DDR Memory Controller Driver API Reference

User Guide

FPGA-TN-02379-1.3

October 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviations	Definition
API	Application Programming Interface
BLTS	Bit-Level Trim Sweep
CA	Command and Address
CBT	Command Bus Training
CK	Clock
CKE	Clock Enable
CS	Chip Select
CSR	Control and Status Register
DBI	Data Bus Inversion
DDR PHY	Double Data Rate Physical Layer
DDR4	Double Data Rate Generation 4
DQ	Data
DQS	Data Strobe
ECLKSYNC	Edge Clock Synchronization
EN	Enable
FPGA	Field Programmable Gate Array
INIT	Initialization
IP	Intellectual Property (FPGA)
JEDEC	Joint Electron Device Engineering Council
LPDDR4	Low Power Double Data Rate Generation 4
MC	Memory Controller
MR	Mode Register
ODT	On-Die Termination
PLL	Phase-Locked Loop
REG	Register
SCL	Self-Calibrating Logic
SCLK	System Clock
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SoC	System on Chip
TRN	Training
VREF	Reference Voltage

1. Introduction

Lattice Semiconductor's DDR Memory Controller Driver provides a comprehensive solution for interfacing with DDR SDRAM. It includes support for LPDDR4 and DDR4 controllers, DDR PHY, and integrated clocking and training logic.

For detailed information on the IP core and register descriptions, refer to the DDR Memory Controller IP Core (FPGA-IPUG-02208) and DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195).

1.1. Purpose

The LPDDR4/DDR4 SDK provides a set of C-language APIs that enable access to specific hardware and software features of Lattice Avant™ devices. This guide serves as a reference for developers, detailing the available APIs and their function flows. The APIs apply to either LPDDR4 or DDR4, depending on the controller present on the Avant FPGA board. For example, the Avant E70 development board uses an LPDDR4 controller, while the customized Avant X70 development board features a DDR4 controller.

1.2. Audience

This document is intended for embedded system designers and software developers working with Lattice Avant FPGA devices. It assumes familiarity with embedded systems and FPGA design principles.

1.3. Driver and IP Compatibility

Table 1.1. Driver and IP Version

Driver version	DDR Memory Controller IP version
25.01.01	2.6.1

Refer to the DDR Memory Controller IP Release Notes (FPGA-RN-02033) for more information on the driver and IP versions.

Table 1.2. Quick Facts of Driver Tested Environment

Driver tested on HW Device	Avant Family	Avant-E70 (LPDDR4)	Customized Avant-x70 (DDR4)
Tool Version	Lattice Propel™ Builder	2025.1	2025.1
	Lattice Propel SDK	2025.1	2025.1
	Lattice Radiant™ Software	2025.1	2025.1
	Radiant Programmer	2025.1	2025.1

2. API Description

This section outlines the available APIs for the DDR Memory Controller. Each API is described with its function prototype, purpose, parameters, and return values.

2.1. ddr_mc_init()

This API initializes the DDR4 or LPDDR4 memory controller. This function configures the controller using the provided base address, detects the DDR type, number of ranks, and operating frequency, and initiates memory training. It monitors the training process and returns a status code indicating success or the specific error encountered.

```
Unsigned int ddr_mc_init(ddr_mc *instancePtr, unsigned int base_addr);
```

Table 2.1. ddr_mc_init()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	Success: DDR_TRAINING_NO_ERR Error codes: RANK0_CBT_ERR RANK0_WR_LVL_ERR RANK0_RD_TRN_ERR RANK0_WR_TRN_ERR RANK0_SCL_ERR RANK0_BLTS_ERR RANK1_CBT_ERR RANK1_WR_LVL_ERR RANK1_RD_TRN_ERR RANK1_WR_TRN_ERR RANK1_SCL_ERR RANK1_BLTS_ERR
In	base_addr	Base address to be assigned to controller.	

2.2. ddr_mc_GetFeatureControlReg()

This API reads the configuration settings from the Feature Control Register. This register reflects the modes of operation specified by the attributes selected during IP configuration and stores the result in the reg_data pointer. These attributes are set during IP configuration and cannot be modified during runtime. The CPU reads this register to determine the modes of operation. Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) for the register description.

```
unsigned int ddr_mc_GetFeatureControlReg(ddr_mc *instancePtr, unsigned int *reg_data);
```

Table 2.2. ddr_mc_GetFeatureControlReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success
Out	reg_data	The current values are read from the Feature Control Register.	

2.3. ddr_mc_GetSettingReg()

The Settings Register controls DDR write and read latencies based on the attributes selected during IP configuration. This API reads latencies from the settings register parameters. Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) for the register description.

```
unsigned int ddr_mc_GetSettingReg(ddr_mc *instancePtr, unsigned int *reg_data);
```

Table 2.3. ddr_mc_GetSettingReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success
Out	reg_data	The current values are read from the Settings Register.	

2.4. ddr_mc_GetPhyClockReg()

This API reads the RO parameters of PHY_CLOCK_REG. The training CPU uses this register during clock frequency changes. The host CPU reads this register to determine the clock frequency and lock status; it does not write to this register. Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) for the register description.

```
unsigned int ddr_mc_GetPhyClockReg(ddr_mc *instancePtr, unsigned int *reg_data);
```

Table 2.4. ddr_mc_GetPhyClockReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success
Out	reg_data	The current values are read from the PHY_CLOCK_REG.	

