CertusPro-NX LPDDR4 Memory Controller Driver API Reference

User Guide

FPGA-TN-02378-1.0

August 2024

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
CA	Command and Address
CBT	Command Bus Training
CS	Chip Select
DBI	Data Bus Inversion
DDR PHY	Double Data Rate Physical Layer
DQ	Data
DQS	Data Strobe
FPGA	Field Programmable Gate Array
IP	Intellectual Property (FPGA)
JEDEC	Joint Electron Device Engineering Council
LPDDR4	Low Power Double Data Rate Generation 4
SCLK	System Clock
SDRAM	Synchronous Dynamic Random Access Memory
SDK	Software Development Kit

1. Introduction

The Lattice Semiconductor LPDDR4 Memory Controller for Nexus Devices provides a turnkey solution consisting of a controller, DDR PHY, and associated clocking and training logic to interface with LPDDR4 SDRAM.

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for more details about the IP core and register description.

1.1. Purpose

LPDDR4 and its SDK are a set of application programming interfaces (APIs) that provide access to specific Lattice hardware and software capabilities. This document is intended to act as a reference guide for developers by providing details of the C language driver APIs and function call flows.

1.2. Audience

The intended audience for this document includes embedded system designers and embedded software developers using Lattice CertusPro™-NX devices. The technical guide assumes readers have expertise in embedded systems and FPGA technologies.

1.3. Driver and IP Compatibility

Table 1.1. Driver and IP Version

Driver version	IP version
24.01.00	2.3.0

Table 1.2. Quick Facts of Driver Tested Environment

Driver tested on HW Device	Nexus Family	CertusPro-NX
Tool Version	Lattice Propel Builder	2023.2, 2024.1.2406150513_p
	Lattice Propel SDK	2023.2, 2024.1.2406150513_p
	Lattice Radiant Software	2023.2, 2024.1.0.34.2
	Radiant Programmer	2023.2, 2024.1.0.34.2

2. API Description

2.1. LPDDR_init()

This API is used to initialize the LPDDR4 controller. This API takes the base address of LPDDR4 and initializes handle.

```
LPDDR_RET lpddr4_init(lpddr4 *instancePtr, unsigned int base_addr)
```

Table 2.1. LPDDR_init()

In/Out	Parameter	Description	Returns
In	instancePtr	Handle of the LPDDR4 structure.	"Enum LPDDR_RET state" NO_FAIL = 0, CBT_FAIL, WR_LVL_FAIL, RD_TRN_FAIL, WR_TRN_FAIL, OTHER_FAIL
In	base_addr	Base address to be assigned to controller.	

2.2. LPDDR4_GetFeatureControlReg()

This API is used to read the configuration settings from the Feature Control Register. This register reflects the modes of operation specified according to 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 run-time. The CPU reads this register to identify the modes of operation. Refer to the to the IP user guide for a register description.

```
unsigned int lpddr4_GetFeatureControlReg(lpddr4 *instancePtr, unsigned int *reg_data)
```

Table 2.2. LPDDR4_GetFeatureControlReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handle of the LPDDR4 structure.	0: failure 1: success
Out	reg_data	Current values read from Feature Control Register.	

2.3. LPDDR4_GetSettingReg()

The Reset Register controls the reset of the internal CPU and Training Engine, where both are reset at power-on. The host should de-assert the reset to the internal CPU and Training Engine to begin memory initialization and training. Upon training completion, the internal CPU sets the trn_eng_rst_n signal low (RESET_REG[0]=0), to place the Training Engine in reset to save power. When the Enable APB I/F attribute is unchecked, the init_start_i signal controls the internal CPU reset (RESET_REG[1]). This register does not reset the Configuration Set Registers (CSRs). Refer IP user guide for register description.

```
unsigned int lpddr4_GetSettingReg(lpddr4 *instancePtr, unsigned int *reg_data);
```

Table 2.3. LPDDR4_GetSettingReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handle of the LPDDR4 structure.	0: failure 1: success
Out	reg_data	Current values read from Feature Control Register.	

