**
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			**Spain - Madrid**
Tel: 951-273-7800			Tel: 34-91-708-08-90
Raleigh, NC			Fax: 34-91-708-08-91
Tel: 919-844-7510			**Sweden - Gothenberg**
New York, NY			Tel: 46-31-704-60-40
Tel: 631-435-6000			**Sweden - Stockholm**
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			**UK - Wokingham**
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			
Fax: 905-695-2078			