CertusPro-NX LPDDR4 Memory Controller Driver API Reference

User Guide

FPGA-TN-02378-1.2

June 2026

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

Contents 3
Abbreviations in This Document 7
1. Introduction 8
1.1. Purpose 8
1.2. Audience 8
1.3. Driver and IP Compatibility 8
2. API Description 9
2.1. lpddr4_init() 9
2.2. lpddr4_wait_pll_lock() 9
2.3. lpddr4_bring_out_of_reset() 9
2.4. lpddr4_start_training() 9
2.5. lpddr4_do_training() 10
2.6. lpddr4_GetFeatureControlReg() 10
2.7. lpddr4_GetSettingReg() 10
2.8. lpddr4_TemperatureChangeInterruptEnable() 10
2.9. lpddr4_TemperatureChangeInterruptDisable() 11
2.10. lpddr4_TrainingErrorInterruptEnable() 11
2.11. lpddr4_TrainingErrorInterruptDisable() 11
2.12. lpddr4_TrainingDoneInterruptEnable() 11
2.13. lpddr4_TrainingDoneInterruptDisable() 12
2.14. lpddr4_GetClkChangeReg() 12
2.15. lpddr4_GetTrainingOperationReg() 12
2.16. lpddr4_GetStatusReg() 12
2.17. lpddr4_GetTrainingPatternReg() 13
2.18. lpddr4_GetTrainedDqsbufWrlvl0Reg() 13
2.19. lpddr4_GetTrainedDqsbufWrlvl1Reg() 13
2.20. lpddr4_GetTrainedVal1Reg(lpddr4 *instance_ptr) 13
2.21. lpddr4_GetTrainedVal2Reg(lpddr4 *instance_ptr) 14
2.22. lpddr4_GetTrainedVal3Reg(lpddr4 *instance_ptr) 14
2.23. lpddr4_SetTrainingOperationReg() 14
2.24. lpddr4_SetResetReg() 14
2.25. lpddr4_SetInterruptEnableReg() 14
3. Function Call Flow Diagrams 15
3.1. lpddr4_init() 15
3.2. lpddr4_wait_pll_lock() 15
3.3. lpddr4_bring_out_of_reset() 16
3.4. lpddr4_start_training() 16
3.5. lpddr4_do_training() 17
3.6. lpddr4_GetFeatureControlReg() 18
3.7. lpddr4_GetSettingReg() 18
3.8. lpddr4_TemperatureChangeInterruptEnable() 19
3.9. lpddr4_TemperatureChangeInterruptDisable() 19
3.10. lpddr4_TrainingErrorInterruptEnable() 20
3.11. lpddr4_TrainingErrorInterruptDisable() 20
3.12. lpddr4_TrainingDoneInterruptEnable() 21
3.13. lpddr4_TrainingDoneInterruptDisable() 21
3.14. lpddr4_GetClkChangeReg() 22
3.15. lpddr4_GetTrainingOperationReg() 22
3.16. lpddr4_GetStatusReg() 23
3.17. lpddr4_GetTrainingPatternReg() 23
3.18. lpddr4_GetTrainedDqsbufWrlvl0Reg() 24
3.19. lpddr4_GetTrainedDqsbufWrlvl1Reg() 24

3.20. lpddr4_GetTrainedVal1Reg() 25
3.21. lpddr4_GetTrainedVal2Reg() 25
3.22. lpddr4_GetTrainedVal3Reg() 26
3.23. lpddr4_SetTrainingOperationReg() 26
3.24. lpddr4_SetResetReg() 27
3.25. lpddr4_SetInterruptEnableReg() 27
4. API Data Structures 28
4.1. Struct lpddr4 28
4.2. Union and Struct feature_ctrl_reg_t 28
4.3. Union and Struct reset_reg_t 29
4.4. Union and Struct settings_reg_t 29
4.5. Union and Struct int_status_reg_t 29
4.6. Union and Struct int_enable_reg_t 30
4.7. Union and Struct clk_chng_t 30
4.8. Union and Struct trn_op_reg_t 31
4.9. Union and Struct status_reg_t 32
5. API Enum 34
5.1. Enum LPDDR_RET 34
6. API Macros 35
References 37
Technical Support Assistance 38
Revision History 39

Figures

Figure 3.1. lpddr4_init() Function Flow Call....15
Figure 3.2. lpddr4_wait_pll_lock() Function Flow Call....15
Figure 3.3. lpddr4_bring_out_of_reset() Function Flow Call....16
Figure 3.4. lpddr4_start_training() Function Flow Call....16
Figure 3.5. lpddr4_do_training() Function Flow Call....17
Figure 3.6. lpddr4_GetFeatureControlReg() Function Flow Call....18
Figure 3.7. lpddr4_GetSettingReg() Function Flow Call....18
Figure 3.8. lpddr4_TemperatureChangeInterruptEnable() Function Flow Call....19
Figure 3.9. lpddr4_TemperatureChangeInterruptDisable() Function Flow Call....19
Figure 3.10. lpddr4_TrainingErrorInterruptEnable() Function Flow Call....20
Figure 3.11. lpddr4_TrainingErrorInterruptDisable() Function Flow Call....20
Figure 3.12. lpddr4_TrainingDoneInterruptEnable() Function Flow Call....21
Figure 3.13. lpddr4_TrainingDoneInterruptDisable() Function Flow Call....21
Figure 3.14. lpddr4_GetClkChangeReg() Function Flow Call....22
Figure 3.15. lpddr4_GetTrainingOperationReg() Function Flow Call....22
Figure 3.16. lpddr4_GetStatusReg() Function Flow Call....23
Figure 3.17. lpddr4_GetTrainingPatternReg() Function Flow Call....23
Figure 3.18. lpddr4_GetTrainedDqsbufWrlvl0Reg() Function Flow Call....24
Figure 3.19. lpddr4_GetTrainedDqsbufWrlvl1Reg() Function Flow Call....24
Figure 3.20. lpddr4_GetTrainedVal1Reg() Function Flow Call....25
Figure 3.21. lpddr4_GetTrainedVal2Reg() Function Flow Call....25
Figure 3.22. lpddr4_GetTrainedVal3Reg() Function Flow Call....26
Figure 3.23. lpddr4_SetTrainingOperationReg() Function Flow Call....26
Figure 3.24. lpddr4_SetResetReg() Function Flow Call....27
Figure 3.25. lpddr4_SetInterruptEnableReg() Function Flow Call....27

Tables

Table 1.1. Driver and IP Version 8
Table 1.2. Quick Facts of Driver Tested Environment 8
Table 2.1. lpddr4_init() 9
Table 2.2. lpddr4_wait_pll_lock() 9
Table 2.3. lpddr4_bring_out_of_reset() 9
Table 2.4. lpddr4_start_training() 9
Table 2.5. lpddr4_do_training() 10
Table 2.6. lpddr4_GetFeatureControlReg() 10
Table 2.7. lpddr4_GetSettingReg() 10
Table 2.8. lpddr4_TemperatureChangeInterruptEnable() 10
Table 2.9. lpddr4_TemperatureChangeInterruptDisable() 11
Table 2.10. lpddr4_TrainingErrorInterruptEnable() 11
Table 2.11. lpddr4_TrainingErrorInterruptDisable() 11
Table 2.12. lpddr4_TrainingDoneInterruptEnable() 11
Table 2.13. lpddr4_TrainingDoneInterruptDisable() 12
Table 2.14. lpddr4_GetClkChangeReg() 12
Table 2.15. lpddr4_GetTrainingOperationReg() 12
Table 2.16. lpddr4_GetStatusReg() 12
Table 2.17. lpddr4_GetTrainingPatternReg() 13
Table 2.18. lpddr4_GetTrainedDqsbufWrlvl0Reg() 13
Table 2.19. lpddr4_GetTrainedDqsbufWrlvl1Reg() 13
Table 2.20. lpddr4_GetTrainedVal1Reg() 13
Table 2.21. lpddr4_GetTrainedVal2Reg() 14
Table 2.22. lpddr4_GetTrainedVal3Reg() 14
Table 2.23. lpddr4_SetTrainingOperationReg() 14
Table 2.24. lpddr4_SetResetReg() 14
Table 2.25. lpddr4_SetInterruptEnableReg() 14
Table 4.1. LPDDR4 Parameters 28
Table 4.2. feature_ctrl_reg_t Parameters 28
Table 4.3. reset_reg_t Parameters 29
Table 4.4. Settings_reg_t Parameters 29
Table 4.5. Int_status_reg_t Parameters 29
Table 4.6. int_enable_reg_t Parameters 30
Table 4.7. Clk_chng_t Parameters 30
Table 4.8. trn_op_reg_t Parameters 31
Table 4.9. Status_reg_t Parameters 32
Table 5.1. LPDDR_RETVariables 34
Table 6.1. API Macros Description 35

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	Double Data Rate
DDR PHY	Double Data Rate Physical Layer
DQ	Data
DQS	Data Strobe
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
IP	Intellectual Property (FPGA)
JEDEC	Joint Electron Device Engineering Council
LPDDR4	Low Power Double Data Rate Generation 4
PHY	Physical Layer
PLL	Phase-Locked Loop
SCLK	System Clock
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random-Access Memory

1. Introduction

The Lattice Semiconductor LPDDR4 Memory Controller for Lattice Nexus™ devices provides a complete solution comprising a controller, a DDR PHY, and the clocking and training logic required to interface with LPDDR4 SDRAM (Synchronous Dynamic Random-Access Memory).

For more information about the IP core and its register description, refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127).

1.1. Purpose

The LPDDR4 driver and its SDK provide a set of application programming interfaces (APIs) that provide access to specific Lattice hardware and software capabilities. Use this document as a reference guide; it describes the C-language driver APIs and function call flows.

1.2. Audience

This document is intended for embedded system designers and embedded software developers who use Lattice CertusPro™-NX devices. The guide assumes you 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
v26.01.00	2.7.0

For driver and IP version information, refer to the LPDDR4 Memory Controller IP Release Notes (FPGA-RN-02057).