2.4. LPDDR4_TemperatureChangeInterruptEnable()

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

```
unsigned int lpddr4_TemperatureChangeInterruptEnable(lpddr4 *instancePtr);
```

Table 2.4. LPDDR4_TemperatureChangeInterruptEnable()

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

2.5. LPDDR4_TemperatureChangeInterruptDisable()

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

```
unsigned int lpddr4_TemperatureChangeInterruptDisable(lpddr4 *instancePtr);
```

Table 2.5. LPDDR4_TemperatureChangeInterruptDisable()

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

2.6. LPDDR4_TrainingErrorInterruptEnable()

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

```
unsigned int lpddr4_TrainingErrorInterruptEnable(lpddr4 *instancePtr);
```

Table 2.6. LPDDR4_TrainingErrorInterruptEnable()

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

2.7. LPDDR4_TrainingErrorInterruptDisable()

This API disables training error interrupt. 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 lpddr4_TrainingErrorInterruptDisable(lpddr4 *instancePtr);
```

Table 2.7. LPDDR4_TrainingErrorInterruptDisable()

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

2.8. LPDDR4_TrainingDoneInterruptEnable()

This API enables training done interrupt. 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 lpddr4_TrainingDoneInterruptEnable(lpddr4 *instancePtr);
```

Table 2.8. LPDDR4_TrainingDoneInterruptEnable()

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

2.9. LPDDR4_TrainingDoneInterruptDisable()

This API disables training done interrupt. 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 lpddr4_TrainingDoneInterruptDisable(lpddr4 *instancePtr);
```

Table 2.9. LPDDR4_TrainingDoneInterruptDisable()

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

2.10. LPDDR4_GetTrainingOperationReg()

This API is used to read the configuration settings from the Training Operation Register. This register controls the memory initialization and training. It is recommended to set these register bits to 1’b0 during simulation to shorten the initialization and training procedure. When Enable APB I/F is unchecked, TRN_OP_REG[7:0] is set to the value of the trn_opr_i input signal. Refer to the IP user guide for the register description.

```
unsigned int lpddr4_GetTrainingOperationReg(lpddr4 *instancePtr, unsigned int
*reg_data);
```

Table 2.10. LPDDR4_GetTrainingOperationReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handle of the LPDDR4 structure.	0: failure 1: success
Out	reg_data	Current values read from Training Operation Register.	

2.11. LPDDR4_GetStatusReg()

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