2.5. ddr_mc_TrainingErrorInterruptEnable()

This API enables training error interrupts. The Interrupt Enable Register lists all configurable interrupts within the Memory Controller IP. Refer to the IP user guide for the register description.

```
unsigned int ddr_mc_TrainingErrorInterruptEnable(ddr_mc *instancePtr);
```

Table 2.5. ddr_mc_TrainingErrorInterruptEnable()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success

2.6. ddr_mc_TrainingErrorInterruptDisable()

This API disables training error interrupts. The Interrupt Enable Register lists all configurable interrupts within the Memory Controller IP. Refer to the IP user guide for the register description.

```
unsigned int ddr_mc_TrainingErrorInterruptDisable(ddr_mc *instancePtr);
```

Table 2.6. ddr_mc_TrainingErrorInterruptDisable()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success

2.7. ddr_mc_TrainingDoneInterruptEnable()

This API enables training done interrupts. The Interrupt Enable Register lists all configurable interrupts within the Memory Controller IP. Refer to the IP user guide for the register description.

```
unsigned int ddr_mc_TrainingDoneInterruptEnable(ddr_mc *instancePtr);
```

Table 2.7. ddr_mc_TrainingDoneInterruptEnable()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success

2.8. ddr_mc_TrainingDoneInterruptDisable()

This API disables training done interrupts. The Interrupt Enable Register lists all configurable interrupts within the Memory Controller IP. Refer to the IP user guide for the register description.

```
unsigned int ddr_mc_TrainingDoneInterruptDisable(ddr_mc *instancePtr);
```

Table 2.8. ddr_mc_TrainingDoneInterruptDisable()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success

2.9. ddr_mc_GetTrainingOperationReg()

This API reads the configuration settings from the Training Operation Register. This register controls memory initialization and training. Disable these registers (except write_lvl_en) during simulation to skip lengthy initialization and training. There is no enable bit for the self-calibrating logic because it is always enabled. Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) for the register description.

```
unsigned int ddr_mc_GetTrainingOperationReg(ddr_mc *instancePtr, unsigned int
*reg_data);
```

Table 2.9. ddr_mc_GetTrainingOperationReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success
Out	reg_data	The current values are read from the Training Operation Register.	

2.10. ddr_mc_GetStatusReg()

This API reads the status register. The memory controller writes to this register to communicate status to the host CPU. Refer to the IP user guide for the register description.

```
unsigned int ddr_mc_GetStatusReg(ddr_mc *instancePtr, unsigned int *reg_data);
```

Table 2.10. ddr_mc_GetStatusReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success
Out	reg_data	The current values are read from the Status Register.	

2.11. ddr_mc_GetDDRType()

This API initializes and identifies the DDR type configured in the SoC by reading the FEATURE_CTRL_REG (Feature Control Register). Based on the register value, it determines the memory type and returns one of the following enumerations: DDR_TYPE_LPDDR4, DDR_TYPE_DDR4, or DDR_TYPE_UNKNOWN.

```
unsigned int ddr_mc_GetDDRType(ddr_mc *instancePtr, unsigned int *ddr_type);
```

Table 2.11. ddr_mc_GetDDRType()

In/Out	Parameter	Description	Returns
In	instancePtr	Handles the LPDDR4/DDR4 structure.	0: failure 1: success
Out	ddr_type	Detected DDR type.	

3. Function Call Flow Diagrams

3.1. ddr_mc_init()