Table 1.2. Quick Facts of Driver Tested Environment

Driver tested on HW Device	Nexus Family	CertusPro-NX
Tool Version	Lattice Propel™ Builder	2025.2
	Lattice Propel™ SDK	2025.2
	Lattice Radiant™ Software	2025.2
	Lattice Radiant™ Programmer	2025.2

2. API Description

2.1. lpddr4_init()

Use this API to initialize the LPDDR4 controller. It takes the LPDDR4 base address and initializes the handle.

```
unsigned int lpddr4_init(lpddr4 *instance_ptr, unsigned int base_addr);
```

Table 2.1. lpddr4_init()

In/Out	Parameter	Description	Returns
In,out	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success
In	base_addr	Base address to be assigned to controller.	

2.2. lpddr4_wait_pll_lock()

This API waits until the PLL (Phase-Locked Loop) locks.

```
unsigned int lpddr4_wait_pll_lock(lpddr4 *instance_ptr);
```

Table 2.2. lpddr4_wait_pll_lock()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.3. lpddr4_bring_out_of_reset()

This API brings LPDDR4 out of reset to start the training sequence.

```
unsigned int lpddr4_bring_out_of_reset(lpddr4 *instance_ptr);
```

Table 2.3. lpddr4_bring_out_of_reset()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.4. lpddr4_start_training()

This API prepares the LPDDR4 training sequence by programming the TRN_OP_REG with the required bit settings to select and enable the desired training steps.

```
unsigned int lpddr4_start_training(lpddr4 *instance_ptr);
```

Table 2.4. lpddr4_start_training()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.5. lpddr4_do_training()

This API performs the complete LPDDR4 training sequence. It combines lpddr4_wait_pll_lock(), lpddr4_start_training(), lpddr4_bring_out_of_reset(), and polls the training status into a single call. It waits for PLL lock, starts the training sequence, and polls the training status until training completes.

```
LPDDR_RET lpddr4_do_training(lpddr4 *instance_ptr);
```

Table 2.5. lpddr4_do_training()

In/Out	Parameter	Description	Returns

In	instance_ptr	Handles the lpddr4 structure.	LPDDR_RET
			{
			NO_FAIL = 0,
			CBT_FAIL,
			WR_LVL_FAIL,
			RD_TRN_FAIL,
			WR_TRN_FAIL,
			OTHER_FAIL
			}

2.6. lpddr4_GetFeatureControlReg()

This API reads the configuration settings from the Feature Control Register. Refer to the IP user guide for the register description.

```
feature_ctrl_reg_t lpddr4_GetFeatureControlReg(lpddr4 *instance_ptr);
```

Table 2.6. lpddr4_GetFeatureControlReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Feature Control Register value

2.7. lpddr4_GetSettingReg()

This API reads the Settings register (SETTINGS_REG). Refer to the IP user guide for the register description.

```
settings_reg_t lpddr4_GetSettingReg(lpddr4 *instance_ptr);
```

Table 2.7. lpddr4_GetSettingReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Settings Register value

2.8. lpddr4_TemperatureChangeInterruptEnable()

This API enables temperature-change 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 lpddr4_TemperatureChangeInterruptEnable(lpddr4 *instance_ptr);
```

Table 2.8. lpddr4_TemperatureChangeInterruptEnable()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.9. 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 IP user guide for the register description.

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

Table 2.9. lpddr4_TemperatureChangeInterruptDisable()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.10. lpddr4_TrainingErrorInterruptEnable()

This API enables the 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 lpddr4_TrainingErrorInterruptEnable(lpddr4 *instance_ptr);
```

Table 2.10. lpddr4_TrainingErrorInterruptEnable()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.11. lpddr4_TrainingErrorInterruptDisable()

This API disables the 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 *instance_ptr);
```

Table 2.11. lpddr4_TrainingErrorInterruptDisable()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.12. lpddr4_TrainingDoneInterruptEnable()

This API enables the 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 *instance_ptr);
```

Table 2.12. lpddr4_TrainingDoneInterruptEnable()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.13. lpddr4_TrainingDoneInterruptDisable()

This API disables the 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 *instance_ptr);
```

Table 2.13. lpddr4_TrainingDoneInterruptDisable()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success

2.14. lpddr4_GetClkChangeReg()

This API reads the Clock Change register (CLK_CHANGE_REG). Refer to the IP user guide for the register description.

```
clk_chng_t lpddr4_GetClkChangeReg(lpddr4 *instance_ptr);
```

Table 2.14. lpddr4_GetClkChangeReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Clock Change Register value

2.15. lpddr4_GetTrainingOperationReg()

This API reads the configuration settings from the Training Operation Register. Refer to the IP user guide for the register description.

```
trn_op_reg_t lpddr4_GetTrainingOperationReg(lpddr4 *instance_ptr);
```

Table 2.15. lpddr4_GetTrainingOperationReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Training Operation Register value

2.16. lpddr4_GetStatusReg()

This API reads the Status Register. Refer to the IP user guide for the register description.

```
status_reg_t lpddr4_GetStatusReg(lpddr4 *instance_ptr);
```

Table 2.16. lpddr4_GetStatusReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Status Register value

2.17. lpddr4_GetTrainingPatternReg()

This API reads the Training Pattern Register (TRN_PATRN_REG). Refer to the IP user guide for the register description.

```
trn_patrn_reg_t lpddr4_GetTrainingPatternReg(lpddr4 *instance_ptr);
```

Table 2.17. lpddr4_GetTrainingPatternReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Training Pattern Register value

2.18. lpddr4_GetTrainedDqsbufWrlvl0Reg()

This API reads the Trained DQS (Data Strobe) Buffer Write Leveling Register 0 (TRAINED_DQSBUF_WRLVL_0). Refer to the IP user guide for the register description.

```
trained_dqsbuf_wrlvl_0_t lpddr4_GetTrainedDqsbufWrlvl0Reg(lpddr4 *instance_ptr);
```

Table 2.18. lpddr4_GetTrainedDqsbufWrlvl0Reg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Trained DQS Buffer Write Leveling Register 0 value

2.19. lpddr4_GetTrainedDqsbufWrlvl1Reg()

This API reads the Trained DQS Buffer Write Leveling Register 1 (TRAINED_DQSBUF_WRLVL_1). Refer to the IP user guide for the register description.

```
trained_dqsbuf_wrlvl_1_t lpddr4_GetTrainedDqsbufWrlvl1Reg(lpddr4 *instance_ptr);
```

Table 2.19. lpddr4_GetTrainedDqsbufWrlvl1Reg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Trained DQS Buffer Write Leveling Register 1 value

2.20. lpddr4_GetTrainedVal1Reg(lpddr4 *instance_ptr)

This API reads the Trained Values Register 1 (TRAINED_VAL_1). Refer to the IP user guide for the register description.

```
trained_val_1_t lpddr4_GetTrainedVal1Reg(lpddr4 *instance_ptr);
```

Table 2.20. lpddr4_GetTrainedVal1Reg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Trained Values Register 1

2.21. lpddr4_GetTrainedVal2Reg(lpddr4 *instance_ptr)

This API reads the Trained Values Register 2 (TRAINED_VAL_2). Refer to the IP user guide for the register description.

```
trained_val_2_t lpddr4_GetTrainedVal2Reg(lpddr4 *instance_ptr);
```

Table 2.21. lpddr4_GetTrainedVal2Reg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Trained Values Register 2

2.22. lpddr4_GetTrainedVal3Reg(lpddr4 *instance_ptr)

This API reads the Trained Values Register 3 (TRAINED_VAL_3). Refer to the IP user guide for the register description.

```
trained_val_3_t lpddr4_GetTrainedVal3Reg(lpddr4 *instance_ptr);
```

Table 2.22. lpddr4_GetTrainedVal3Reg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	Trained Values Register 3

2.23. lpddr4_SetTrainingOperationReg()

This API writes the Training Operation Register (TRN_OP_REG). Refer to the IP user guide for the register description.

```
unsigned int lpddr4_SetTrainingOperationReg(lpddr4 *instance_ptr, trn_op_reg_t reg_val);
```

Table 2.23. lpddr4_SetTrainingOperationReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success
In	reg_val	New register value to be written.	

2.24. lpddr4_SetResetReg()

This API writes the Reset Register (RESET_REG). Refer to the IP user guide for the register description.

```
unsigned int lpddr4_SetResetReg(lpddr4 *instance_ptr, reset_reg_t reg_val);
```

Table 2.24. lpddr4_SetResetReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success
In	reg_val	New register value to be written.	

2.25. lpddr4_SetInterruptEnableReg()

This API writes the Interrupt Enable Register (INT_ENABLE_REG). Refer to the IP user guide for the register description.