```
unsigned int lpddr4_GetStatusReg(lpddr4 *instancePtr, unsigned int *reg_data);
```

Table 2.11. LPDDR4_GetStatusReg()

In/Out	Parameter	Description	Returns
In	instancePtr	Handle of the LPDDR4 structure.	0: failure 1: success
Out	reg_data	Current values read from Status Register.	

3. Function Call Flow Diagrams

3.1. LPDDR4_init()

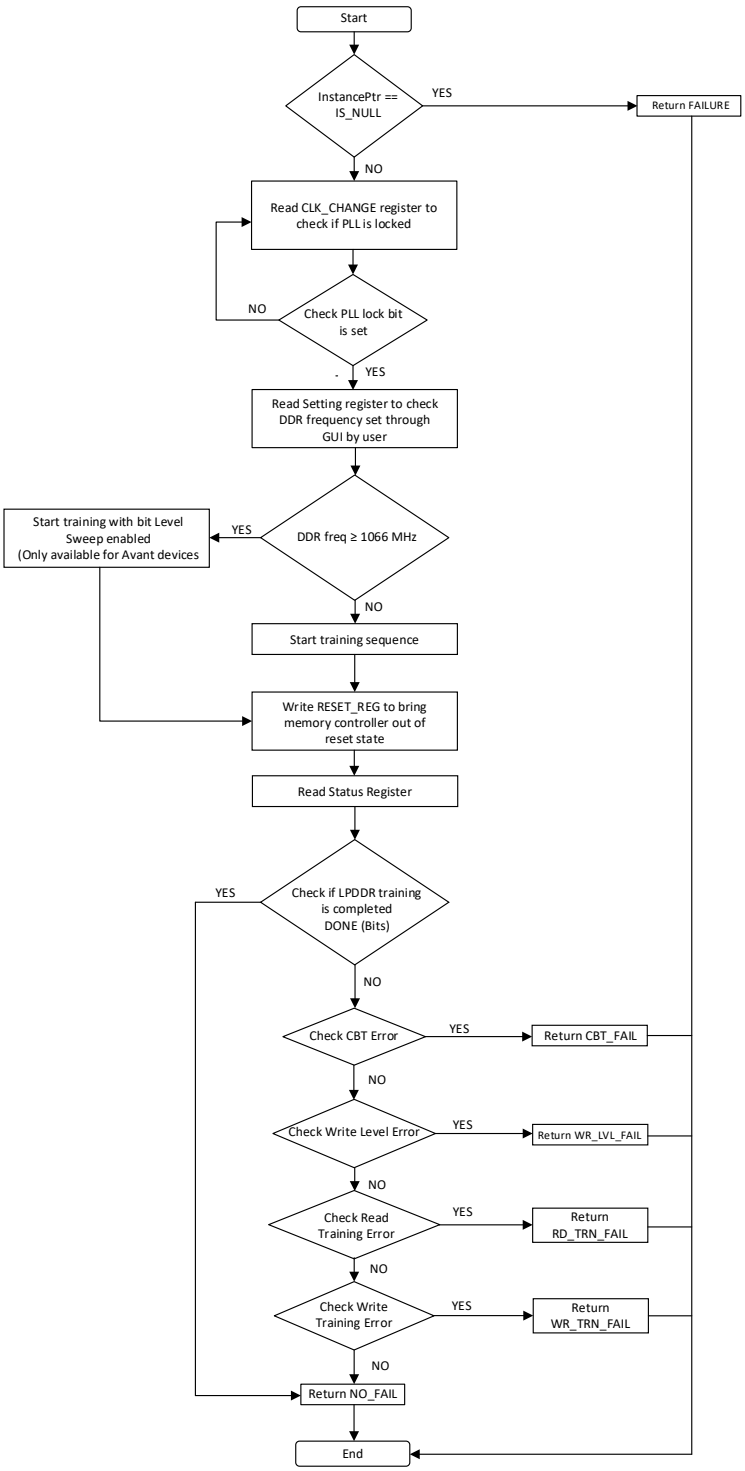

Figure 3.1. LPDDR4_init()

3.2. LPDDR4_GetFeatureControlReg()

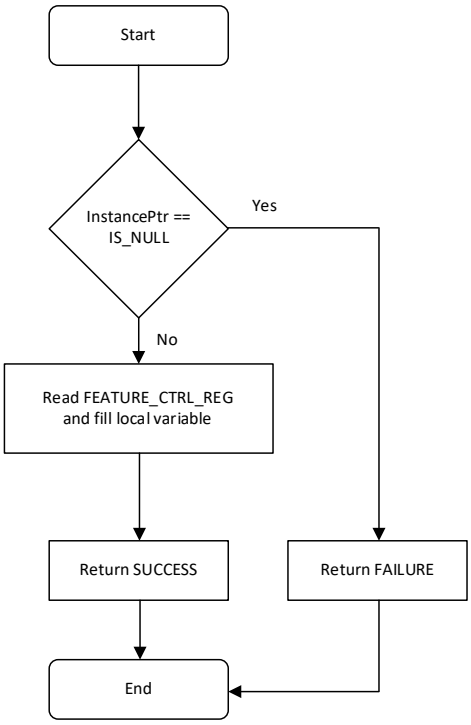

Figure 3.2. LPDDR4_GetFeatureControlReg()

3.3. LPDDR_GetSettingReg()

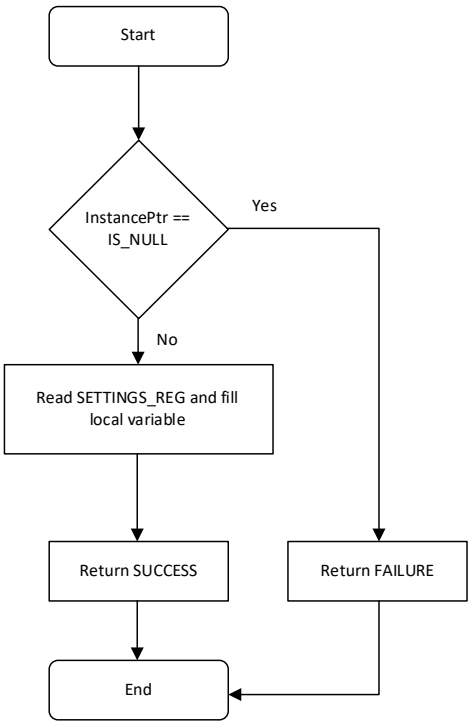

Figure 3.3. LPDDR4_GetSettingReg()

3.4. LPDDR4_TemperatureChangeInterruptEnable()

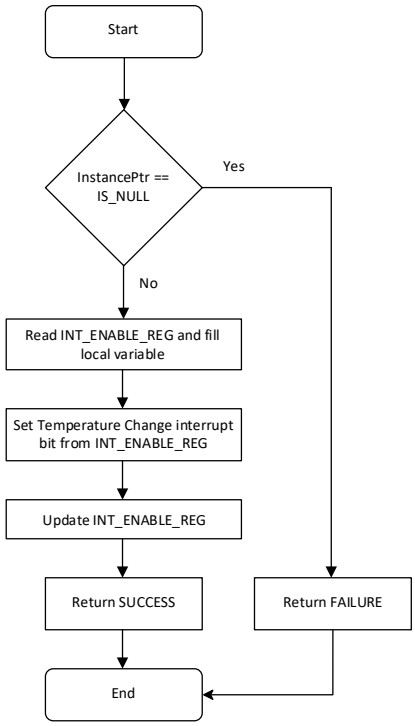

Figure 3.4. LPDDR4_TemperatureChangeInterruptEnable()

3.5. LPDDR4_TemperatureChangeInterruptDisable()

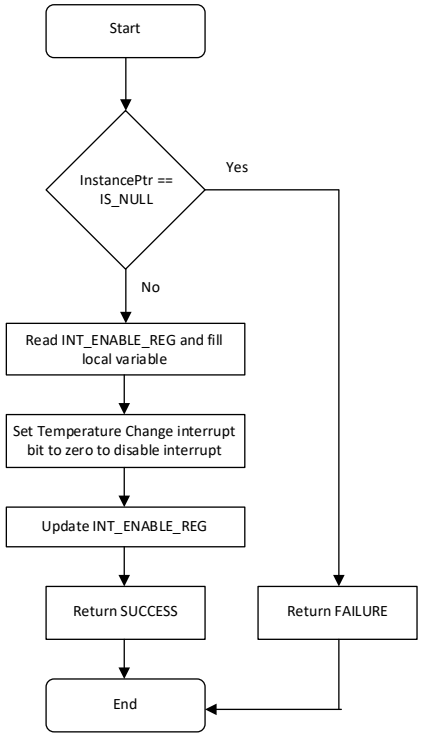

Figure 3.5. LPDDR4_TemperatureChangeInterruptDisable()

3.6. LPDDR4_TrainingErrorInterruptEnable()

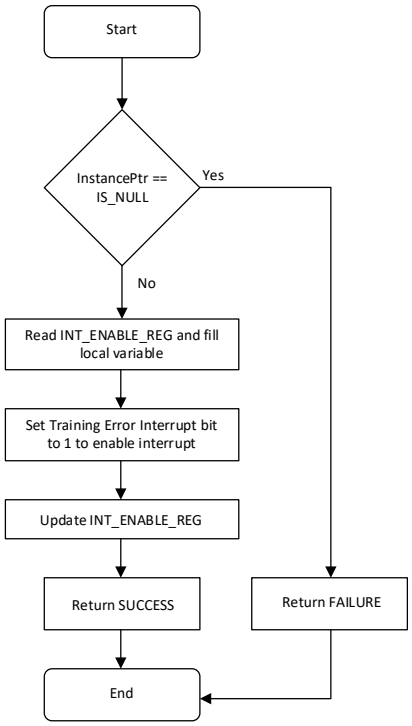

Figure 3.6. LPDDR4_TrainingErrorInterruptEnable()

3.7. LPDDR4_TrainingErrorInterruptDisable()

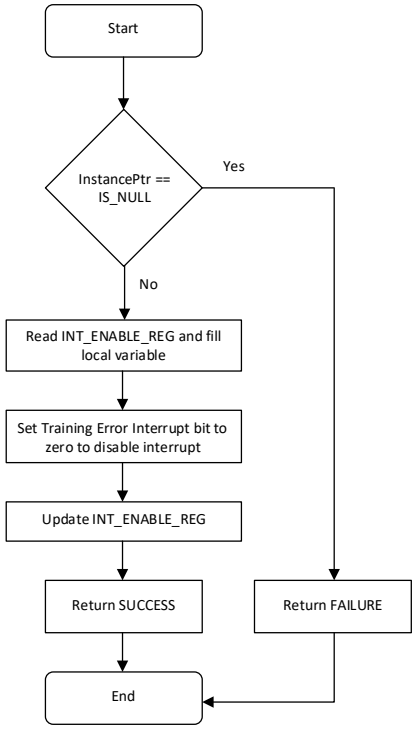

Figure 3.7. LPDDR4_TrainingErrorInterruptDisable()

3.8. LPDDR4_TrainingDoneInterruptEnable()

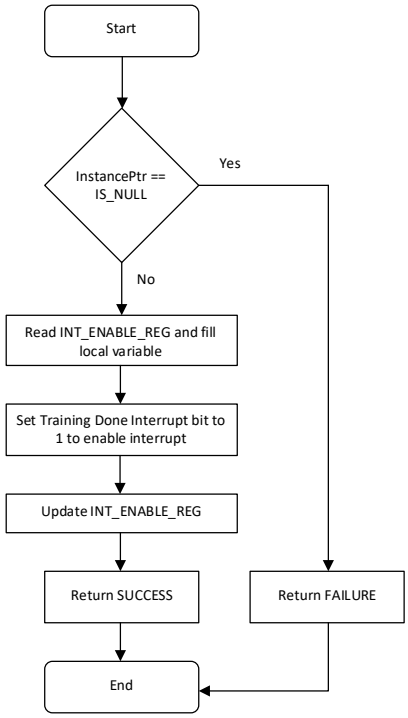

Figure 3.8. LPDDR4_TrainingDoneInterruptEnable()

3.9. LPDDR4_TrainingDoneInterruptDisable()

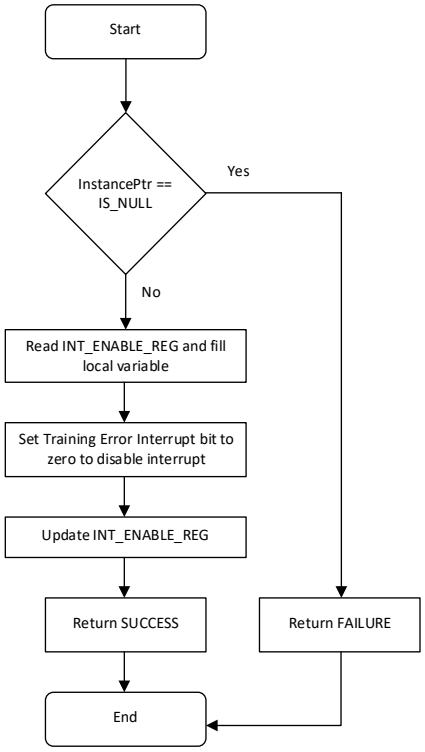

Figure 3.9. LPDDR4_TrainingDoneInterruptDisable()

3.10. LPDDR4_GetTrainingOperationReg()

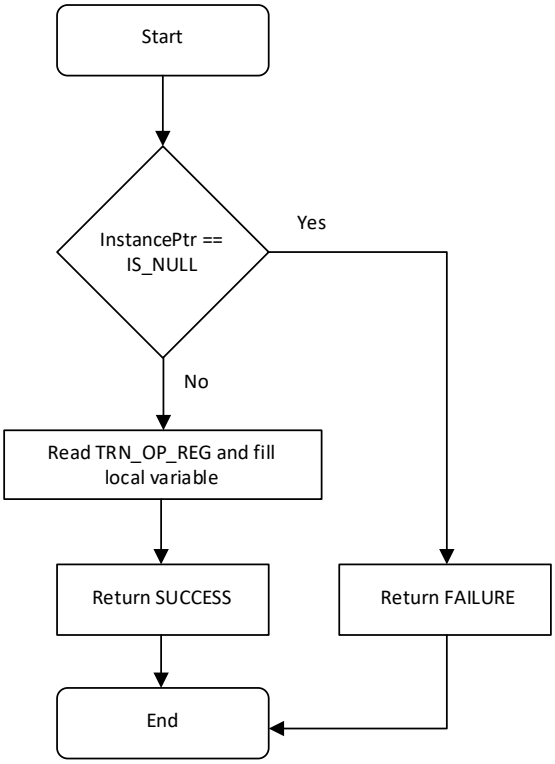

Figure 3.10. LPDDR4_GetTrainingOperationReg()

3.11. LPDDR4_GetStatusReg()

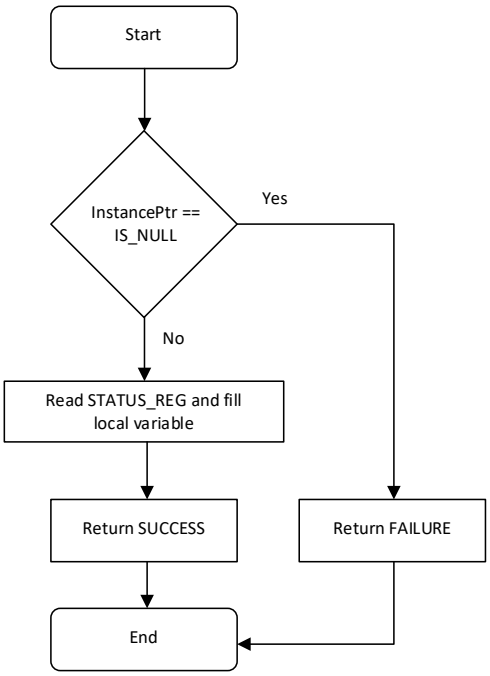

Figure 3.11. LPDDR4_GetStatusReg()

4. API Data Structures

4.1. Struct LPDDR4

Table 4.1. LPDDR4 Parameters

Data Type	Struct Member	Description
unsigned int	lpddr4_base_address	Base address of the LPDDR4 memory controller.

4.2. Union and Struct Feature_ctrl_reg_t

Refer to the IP user guide for the detailed register description.

Table 4.2. Feature_ctrl_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the feature_ctrl_reg_t register members.
unsigned int	reg	Variable to access the feature_ctrl_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	fea_rsvd_1	reserved
unsigned int	dbi_en	The dbi_en enables the Data Bus Inversion function to reduce the toggling of DDR data signals, thus improving the signal integrity and reducing dynamic power consumption as specified in the Enable DBI attribute.
unsigned int	power_down_en	The pwr_down_en enables the power saving mode by putting the memory in self-refresh when there is no traffic for sclk_o cycles as specified in the No. of SCLK to enter Self-Refresh from no traffic attribute.