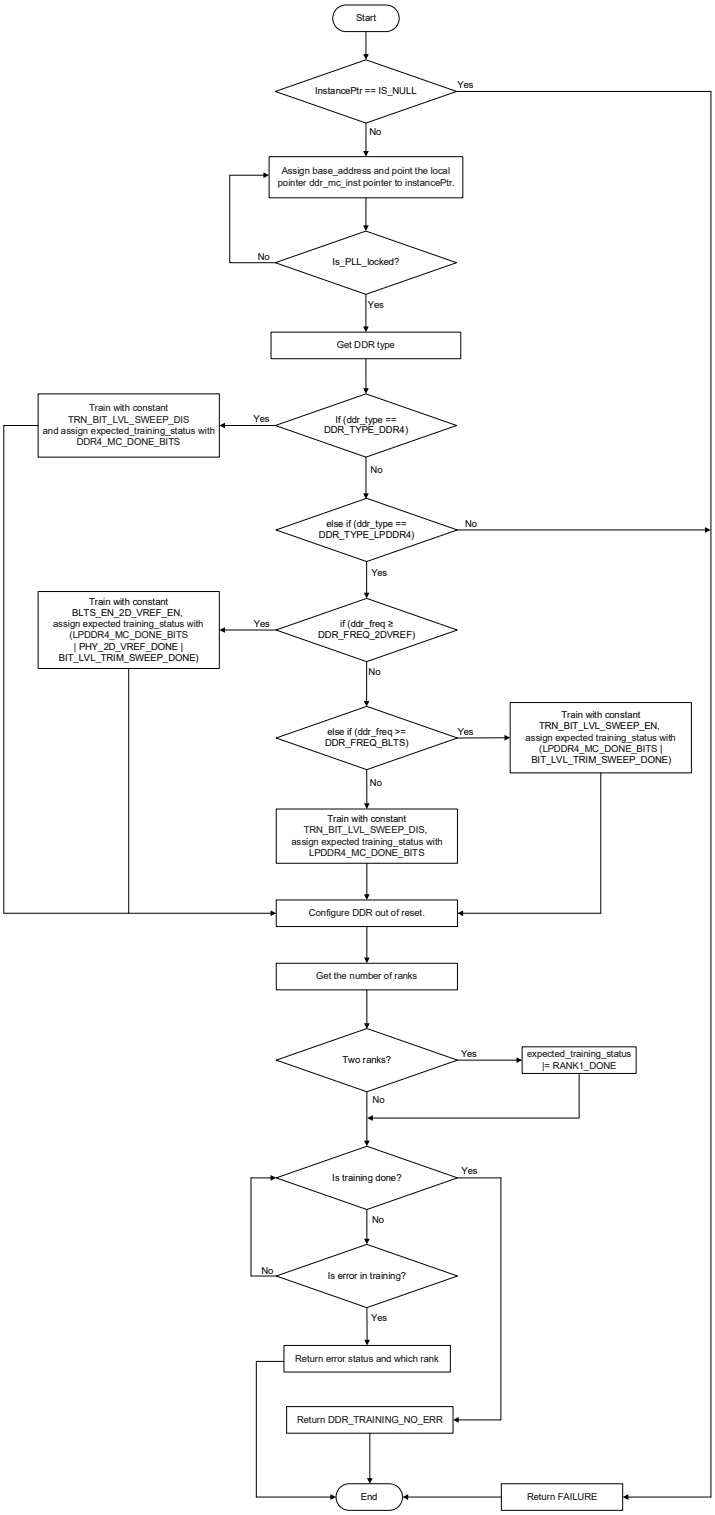

Figure 3.1. ddr_mc_init()

3.2. ddr_mc_GetFeatureControlReg()

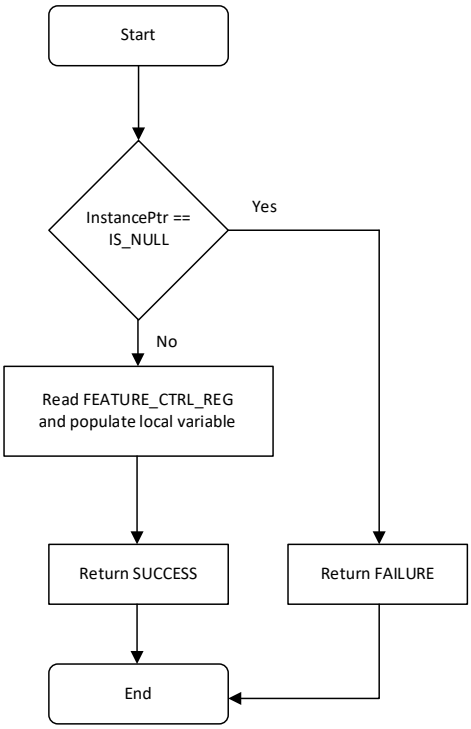

Figure 3.2. ddr_mc_GetFeatureControlReg()

3.3. ddr_mc_GetSettingReg()

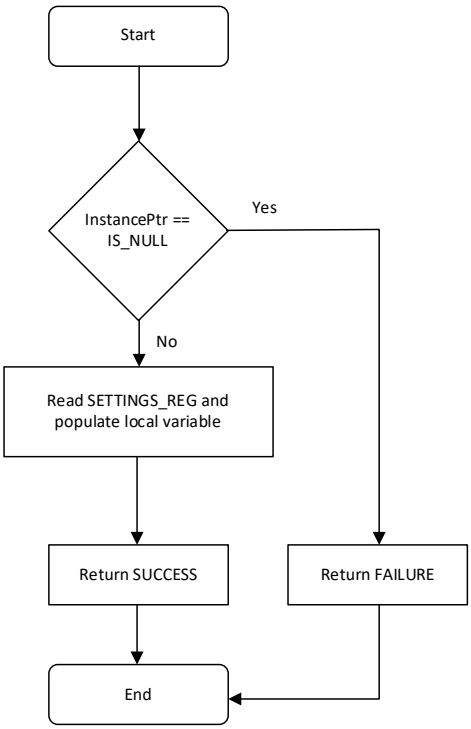

Figure 3.3. ddr_mc_GetSettingReg()

3.4. ddr_mc_GetPhyClockReg()

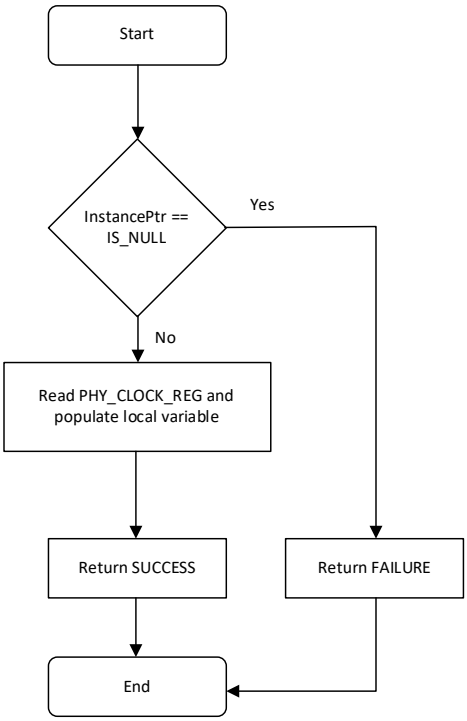

Figure 3.4. ddr_mc_GetPhyClockReg()