```
unsigned int lpddr4_SetInterruptEnableReg(lpddr4 *instance_ptr, int_enable_reg_t
reg_val);
```

Table 2.25. lpddr4_SetInterruptEnableReg()

In/Out	Parameter	Description	Returns
In	instance_ptr	Handles the lpddr4 structure.	0: failure 1: success
In	reg_val	New register value to be written.	

3. Function Call Flow Diagrams

3.1. lpddr4_init()

This section shows the function call flow for the *lpddr4_init()* API.

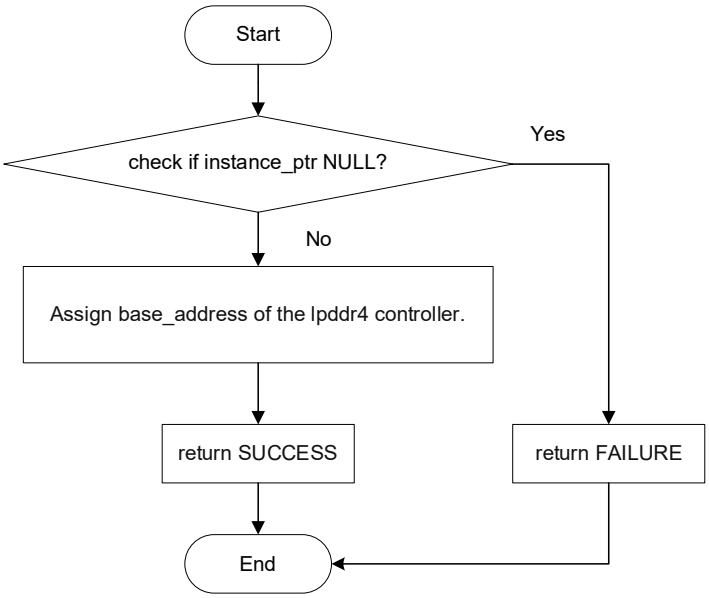

Figure 3.1. lpddr4_init() Function Flow Call

3.2. lpddr4_wait_pll_lock()

This section shows the function call flow for the *lpddr4_wait_pll_lock()* API.

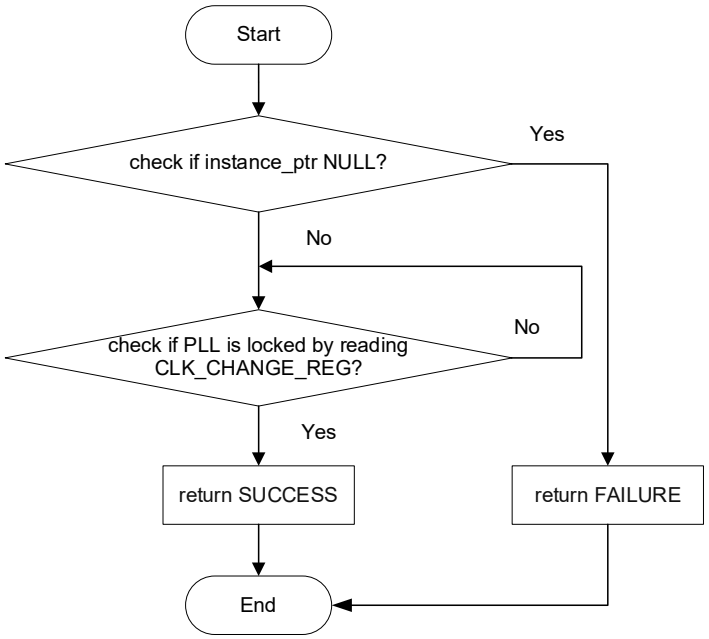

Figure 3.2. lpddr4_wait_pll_lock() Function Flow Call

3.3. lpddr4_bring_out_of_reset()

This section shows the function call flow for the *lpddr4_bring_out_of_reset()* API.

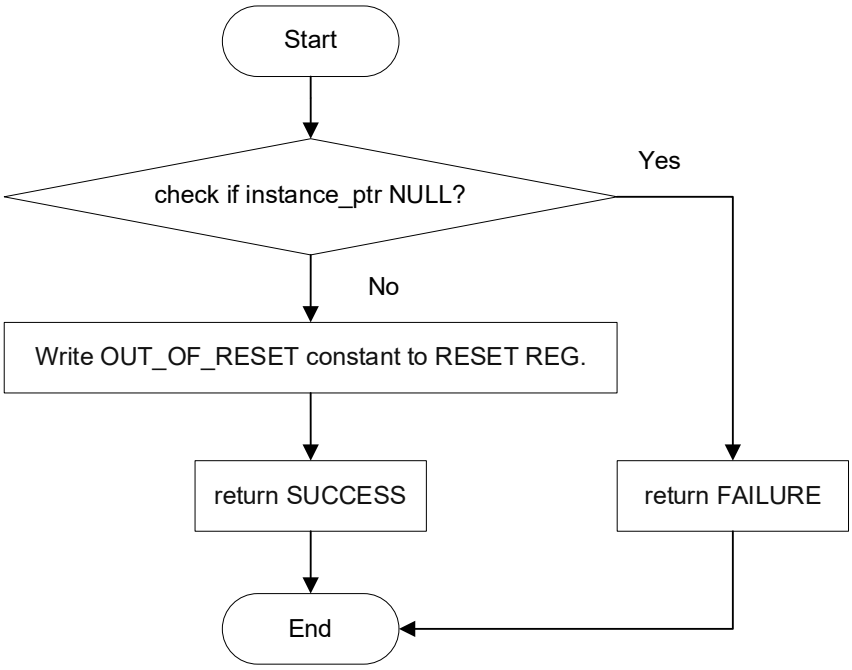

Figure 3.3. lpddr4_bring_out_of_reset() Function Flow Call

3.4. lpddr4_start_training()

This section shows the function call flow for the *lpddr4_start_training()* API.

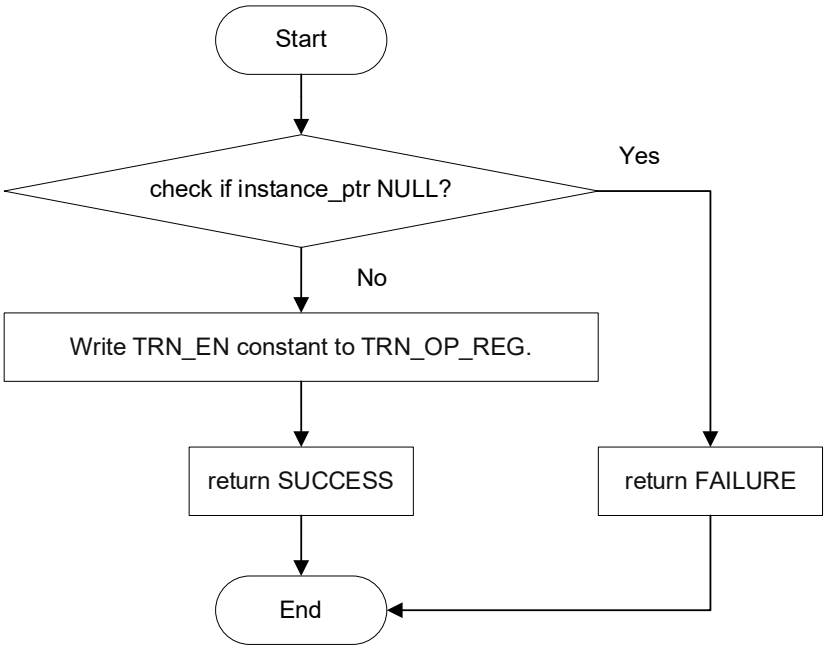

Figure 3.4. lpddr4_start_training() Function Flow Call

3.5. lpddr4_do_training()

This section shows the function call flow for the *lpddr4_do_training()* API.

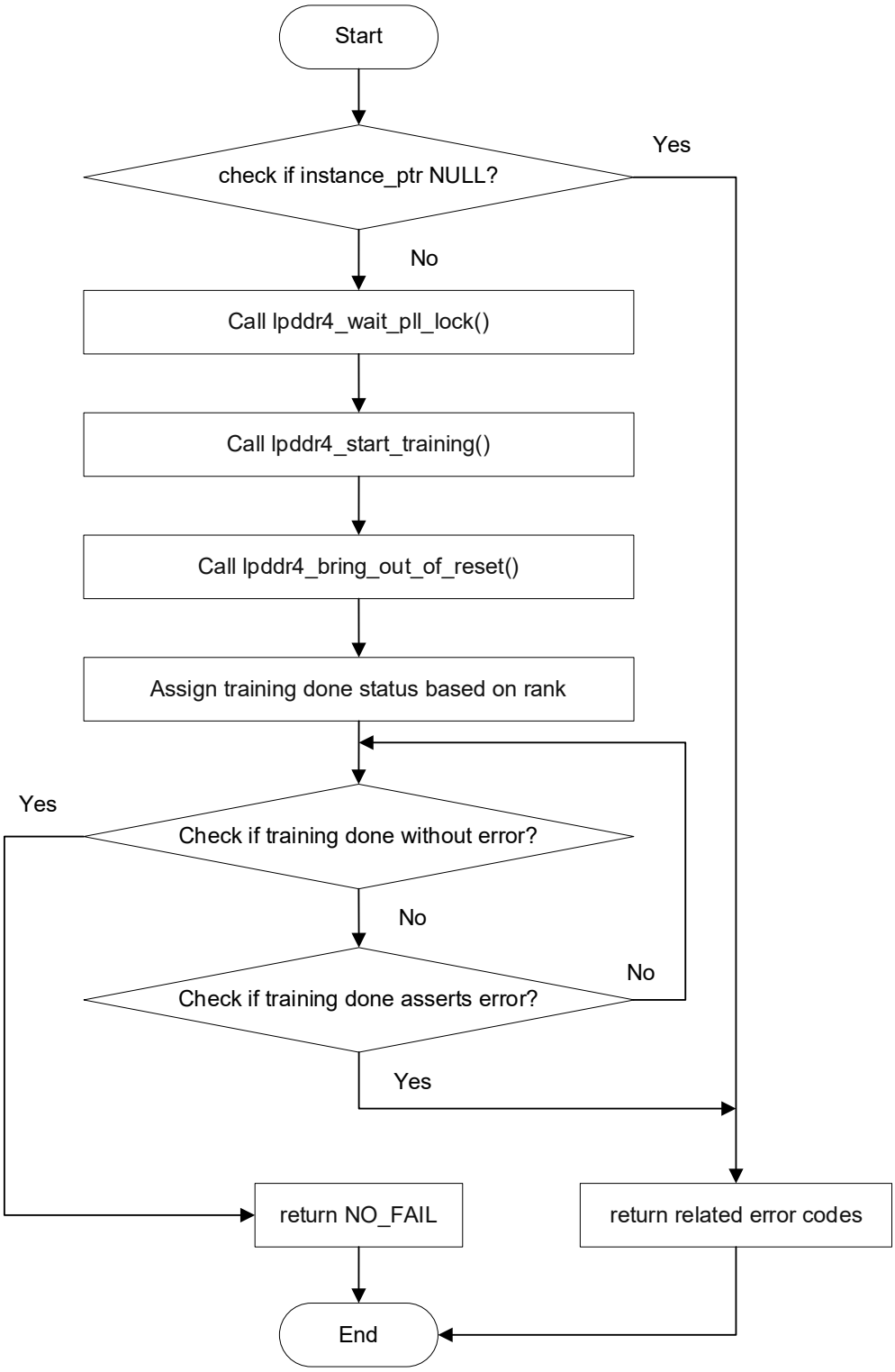

Figure 3.5. lpddr4_do_training() Function Flow Call

3.6. lpddr4_GetFeatureControlReg()

This section shows the function call flow for the *lpddr4_GetFeaturedControlReg()* API.

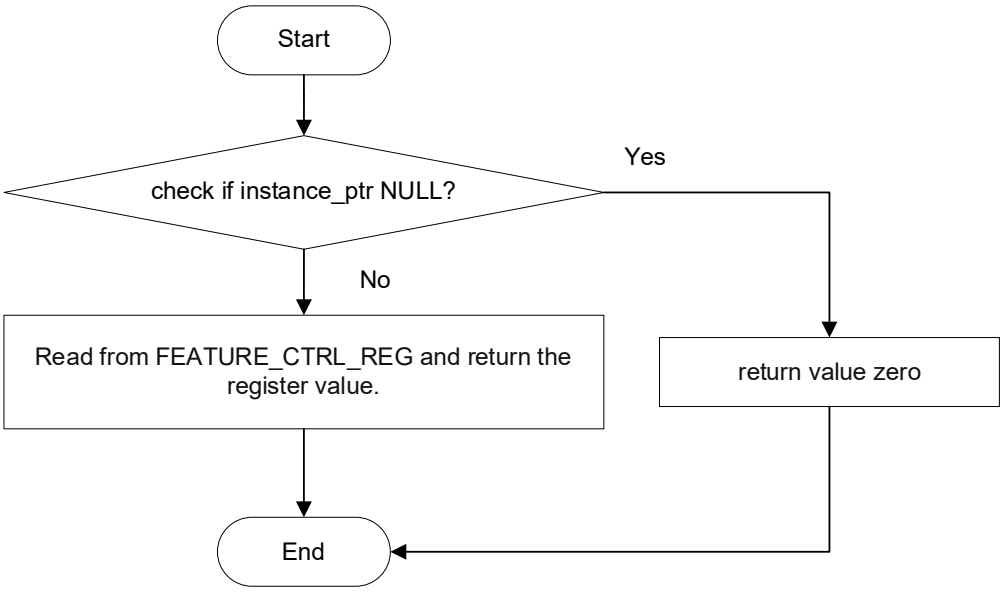

Figure 3.6. lpddr4_GetFeatureControlReg() Function Flow Call

3.7. lpddr4_GetSettingReg()

This section shows the function call flow for the *lpddr4_GetSettingReg()* API.

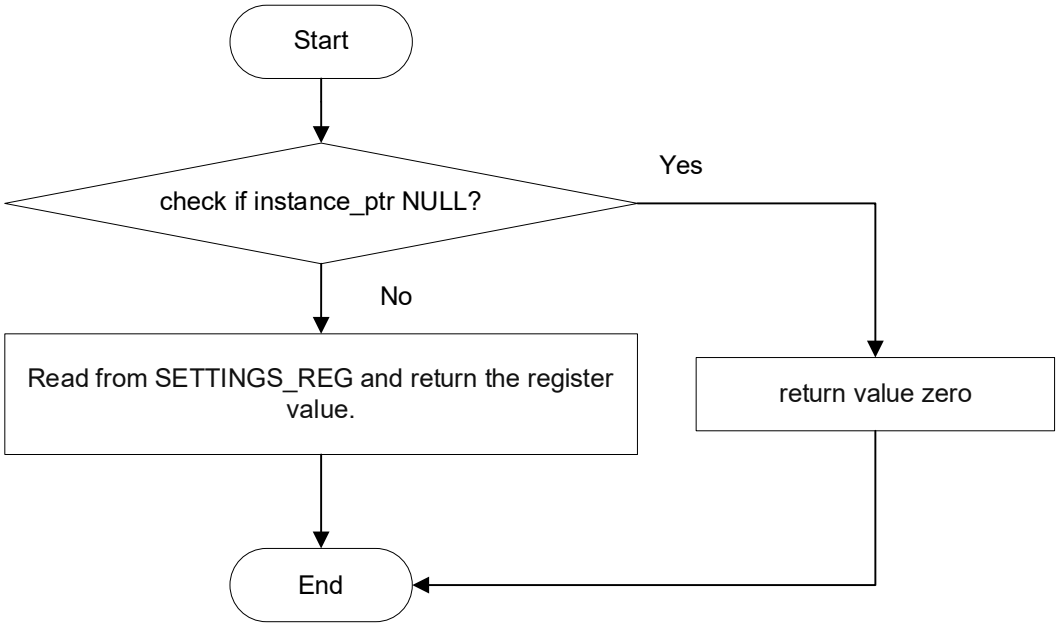

Figure 3.7. lpddr4_GetSettingReg() Function Flow Call

3.8. lpddr4_TemperatureChangeInterruptEnable()

This section shows the function call flow for the *lpddr4_TemperatureChangeInterruptEnable()* API.

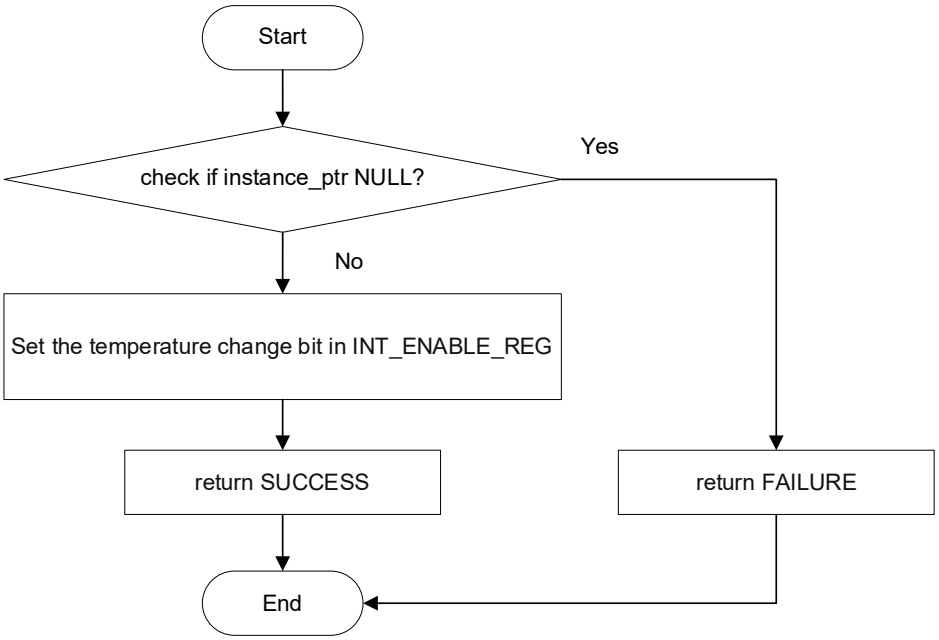