unsigned int	gear_ratio	The gear_ratio is fixed at value 1, which corresponds to 8:1 gearing and specifies the ratio of the DDR clock domain and system clock domain as ddr_ck_o frequency = 4 × sclk_o frequency.
unsigned int	addr_translation	The address translation specifies the mapping of the address bits of the local memory-mapped address and the memory address in terms of row, column, bank, and rank. Only 1 address translation scheme (addr_translation=0) is currently supported; Refer IP user guide for more details.
unsigned int	ddr_type	The ddr_type is fixed at value 1, which corresponds to LPDDR4 as the standard implemented by the Memory Controller IP Core.
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 • 4 – reserved • 5 – reserved • 6 – reserved • 7 – DDR Bus Width = 64 bits
unsigned int	num_ranks	The num_ranks is fixed at value 0, which corresponds to single rank.
unsigned int	fea_rsvd_2	reserved

4.3. Union and Struct clk_chng_t

Refer to the IP user guide for the detailed register description.

Table 4.3. Clk_chng_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the clk_chng_t register members.
unsigned int	reg	Variable to access the clk_chng_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	ddr_clk_sel	Indicates target DDR clock frequency for clock frequency change. Used by the Training Engine. Do not write 1 to this register.
unsigned int	dll_update_en	Re-calibrates DLL after clock frequency change when asserted. Used by the Training Engine. Do not write 1 to this register.
unsigned int	clk_update	Initiates clock frequency change when asserted. Used by the Training Engine. Do not write 1 to this register.
unsigned int	pll_lock	Asserts when PLL is locked.
unsigned int	clk_chng_rsvd	Reserved

4.4. Union and Struct reset_reg_t

Refer to the IP user guide for the detailed register description.

Table 4.4. reset_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the reset_reg_t register members.
unsigned int	reg	Variable to access the reset_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	trn_eng_rst_n	De-assert to reset training engine to begin memory initialization and training. Refer IP user guide for more details.
unsigned int	cpu_reset_n	De-assert to reset internal CPU to begin memory initialization and training. Refer IP user guide for more details.
unsigned int	res_rsvd	Reserved

4.5. Union and Struct settings_reg_t

Refer to the IP user guide for the detailed register description.

Table 4.5. Settings_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the settings_reg_t register members.
unsigned int	reg	Variable to access 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 write command to the first write data.