3.5. ddr_mc_TrainingErrorInterruptEnable()

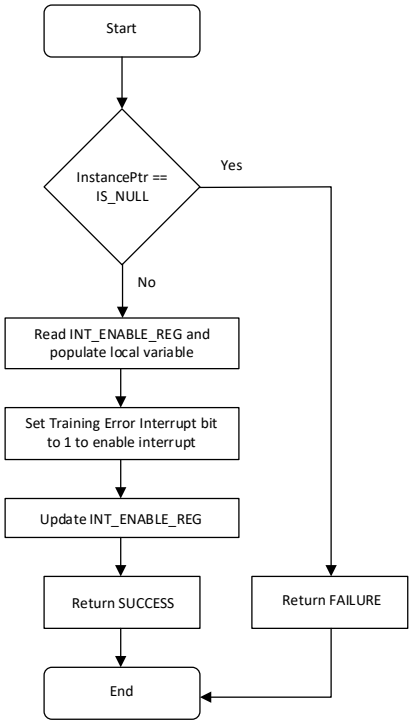

Figure 3.5. ddr_mc_TrainingErrorInterruptEnable()

3.6. ddr_mc_TrainingErrorInterruptDisable()

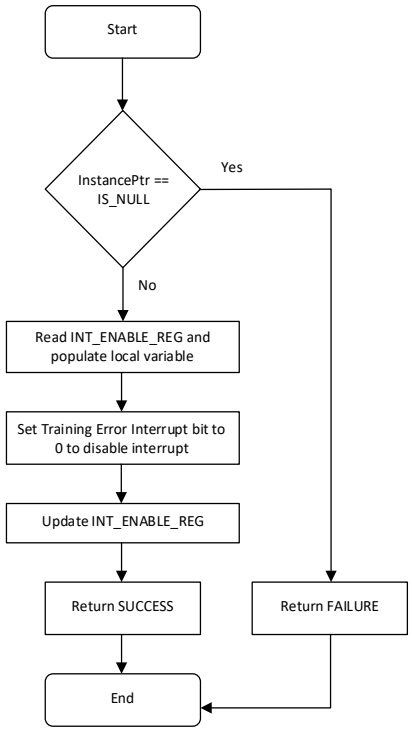

Figure 3.6. ddr_mc_TrainingErrorInterruptDisable()

3.7. ddr_mc_TrainingDoneInterruptEnable()

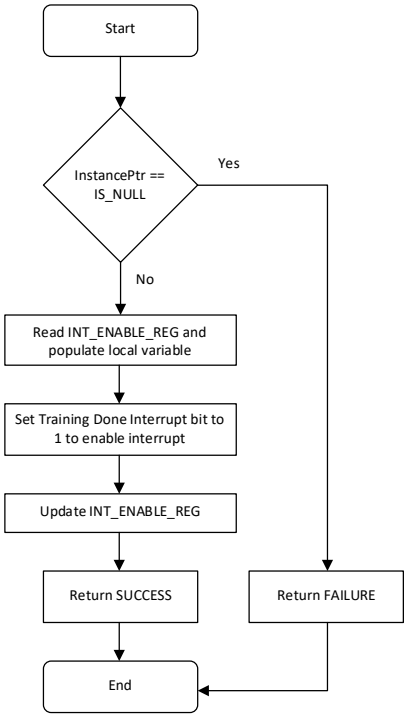

Figure 3.7. ddr_mc_TrainingDoneInterruptEnable()

3.8. ddr_mc_TrainingDoneInterruptDisable()

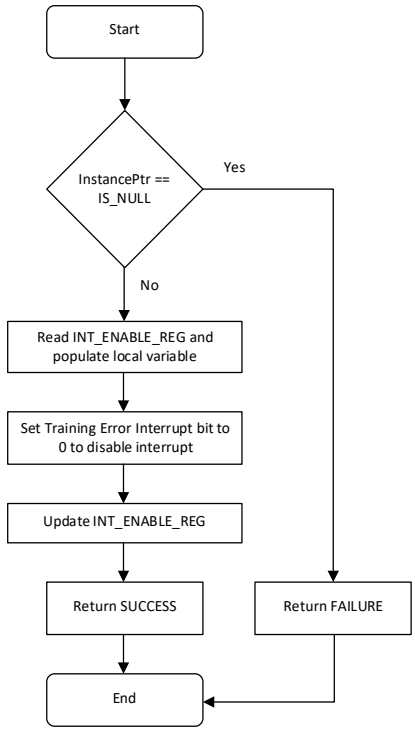

Figure 3.8. ddr_mc_TrainingDoneInterruptDisable()

3.9. ddr_mc_GetTrainingOperationReg()