Figure 3.8. lpddr4_TemperatureChangeInterruptEnable() Function Flow Call

3.9. lpddr4_TemperatureChangeInterruptDisable()

This section shows the function call flow for the *lpddr4_TemperatureChangeInterruptDisable()* API.

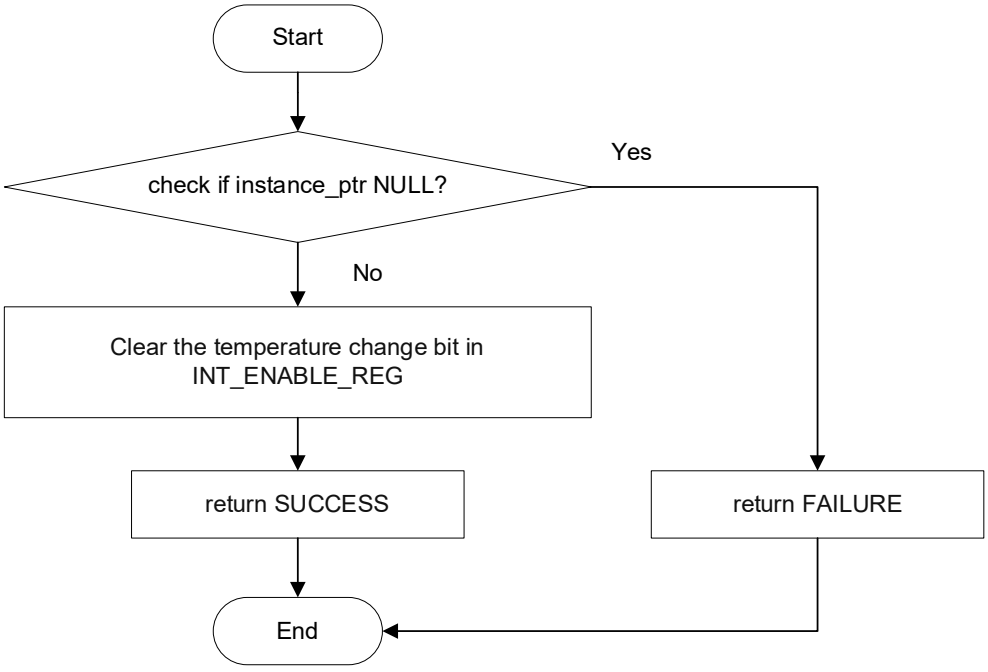

Figure 3.9. lpddr4_TemperatureChangeInterruptDisable() Function Flow Call

3.10. lpddr4_TrainingErrorInterruptEnable()

This section shows the function call flow for the *lpddr4_TrainingErrorInterruptEnable()* API.

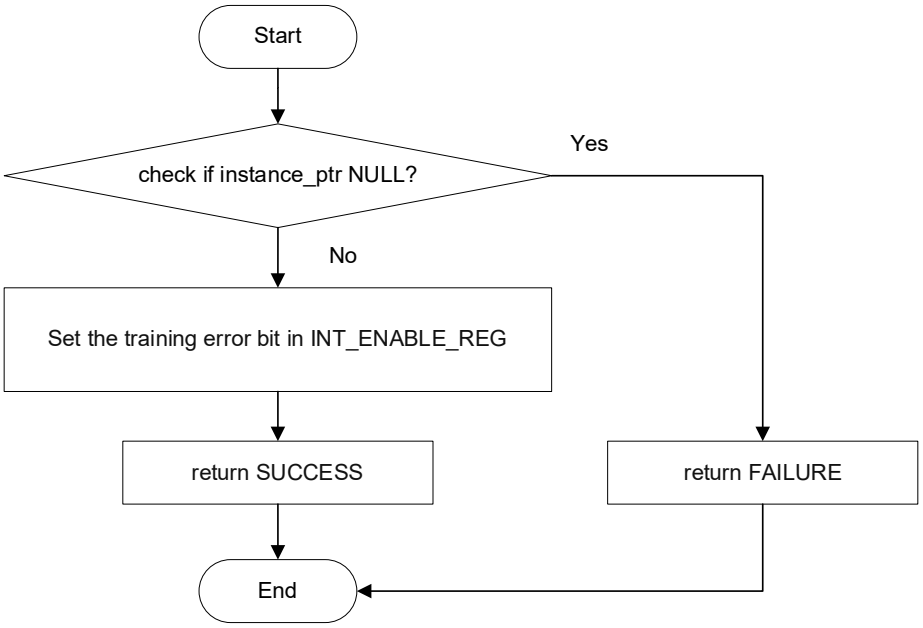

Figure 3.10. lpddr4_TrainingErrorInterruptEnable() Function Flow Call

3.11. lpddr4_TrainingErrorInterruptDisable()

This section shows the function call flow for the *lpddr4_TrainingErrorInterruptDisable()* API.

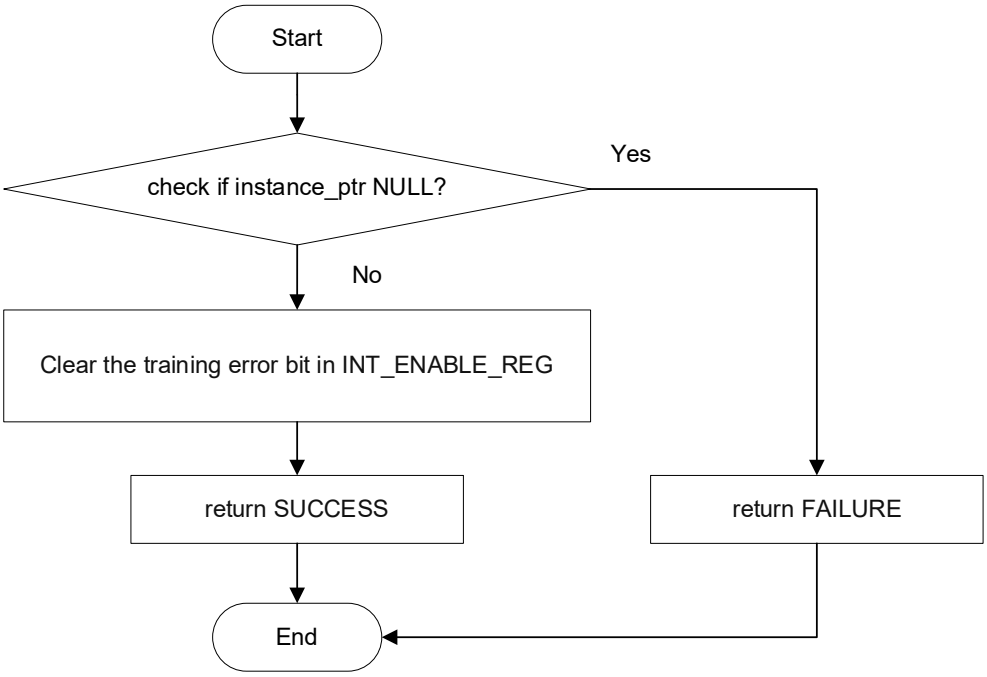

Figure 3.11. lpddr4_TrainingErrorInterruptDisable() Function Flow Call

3.12. lpddr4_TrainingDoneInterruptEnable()

This section shows the function call flow for the *lpddr4_TrainingDoneInterruptEnable()* API.

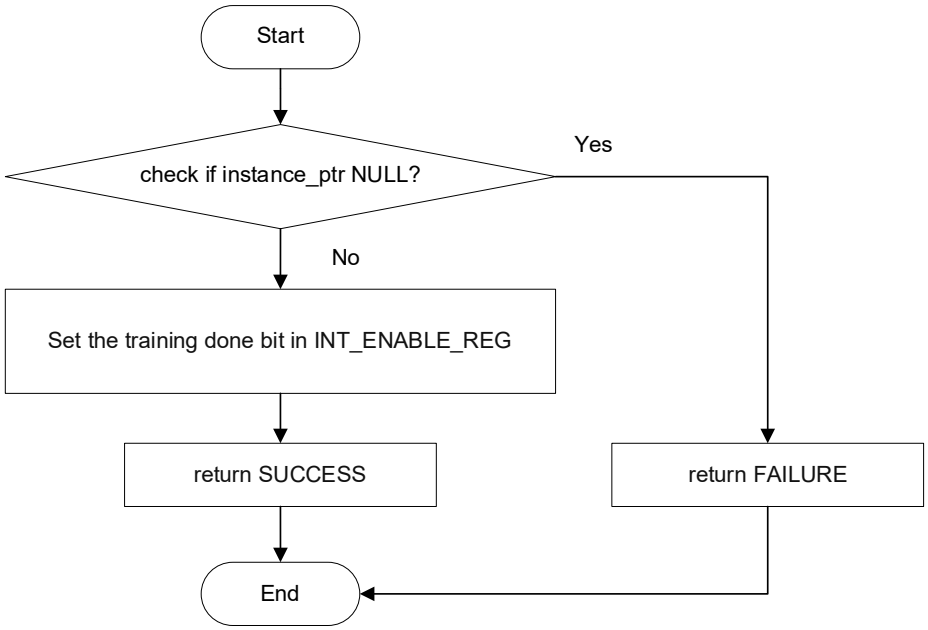

Figure 3.12. lpddr4_TrainingDoneInterruptEnable() Function Flow Call

3.13. lpddr4_TrainingDoneInterruptDisable()

This section shows the function call flow for the *lpddr4_TrainingDoneInterruptDisable()* API.

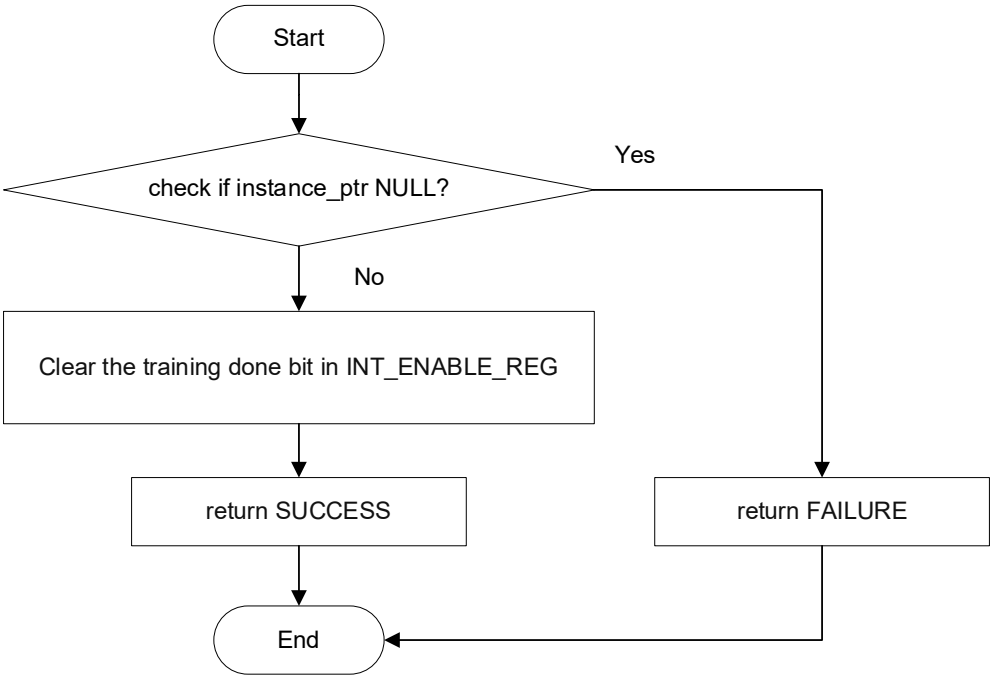

Figure 3.13. lpddr4_TrainingDoneInterruptDisable() Function Flow Call

3.14. lpddr4_GetClkChangeReg()

This section shows the function call flow for the *lpddr4_GetClkChangeReg()* API.

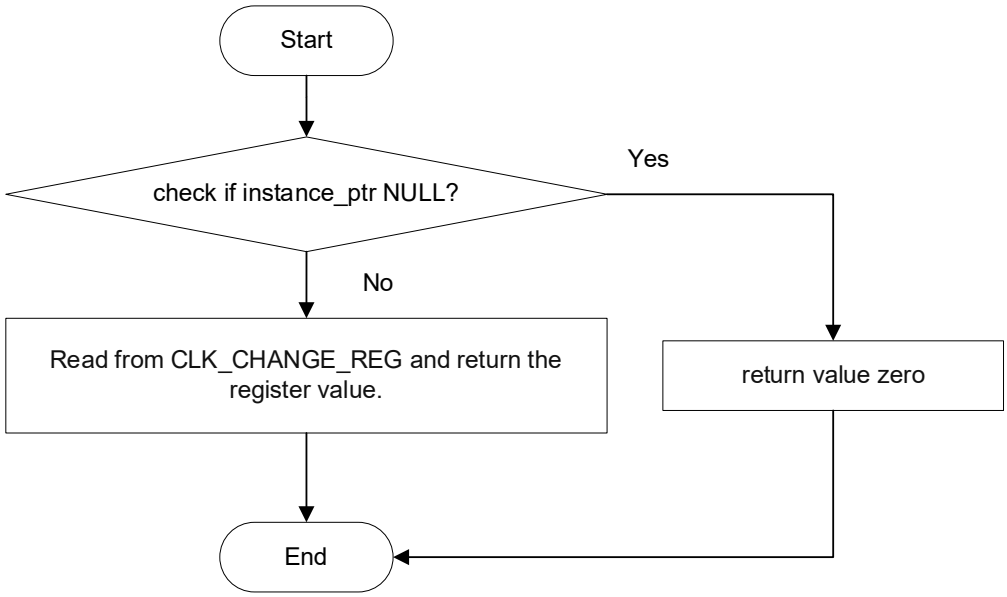

Figure 3.14. lpddr4_GetClkChangeReg() Function Flow Call

3.15. lpddr4_GetTrainingOperationReg()

This section shows the function call flow for the *lpddr4_GetTrainingOperationReg()* API.

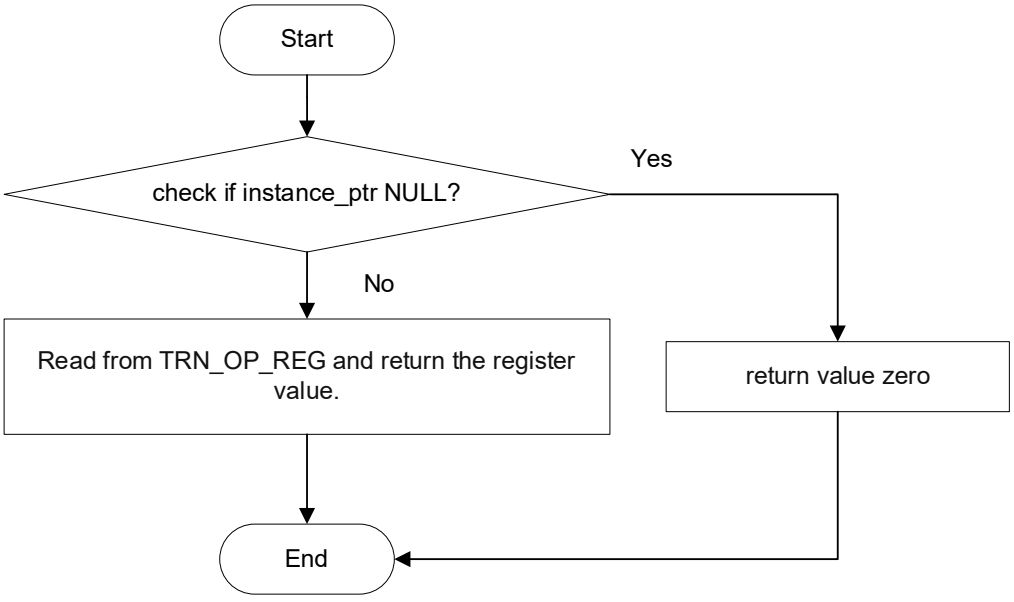

Figure 3.15. lpddr4_GetTrainingOperationReg() Function Flow Call

3.16. lpddr4_GetStatusReg()

This section shows the function call flow for the *lpddr4_GetStatusReg()* API.

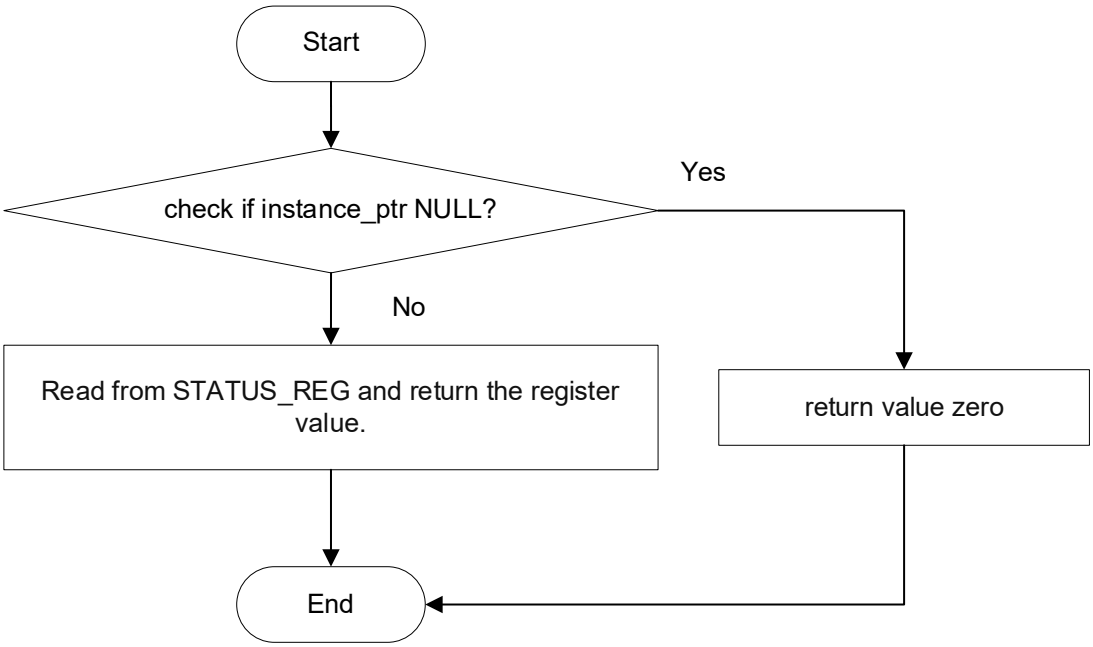

Figure 3.16. lpddr4_GetStatusReg() Function Flow Call