unsigned int	set_rsvd_1	Reserved
unsigned int	read_latency	The read_latency specifies the number of DDR clock cycles from read command to the first read data.
unsigned int	set_rsvd_2	Reserved

Data Type	Union Member	Description
unsigned int	clk_freq	The clk_freq specifies the DDR command clock frequency in MHz.
unsigned int	set_rsvd_3	Reserved

4.6. Union and Struct int_status_reg_t

Refer to the IP user guide for the detailed register description.

Table 4.6. Int_status_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the int_status_reg_t register members.
unsigned int	reg	Variable to access 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 is 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. The user should read STATUS_REG to determine the specific error. • 0 – No interrupt • 1 – Interrupt pending
unsigned int	int_sts_rsvd_1	Reserved
unsigned int	temp_change_int	Temperature Change Interrupt. The Memory Controller periodically reads MR4 of the LPDDR4 memory device according to the Temperature Check Period attribute. This interrupt bit asserts when MR4 indicates a temperature change, and the refresh rate is not equal to 1x refresh. • 0 – No interrupt • 1 – Interrupt pending
unsigned int	int_sts_rsvd_2	Reserved

4.7. Union and Struct status_reg_t

Refer to the IP user guide for the detailed register description.

Table 4.7. Status_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the status_reg_t register members.
unsigned int	reg	Variable to access the status_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	phy_ready	Asserts when PHY initialization of the PHY is complete and ready for operation.
unsigned int	cbt_done	Asserts when Command Bus Training has completed.
unsigned int	write_lvl_done	Asserts when write leveling has completed.
unsigned int	read_trn_done	Asserts when read training has completed.
unsigned int	write_trn_done	Asserts when write training has completed.
unsigned int	in_self_refresh	Asserts when the memory is in self-refresh.
unsigned int	sts_rsvd1	Reserved
unsigned int	cbt_err	Asserts when a failure occurs during Command Bus Training.
unsigned int	write_lvl_err	Asserts when a failure occurs during write leveling.

Data Type	Union Member	Description
unsigned int	read_trn_err	Asserts when a failure occurs during read training. If the read_trn_done signal is low (STATUS_REG[3]=0), the training failed to find a setting that properly captures the read DQS burst. If the read_trn_done signal is high (STATUS_REG[3]=1), the training failed to find an optimal read DQS-DQ delay for capturing the read data.
unsigned int	write_trn_err	Asserts when a failure occurs during write training.