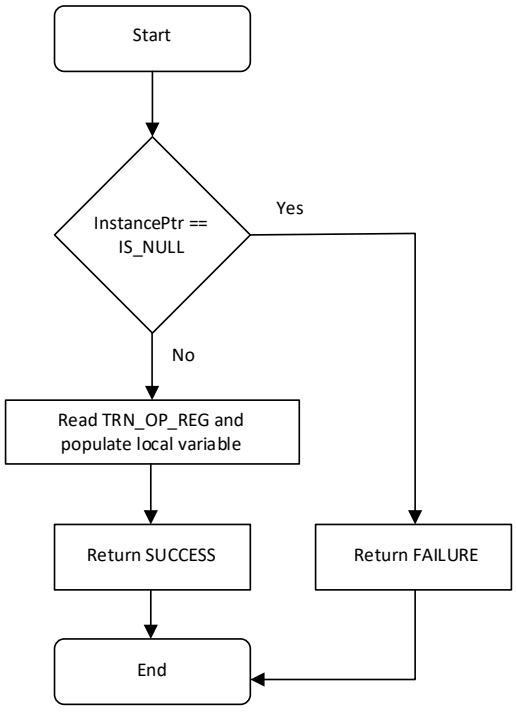

Figure 3.9. ddr_mc_GetTrainingOperationReg()

3.10. ddr_mc_GetStatusReg()

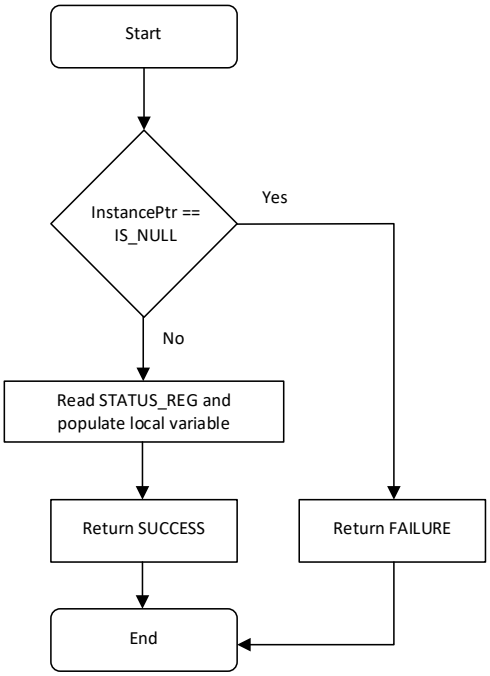

Figure 3.10. ddr_mc_GetStatusReg()

3.11. ddr_mc_GetDDRType()

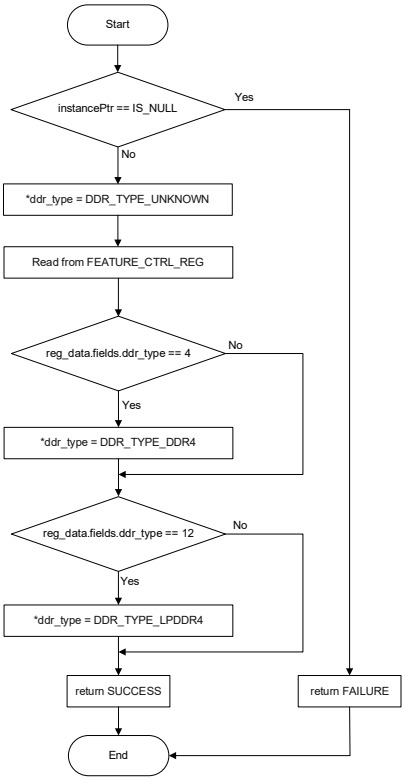

Figure 3.11. ddr_mc_GetDDRType()

4. API Data Structures

4.1. Struct DDR_MC

Table 4.1. DDR_MC Parameters

Data Type	Struct Member	Description
unsigned int	ddr_mc_base_address	Base address of the ddr_mc memory controller.

4.2. Union and Struct Feature_ctrl_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.2. Feature_ctrl_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the feature_ctrl_reg_t register.
unsigned int	reg	This variable accesses the feature_ctrl_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	ecc_en	The ecc_en enables the sideband ECC function. This mode is based on the enable sideband ECC attribute.
unsigned int	dbi_en	The dbi_en enables the data bus inversion function to reduce toggling of the DDR data signal on the board, thus improving signal integrity and reducing dynamic power consumption. The DBI is enabled only on the read path and is always disabled on the write path to support masked write function. This mode is based on the enable read DBI attribute.
unsigned int	fea_rsvd_1	Reserved
unsigned int	gear_ratio	The gear_ratio specifies the ratio of clock frequency between the DDR clock domain and the system clock domain. This is based on the gearing ratio attribute. • 0 – 4:1 gearing: ddr_ck_o frequency = 2 x sclk_o frequency • 1 – 8:1 gearing: ddr_ck_o frequency = 4 x sclk_o frequency
unsigned int	fea_rsvd_2	Reserved
unsigned int	ddr_type	The ddr_type specifies the DDR standard implemented by the memory controller IP core. This is based on the interface type attribute. • 3 – Interface Type = DDR3 • 4 – Interface Type = DDR4 • 12 – Interface Type = LPDDR4
unsigned int	ddr_width	The ddr_width specifies the bit width of the DDR data bus as follows: • 0 – DDR Bus Width = 8 bits • 1 – DDR Bus Width = 16 bits • 2 – Reserved • 3 – DDR Bus Width = 32 bits or 40 bits • 4 – 4'h6: reserved • 7 – DDR Bus Width = 64 bits or 72 bits When the enable sideband ECC is checked, the DDR bus width can be [40, 72] but the ddr_width is [3, 7]. The ECC byte is not counted in this field.