3.17. lpddr4_GetTrainingPatternReg()

This section shows the function call flow for the *lpddr4_GetTrainingPatternReg()* API.

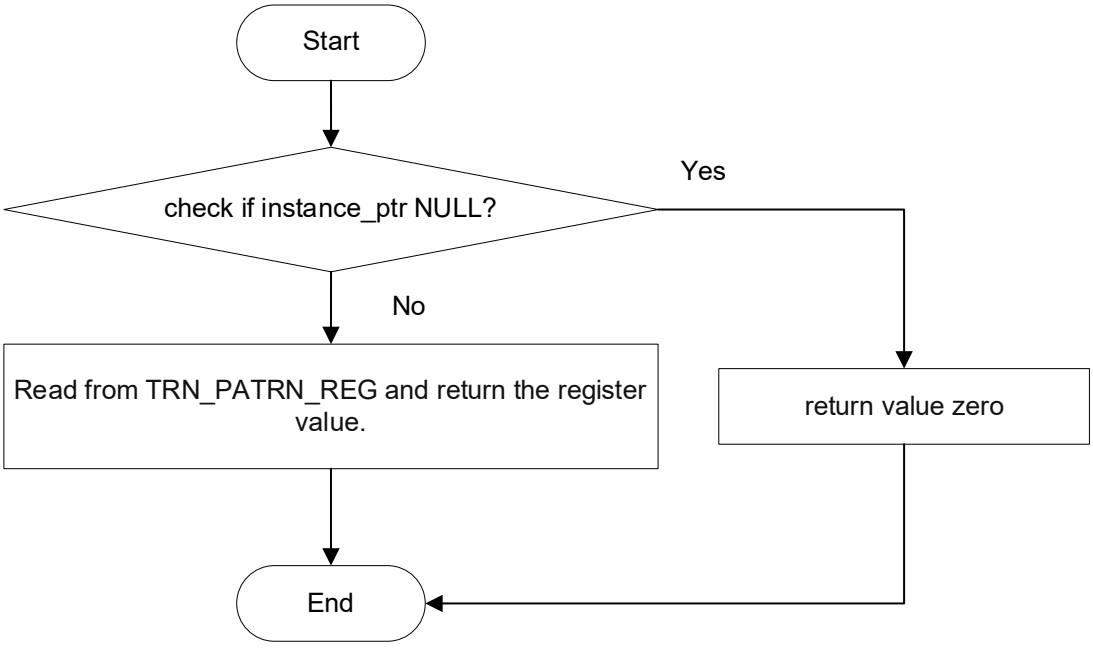

Figure 3.17. lpddr4_GetTrainingPatternReg() Function Flow Call

3.18. lpddr4_GetTrainedDqsbufWrlvl0Reg()

This section shows the function call flow for the *lpddr4_GetTrainedDqsbufWrlvl0Reg()* API.

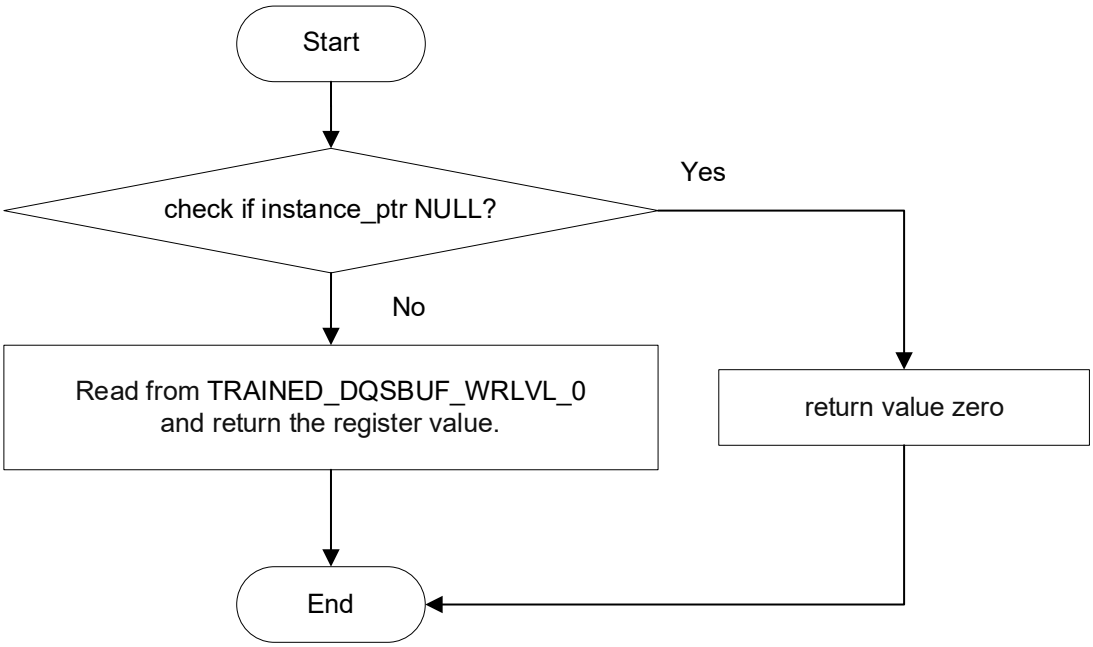

Figure 3.18. lpddr4_GetTrainedDqsbufWrlvl0Reg() Function Flow Call

3.19. lpddr4_GetTrainedDqsbufWrlvl1Reg()

This section shows the function call flow for the *lpddr4_GetTrainedDqsbufWrlvl1Reg()* API.

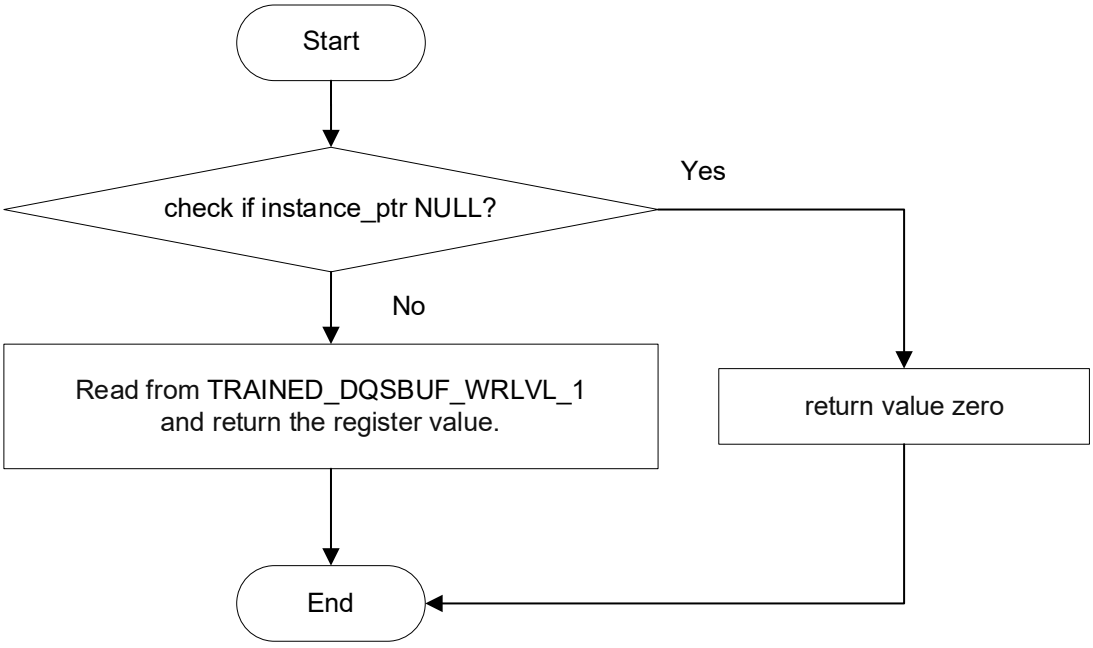

Figure 3.19. lpddr4_GetTrainedDqsbufWrlvl1Reg() Function Flow Call

3.20. lpddr4_GetTrainedVal1Reg()

This section shows the function call flow for the *lpddr4_GetTrainedVal1Reg()* API.

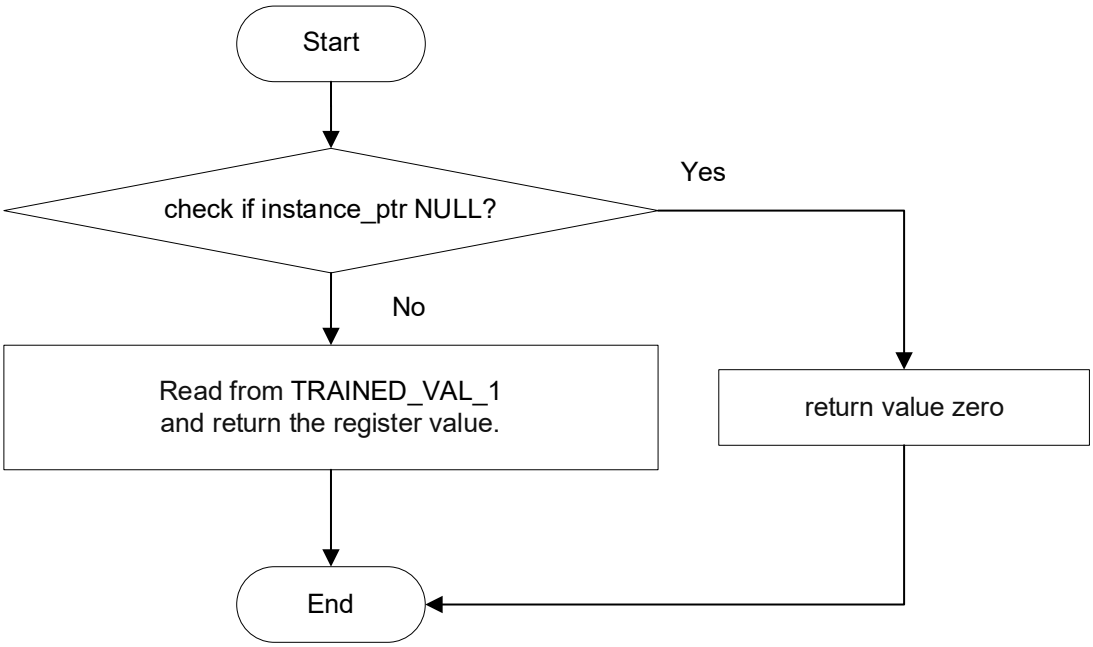

Figure 3.20. lpddr4_GetTrainedVal1Reg() Function Flow Call

3.21. lpddr4_GetTrainedVal2Reg()

This section shows the function call flow for the *lpddr4_GetTrainedVal2Reg()* API.

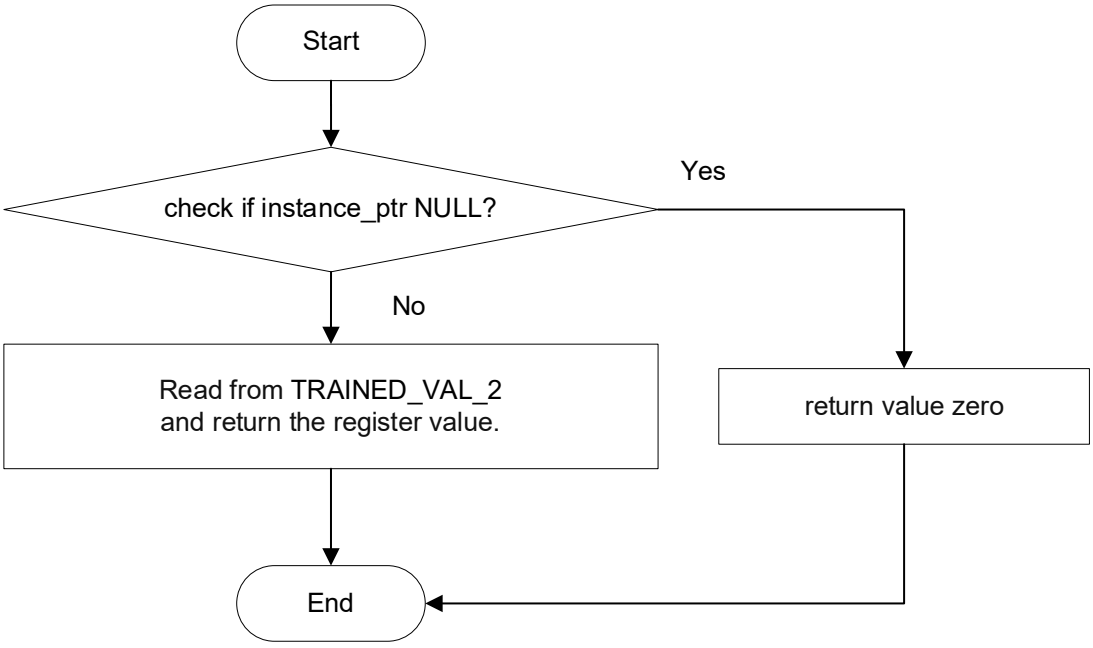

Figure 3.21. lpddr4_GetTrainedVal2Reg() Function Flow Call

3.22. lpddr4_GetTrainedVal3Reg()

This section shows the function call flow for the *lpddr4_GetTrainedVal3Reg()* API.

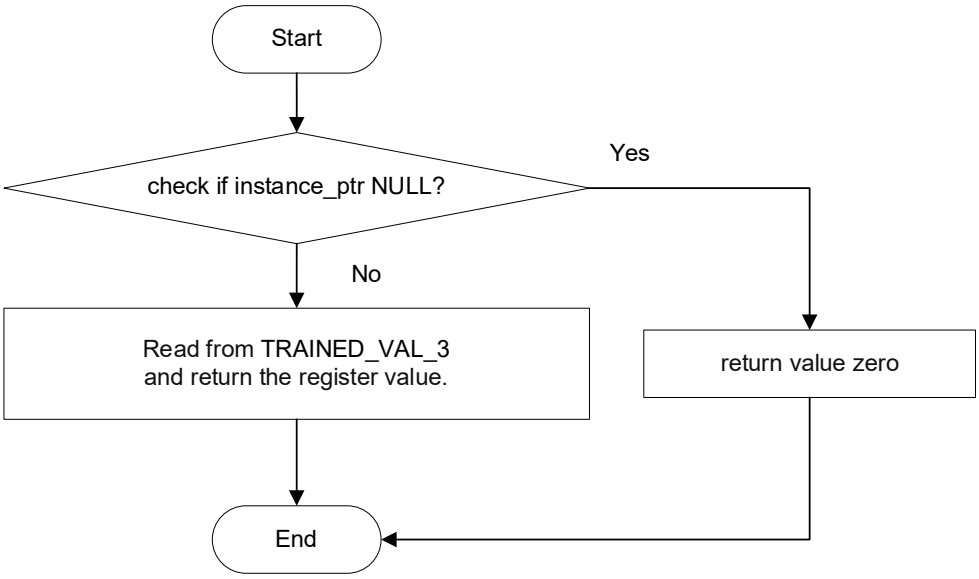

Figure 3.22. lpddr4_GetTrainedVal3Reg() Function Flow Call

3.23. lpddr4_SetTrainingOperationReg()

This section shows the function call flow for the *lpddr4_SetTrainingOperationReg()* API.

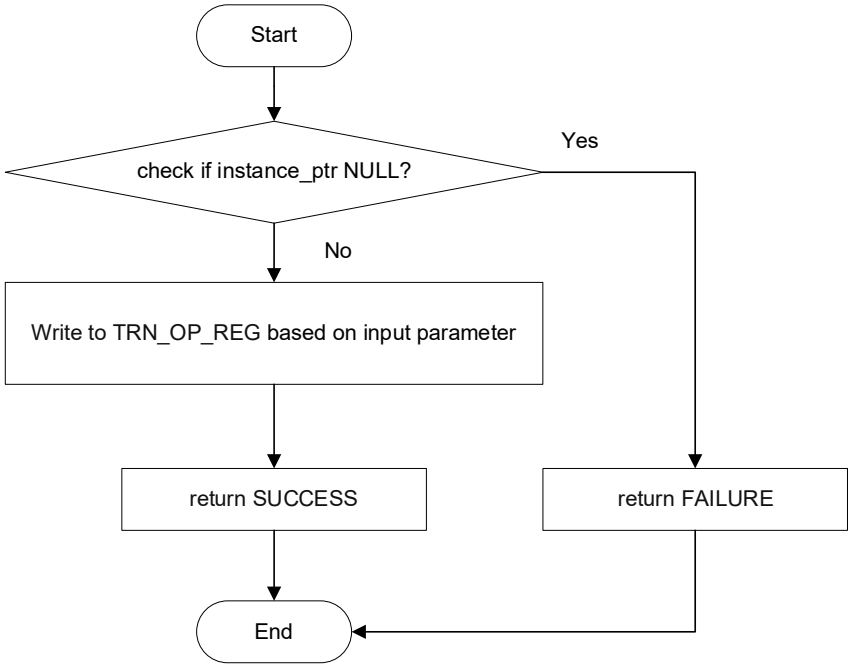

Figure 3.23. lpddr4_SetTrainingOperationReg() Function Flow Call

3.24. lpddr4_SetResetReg()

This section shows the function call flow for the *lpddr4_SetResetReg()* API.

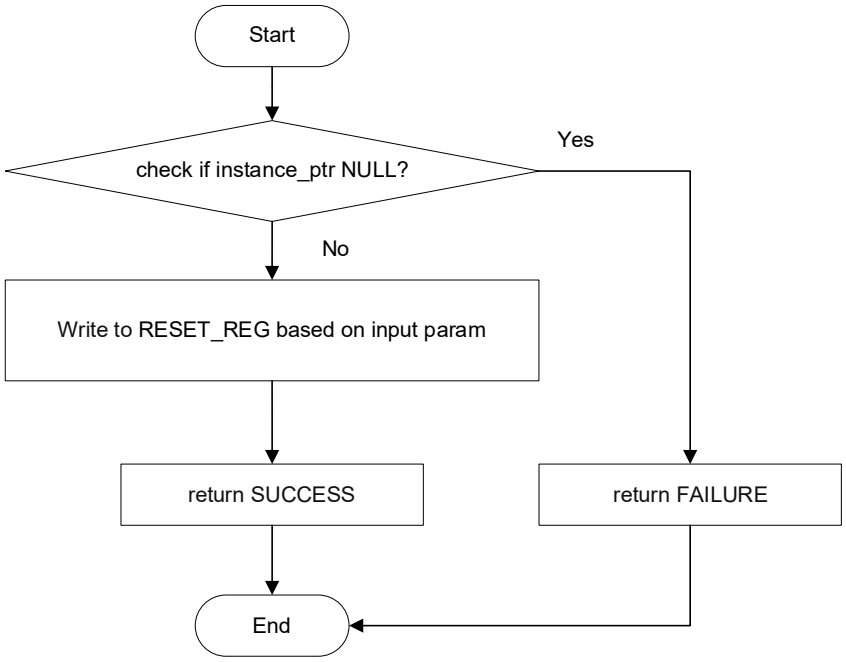

Figure 3.24. lpddr4_SetResetReg() Function Flow Call

3.25. lpddr4_SetInterruptEnableReg()

This section shows the function call flow for the *lpddr4_SetInterruptEnableReg()* API.

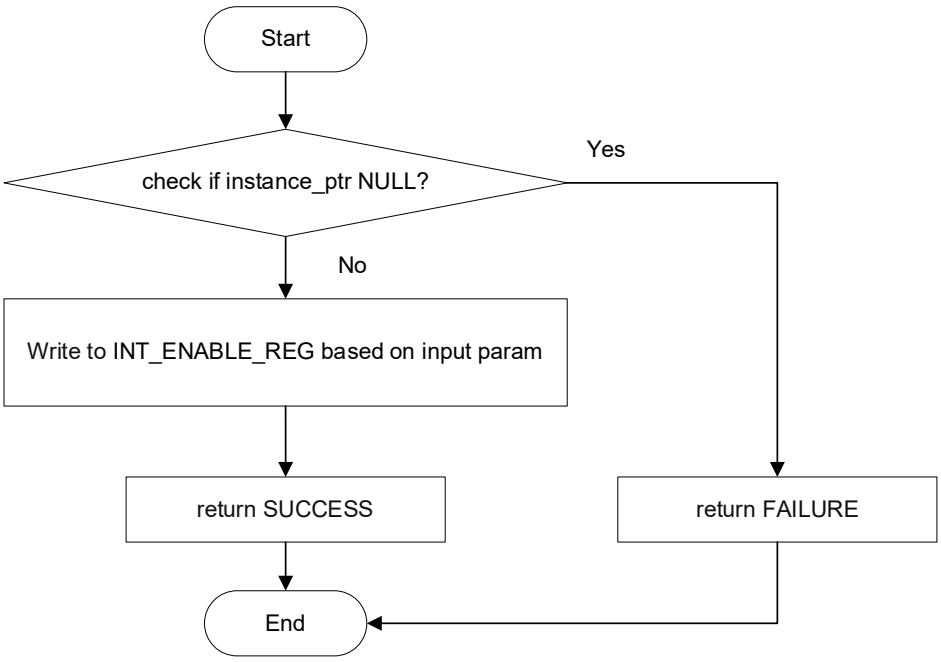