unsigned int	err_on_rank	When error_on_rank is set to 2'bx1, it indicates a training error has occurred on rank 0.
unsigned int	sts_rsvd2	Reserved
unsigned int	refresh_rate	Reflects the value of the refresh rate set in MR4 within LPDDR4 memory. It specifies the required refresh period based on the temperature of the LPDDR4 device. • 3'b000: SDRAM Low temperature operating limit exceeded • 3'b001: 4x refresh • 3'b010: 2x refresh • 3'b011: 1x refresh (default) • 3'b100: 0.5x refresh • 3'b101: 0.25x refresh, no de-rating • 3'b110: 0.25x refresh, with de-rating • 3'b111: SDRAM High temperature operating limit exceeded
unsigned int	rank0_done	Asserts when training has completed for rank 0.
unsigned int	rank1_done	Asserts when training has completed for rank 1.
unsigned int	sts_rsvd3	Reserved

4.8. Union and Struct int_enable_reg_t

Refer to the IP user guide for the detailed Register description.

Table 4.8. int_enable_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the int_enable_reg_t register members.
unsigned int	reg	Variable to access 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_1	Reserved
unsigned int	temp_change_int	Temperature Change Interrupt Enable. • 0 – Interrupt disabled • 1 – Interrupt enabled
unsigned int	int_en_rsvd_2	Reserved

4.9. Union and Struct trn_op_reg_t

Refer to the IP user guide for the detailed register description.

Table 4.9. trn_op_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Variable to access the trn_op_reg_tregister members.
unsigned int	reg	Variable to access the trn_op_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	init_en	Provides ability to shorten initialization for simulation purposes. • 0 – Initialization is greatly reduced. For example, reset time and CKE low time is greatly shortened. • 1 – Initialization is performed according to the LPDDR4 JEDEC Standard.
unsigned int	cbt_en	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. • 0 – The Training Engine will shorten the command bus training. Instead of iterating through the different CA delays to find the optimal delay value, it will only program the Trained CA Delay attribute to PHY. • 1 – The Training Engine performs command bus training to find the optimal CA delay value.
unsigned int	write_lvl_en	Enables write leveling during initialization and training. Write leveling compensates for CK-DQS timing skews. The Training Engine performs write leveling according to the JEDEC standard. • 0 – The Training Engine programs the DQS<0,1,2,3,4,5,6,7> Trained Write Leveling Delay attributes to the PHY and checks that the DQ feedback after DQS pulse is high. • 1 – The Training Engine performs write leveling.