Data Type	Union Member	Description
unsigned int	num_ranks	The num_ranks specifies the number of DDR ranks as follows: • 0 – Single rank • 1 – Dual rank
unsigned int	fea_rsvd_3	Reserved
unsigned int	reset_init_by_phy	This enables the logic and routine that drives the ddr_reset_n_o and ddr_cke_o signals during memory initialization, as well as the initial mode register writes and ZQ calibration right after initialization. This is fixed to 1 for the first release. • 0 – Memory controller resets and initializes the memory • 1 – PHY resets and initializes the memory
unsigned int	fea_rsvd_4	Reserved

4.3. Union and Struct phy_clk_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.3. Clk_chng_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the phy_clk_reg_t register.
unsigned int	reg	This variable accesses the phy_clk_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	disable_clkophy	Set this to 1 to disable the PLL output clock going to the DDRPHY and ECLKSYNC hard IPs.
unsigned int	pll_lock	The cmd_freq reflects the DDR memory frequency attribute value. The training routine uses this value to trim the initial delay settings of the DDR signals.
unsigned int	pll_reset	The training CPU uses this to reset the PLL during clock frequency changes so that the PLL locks on the new frequency.
unsigned int	prim_rst_en_cfc	This enables the reset of the DDR primitives during clock frequency changes.
unsigned int	pll_refclk	This returns the PLL reference clock frequency in MHz.
unsigned int	phy_clk_rsvd	Reserved

4.4. Union and Struct reset_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.4. reset_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the reset_reg_t register.
unsigned int	reg	This variable accesses the reset_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	cpu_reset_n	This resets the input to the Training CPU. This register is in the reset state at power-on. The host CPU deasserts this register to start memory initialization and training. Upon training completion, the internal CPU writes to this register to return the memory training engine to the reset state, thus saving power consumption. This register does not reset the CSR.
unsigned int	res_rsvd	Reserved

4.5. Union and Struct settings_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.5. Settings_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the settings_reg_t register.
unsigned int	reg	This variable accesses the settings_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	write_latency	The write_latency specifies the number of DDR clock cycles from the write command to the first write data.
unsigned int	read_latency	The read_latency specifies the number of DDR clock cycles from the read command to the first read data.
unsigned int	cmd_freq	The cmd_freq reflects the DDR memory frequency attribute value. The training routine uses this value to trim the initial delay settings of the DDR signals.
unsigned int	set_rsvd	Reserved

4.6. Union and Struct int_status_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.6. Int_status_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the int_status_reg_t register.
unsigned int	reg	This variable accesses the int_status_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	trn_done_int	Training Done Interrupt. This interrupt bit asserts when initialization and training have completed successfully. • 0 – No interrupt • 1 – Interrupt pending
unsigned int	trn_err_int	Training Error Interrupt. This interrupt bit asserts when the Training Engine encounters an error during training. Read STATUS_REG to determine the specific error. • 0 – No interrupt • 1 – Interrupt pending
unsigned int	int_sts_rsvd	Reserved

4.7. Union and Struct status_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.7. Status_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the status_reg_t register.
unsigned int	reg	This variable accesses the status_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	init_done	This asserts when the PHY initialization is complete.
unsigned int	cbt_done	This asserts when the command bus training completes.
unsigned int	write_lvl_done	This asserts when the write leveling completes.
unsigned int	read_trn_done	This asserts when the read training completes.
unsigned int	write_trn_done	This asserts when the write training completes.
unsigned int	in_self_refresh	This asserts when the memory enters self-refresh mode.
unsigned int	scl_done	This asserts when the self-calibrating logic completes successfully.
unsigned int	rank_0_done	This asserts when all training for rank 0 completes without error.
unsigned int	rank_1_done	This asserts when all training for rank 1 completes without error.
unsigned int	cbt_err	This asserts when a failure occurs during command bus training.
unsigned int	write_lvl_err	This asserts when a failure occurs during write leveling.
unsigned int	read_trn_err	This asserts when a failure occurs during in read training.
unsigned int	write_trn_err	This asserts when a failure occurs during write training.
unsigned int	scl_err	This asserts when a failure occurs in self-calibrating logic.
unsigned int	blts_err	This asserts when a failure occurs during bit-level trim sweep.
unsigned int	err_on_rank	This specifies the rank where the training error occurred. • 2'bx1 – Training error occurred on rank 0 • 2'b1x – Training error occurred on rank 1
unsigned int	bit_lvl_trim_sweep_done	This asserts when the bit-level trim sweep completes.
unsigned int	phy_2d_vref_done	This asserts when 2D VREF training completes.
unsigned int	trn_sts_rsvd2	Reserved
unsigned int	retraining_count	If a command bus training, write leveling, or read DQ-bit leveling error occurs, the Training CPU recalibrates the DDR/LPDDR interface again with updated training settings. The retraining_count tracks the number of retries by the Training CPU attempts to successfully pass the DDR/LPDDR training routine, with a maximum number of five retrains.
unsigned int	trn_sts_rsvd3	Reserved

4.8. Union and Struct int_enable_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.8. int_enable_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the int_enable_reg_t register.
unsigned int	reg	This variable accesses the int_enable_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	trn_done_en	Training Done Interrupt Enable. • 0 – Interrupt disabled • 1 – Interrupt enabled
unsigned int	trn_err_en	Training Error Interrupt Enable. • 0 – Interrupt disabled • 1 – Interrupt enabled
unsigned int	int_en_rsvd	Reserved

4.9. Union and Struct trn_op_reg_t

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description.