Figure 3.25. lpddr4_SetInterruptEnableReg() Function Flow Call

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 for the LPDDR4 memory controller.

4.2. Union and Struct feature_ctrl_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.2. feature_ctrl_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the feature_ctrl_reg_t register members.
unsigned int	reg	Use this 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	Enables data bus inversion (DBI) to reduce DDR data-signal toggling, improving signal integrity and reducing dynamic power consumption, as specified by the Enable DBI attribute.
unsigned int	power_down_en	Enables power-saving mode by putting the memory in self-refresh when there is no traffic for sclk_o cycles, as specified by the Number of SCLK to Enter Self-Refresh from No Traffic attribute.
unsigned int	gear_ratio	Must be 1, which corresponds to 8:1 gearing, and specifies the DDR-to-system ratio: ddr_ck_o frequency = 4 × sclk_o frequency.
unsigned int	addr_translation	Specifies the mapping the local memory-mapped address bits and memory address bits (row, column, bank, and rank). Only one address-translation scheme is currently supported (addr_translation=0). Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) IP user guide for details.
unsigned int	ddr_type	Must be 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	Number of ranks (0 = 1 rank, 1 = 2 ranks)
unsigned int	dgb_mem_en	Debug memory access enable.
unsigned int	fea_rsvd_2	Reserved.

4.3. Union and Struct reset_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.3. reset_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the reset_reg_t register members.
unsigned int	reg	Use this variable to access the reset_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	trn_eng_rst_n	De-assert the reset to the training engine to begin memory initialization and training. Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) IP user guide for details.