unsigned int	read_trn_en	Enables read training during initialization and training. Read training tunes the PHY to capture the incoming read DQS burst according to the read latency setting. If read training fails for a certain latency setting, the Training Engine will increase the latency setting by 1 and retry read training. Once the read DQS burst is captured properly, the read DQS-DQ timing is trained to improve the read data valid window. • 0 – The Training Engine programs the DQS<0,1,2,3,4,5,6,7> Trained RDCLKSEL, Trained DQSBUF Read Delay/Sign, and Trained Read Latency attributes to the PHY and checks that read access will be successful. • 1 – The Training Engine performs read training.
unsigned int	write_trn_en	Enables write training during initialization and training. Write training optimizes the write DQ delay with respect to the write DQS to improve the data valid window for write operations. If write training fails for a certain latency, the Training Engine will increase the latency setting by 1 and retry write training. • 0 – The Training Engine programs the Trained Write DQ/DBI delay and Trained Write Latency to PHY attributes, and checks that write and read FIFO access is equal. • 1 – The Training Engine performs write training.
unsigned int	ca_vref_training_en	Enables VREF training on command and address bus as part of command bus training. VREF training is embedded in the

Data Type	Union Member	Description
		CBT route where the GUI values are read in as starting values. . • 0 – VREF training is not included in the CBT routine • 1 – The Training Engine performs command and address VREF training as part of command bus training.
unsigned int	mc_vref_training_en	Enables VREF training on the memory controller as part of read training. VREF training is embedded in the training routing where the VREF values in the GUI are read in as starting values. . • 0 – disable VREF training during read training • 1 – The Training Engine performs memory controller VREF training as part of read training.
unsigned int	mem_vref_training_en	Enables VREF training on the SDRAM memory as part of write training. VREF training is embedded in the training routine where the VREF values in the GUI are read in as starting values. . • 0 – VREF training is not included in write training • 1 – The Training Engine performs memory controller VREF training as part of write training.
unsigned int	trn_op_rsvd1	Reserved

5. API Enum

5.1. Enum LPDDR_RET

Table 5.1. Qspi_reg_type_t Variables

Enum Member	Enum Decimal Value
NO_FAIL	0
CBT_FAIL	1
WR_LVL_FAIL	2
RD_TRN_FAIL	3
WR_TRN_FAIL	4
OTHER_FAIL	5

6. API Variables

This is used to access the base address of LPDDR4 IP from Propel Builder Configuration table/sys_platform.h.