Table 4.9. trn_op_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the trn_op_reg_t register.
unsigned int	reg	This variable accesses the trn_op_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	init_en	This feature shortens initialization for simulation. • 0 – Initialization is reduced. For example, shortened reset time and CKE low time. • 1 – Initialization follows the LPDDR4/DDR4 JEDEC Standard.
unsigned int	cbt_en	This feature enables Command Bus Training (CBT) during initialization and training. CBT performs CA_VREF programming and aligns the CS/CA and CK for high frequency operation. The command bus training always follows the LPDDR4/DDR4 standard. This bit disables clock frequency changes for faster simulation, as the process requires extended simulation time. • 0 – The clock frequency change is not performed during CBT. • 1 – The clock frequency change is performed during CBT.
unsigned int	write_lvl_en	The write_lvl_en enables write leveling during initialization and training. Write leveling compensates for CK-DQS timing skews. See the write leveling section in the IP user guide[1] for details. Write leveling is always performed during training.
unsigned int	read_trn_en	The read_trn_en enables read DQ bit leveling. This setting optimizes read DQ relative to read DQS to improve the valid data window for reads. See the read DQ bit leveling section in IP user guide[1] for details.
unsigned int	write_trn_en	The write_trn_en enables write DQ bit leveling. Write training optimizes the delay of DQ relative to DQS to improve the valid data window for writes. See the write DQ bit leveling section for details in IP user guide for details.

Data Type	Union Member	Description
unsigned int	ca_vref_training_en	The ca_vref_trn_en enables the memory's CA Vref training. When enabled, the DDRPHY performs CA training across multiple CA_Vref points and selects the optimal CA_Vref value. This bit is currently unused.
unsigned int	mc_verf_training_en	The mc_vref_trn_en enables the DDRPHY's DQ Vref training. When enabled, the DDRPHY performs write training across multiple Vref points of the FPGA's I/O and selects the optimal Vref value.
unsigned int	mem_vref_training_en	The mem_vref_trn_en enables the memory's DQ Vref training. When enabled, the DDRPHY performs write training across multiple memory Vref points and selects the optimal Vref value.
unsigned int	bit_lvl_trim_sweep_en	The bit_lvl_trim_sweep centers the DQ valid window at the DQS edge for each DQ bit. See bit-level trim sweep section in the IP user guide[1] for details.
unsigned int	phy_2d_vref_en	The phy_2d_vref_en improves performance by addressing voltage variation both horizontally and vertically through 2D VREF training. This setting requires bit_lvl_trim_sweep_en to be set to 1. Refer to the VREF Training and Bit-Level Trim Sweep section in the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) for more details.
unsigned int	trn_op_rsvd	Reserved

Note:

1. Please refer to DDR Memory Controller IP Core (FPGA-IPUG-02208) and DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195).

4.10. Union and Struct odt_settings_reg

Refer to the DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195) IP user guide for the detailed register description. This register is currently unused for DDR3L and DDR4. The training CPU reads this register and sets the LPDDR4/DDR4 memory mode register MR11 and MR22 during memory initialization.

Table 4.10. odt_settings_reg Parameters

Data Type	Union Member	Description
Struct	fields	This variable accesses the status_reg_t register.
unsigned int	reg	This variable accesses the status_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	dq_odt_val	Specifies the ODT settings to be written to the LPDDR4 memory mode register MR11.
unsigned int	odt_rsvd1	Reserved
unsigned int	ca_odt_val	Specifies the ODT settings to be written to the LPDDR4 memory mode register MR11.
unsigned int	odt_rsvd2	Reserved
unsigned int	soc_odt_val	Specifies the ODT settings to be written to the LPDDR4 memory mode register MR22.
unsigned int	odte_ck	
unsigned int	odte_cs	
unsigned int	odte_ca	
unsigned int	X8ODTD	
unsigned int	odtr_pdds	Specifies the pull-down drive strength settings to be written to the LPDDR4 memory mode register MR3.
unsigned int	odt_rsvd3	Reserved

5. API Enum

5.1. Enum

Table 5.1. DDR_MC_RET Variables

Enum Member	Enum Decimal Value
DDR_TYPE_LPDDR4	0
DDR_TYPE_DDR4	1
DDR_TYPE_UNKNOWN	2

6. API Variables

Accesses the base address of the LPDDR4/DDR4 IP from the Lattice Propel Builder configuration table (sys_platform.h).

```
static ddr_mc *ddr_mc_inst;
```