unsigned int	cpu_reset_n	De-assert the reset to the internal CPU to begin memory initialization and training. Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) IP user guide for details.
unsigned int	res_rsvd	Reserved.

4.4. Union and Struct settings_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.4. Settings_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the settings_reg_t register members.
unsigned int	reg	Use this variable to access the settings_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	write_latency	Specifies the number of DDR clock cycles from the write command to the first write data.
unsigned int	set_rsvd_1	Reserved.
unsigned int	read_latency	Specifies the number of DDR clock cycles from the read command to the first read data.
unsigned int	set_rsvd_2	Reserved.
unsigned int	clk_freq	Specifies the DDR (Double Data Rate) command clock frequency in MHz.
unsigned int	set_rsvd_3	Reserved.

4.5. Union and Struct int_status_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.5. Int_status_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the int_status_reg_t register members.
unsigned int	reg	Use this 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 complete successfully. 0 – No interrupt 1 – Interrupt pending

Data Type	Union Member	Description
unsigned int	trn_err_int	Training Error Interrupt. This interrupt bit asserts when the training engine encounters an error during training. To determine the specific error, read the status register (STATUS_REG). 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.

4.6. Union and Struct int_enable_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.6. int_enable_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the int_enable_reg_t register members.
unsigned int	reg	Use this 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.

4.7. Union and Struct clk_chng_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.7. Clk_chng_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the clk_chng_t register members.