```
static lpddr4 *lpddr4_inst;
```

7. API Macros

Table 7.1. API Macros Description

Macro	Description
#define LPDDR4_DRV_VER	LPDDR4 Driver version.
#define FEATURE_CTRL_REG	Register address of Feature Control Register.
#define RESET_REG	Register address of Reset Register.
#define SETTINGS_REG	Register address of Settings Register.
#define RSVD_1	Reserved
#define INT_STATUS_REG	Register address of Interrupt Status Register.
#define INT_ENABLE_REG	Register address of Interrupt Enable Register.
#define INT_SET_REG	Register address of Interrupt Set Register. (for debug purpose only)
#define CLK_CHANGE	Register address of Clock Change Register.
#define TRN_OP_REG	Register address of Training Operation Register.
#define STATUS_REG	Register address of Status Register.
#define SUCCESS	Success.
#define FAILURE	Failure.
#define IS_NULL	Null value.
#define OUT_OF_RESET	Out of Reset Value for bringing training and CPU out of reset.
#define TRN_BIT_LVL_SWEEP_EN	Bit Level Sweep training enable (only available for Avant devices)
#define TRN_BIT_LVL_SWEEP_DIS	Bit Level Sweep training disable
#define DDR_FREQ	DDR frequency for Bit Level Sweep training
#define LPDDR_DONE_BITS	LPDDR initialization done bits.
#define LPDDR_ERR_DONE_BITS	LPDDR initialization done bits.
#define PLL_LOCK_BIT	LPDDR PLL lock bit
#define TRN_EN	LPDDR training enable bits

References

- LPDDR4 Memory Controller IP Core for Nexus Devices User Guide (FPGA-IPUG-02127)
- Lattice Propel 2024.1 Builder User Guide (FPGA-UG-02212)
- Lattice Solutions IP Cores web page
- Lattice Propel Design Environment web page
- Lattice Radiant Software 2024.1 User Guide
- 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 1.0, August 2024

Section	Change Summary
All	Initial release.

www.latticesemi.com