7. API Macros

Table 7.1. API Macros Description

Macro	Description
#define DDR_MC_DRV_VER	Specifies the LPDDR4/DDR4 driver version.
#define FEATURE_CTRL_REG	Register address of the Feature Control Register.
#define RESET_REG	Register address of the Reset Register.
#define SETTINGS_REG	Register address of the Settings Register.
#define PHY_CLOCK_REG	Register address of the Initialization Control Register.
#define INT_STATUS_REG	Register address of the Interrupt Status Register.
#define INT_ENABLE_REG	Register address of the Interrupt Enable Register.
#define INT_SET_REG	Register address of the Interrupt Set Register (for debug purpose only).
#define TRN_OP_REG	Register address of the Training Operation Register.
#define STATUS_REG	Register address of the Status Register.
#define MRW_CTRL_REG	Mode Register Write Control Register.
#define CK_ADRCTRL_TRIM	Clock/Address/Control Trim Setting Register to specify delay.
#define ODT_SETTING_REG	Register address of the ODT Settings register
#define DRAM_VREF_REG	DRAM Vref Register Setting.
#define PHY_VREF_0_3_REG	PHY Vref setting for DQS Groups 0 to 3.
#define PHY_VREF_4_7_REG	PHY Vref setting for DQS Groups 4 to 7.
#define PHY_VREF_8_REG	PHY Vref setting for DQS Group 8.
#define SUCCESS	1
#define FAILURE	0
#define IS_NULL	0
#define OUT_OF_RESET	Out of Reset Value. This value brings training and CPU out of reset.
#define TRN_EN	Training enable bit.
#define TRN_BIT_LVL_SWEEP_EN	Bit Level Sweep training enable.
#define BLTS_EN_2D_VREF_EN	Training enabled with 2D Vref.
#define TRN_BIT_LVL_SWEEP_DIS	Bit Level Sweep training disable.
#define LPDDR4_MC_DONE_BITS	LPDDR4 initialization done bits.
#define DDR_MC_DONE_BITS	DDR4 initialization done bits.
#define DDR_MC_ERR_DONE_BITS	LPDDR4/DDR4 initialization done bits.
#define PLL_LOCK_BIT	LPDDR4/DDR4 PLL lock bit.
#define DDR_FREQ_2DVREF	DDR frequency for Bit Level Sweep training and 2D VREF training.
#define DDR_FREQ_BLTS	DDR frequency for Bit Level Sweep training.
#define PHY_2D_VREF_DONE	Status bit indicates completion of 2D VREF training.
#define BIT_LVL_TRIM_SWEEP_DONE	Status bit indicates completion of bit-level trim sweep.
#define RANK1_DONE	Status bit indicates completion of rank 1.
#define ERROR_ON_RANK0	Error detected on rank 0.
#define ERROR_ON_RANK1	Error detected on rank 1.
#define RANK1_ERROR_INDICATOR_START_POS	Start position for rank 1 error indicator.
#define RANK0_CBT_ERR	Command bus training error on rank 0.
#define RANK0_WR_LVL_ERR	Write levelling error on rank 0.
#define RANK0_RD_TRN_ERR	Read transaction error on rank 0.
#define RANK0_WR_TRN_ERR	Write transaction error on rank 0.
#define RANK0_SCL_ERR	Self-calibrating logic error on rank 0.
#define RANK0_BLTS_ERR	Bit level trim sweep error on rank 0.
#define DDR_TRAINING_NO_ERR	DRR training passed.

Macro	Description
#define RANK1_CBT_ERR	Command bus training error on rank 1.
#define RANK1_WR_LVL_ERR	Write levelling error on rank 1.
#define RANK1_RD_TRN_ERR	Read transaction error on rank 1.
#define RANK1_WR_TRN_ERR	Write transaction error on rank 1.
#define RANK1_SCL_ERR	Self-calibrating logic error on rank 1.
#define RANK1_BLTS_ERR	Bit level trim sweep error on rank 1.

References

- DDR Memory Controller IP Core (FPGA-IPUG-02208)
- DDR Memory Controller IP Release Notes (FPGA-RN-02033)
- DDR Memory PHY IP Module User Guide (FPGA-IPUG-02195)
- Lattice Avant Platform web page
- Lattice Avant-E web page
- Lattice Propel System Design Environment web page
- Lattice Propel Builder 2.1 User Guide (FPGA-UG-02143)
- Lattice Radiant FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.3, October 2025

Section	Change Summary
All	Minor editorial fixes.
Abbreviations in This Document	Updated section contents.

Introduction	• Updated the following in Table 1.1. Driver and IP Version.
	  • Driver version: *25.01.01*
	  • IP version: *2.6.1*
	• Updated Table 1.2. Quick Facts of Driver Tested Environment.
	  • Added Customized Avant-x70 (DDR4) column.
	  • Updated Avant E70 tool version to *2025.1*.

API Description	• Reworked ddr_mc_init() section contents. • Reworked ddr_mc_GetDDRType() section contents.
Function Call Flow Diagrams	Reworked Figure 3.1. ddr_mc_init() and Figure 3.11. ddr_mc_GetDDRType().
API Enum	Reworked section contents.
API Macros	Reworked section contents.

Revision 1.2, June 2025

Section	Change Summary
All	• Merged LPDDR4 and DDR4 IPs, ensuring the naming convention of APIs remains consistent for both LPDDR4 and DDR4. • Updated the register bits according to the latest IP user guide. • Minor editorial fixes.

Introduction	• Updated the Purpose section.
	• Updated Table 1.1. Driver and IP Version:
	  • Driver version: *25.01.00*
	  • DDR Memory Controller IP version: *2.6.0*
	• Updated Table 1.2. Quick Facts of Driver Tested Environment:
	  • Lattice Propel™ Builder, SDK, Radiant, Programmer versions.

Function Call Flow Diagrams	Added API Description *ddr_mc_GetDDRType()* .
API Enum	Updated the Enum_DDR_MC_RET struct.

Revision 1.1, March 2025

Section	Change Summary
All	• Changed the document title from *Avant LPDDR4 Memory Controller Driver* to *DDR Memory Controller Driver* to align with the document title of the IPUG. • Minor editorial fixes.
Introduction	Updated the following in Table 1.1. Driver and IP Version: • Driver version: *24.02.00* • IP version: *2.5.2*

Revision 1.0, August 2024

Section	Change Summary
All	Initial release.

www.latticesemi.com