unsigned int	reg	Use this variable to access the clk_chng_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	ddr_clk_sel	Indicates the target DDR clock frequency for a clock-frequency change. The training engine uses this register. You must not write 1 to this register.

Data Type	Union Member	Description
unsigned int	dll_update_en	Recalibrates the DLL after a clock-frequency change when asserted. The training engine uses this register. You must not write 1 to this register.
unsigned int	clk_update	Initiates a clock- frequency change when asserted. The training engine uses this register. You must not write 1 to this register.
unsigned int	pll_lock	Asserts when PLL is locked.
unsigned int	clk_chng_rsvd	Reserved.

4.8. Union and Struct trn_op_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.8. trn_op_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the trn_op_reg_t register members.
unsigned int	reg	Use this variable to access the trn_op_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	init_en	Lets you shorten initialization for simulation. 0 – Initialization is reduced. For example, reset time and CKE-low time are shortened. 1 – Initialization is performed according to the LPDDR4 JEDEC standard.
unsigned int	cbt_en	Enables the Command Bus Training (CBT) during initialization and training. CBT performs CA_VREF programming and aligns CS/CA and CK for high-frequency operation. 0 – The training engine shortens the CBT instead of iterating through different CA delays to find the optimal delay value. It programs only the trained CA delay attribute to the PHY. 1 – The training engine performs CBT 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 the DQ feedback is high after the DQS pulse. 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 given latency setting, the training engine increments the latency setting by 1 and retries read training. Once the read DQS burst is captured correctly, the training engine trains the DQS-DQ timing 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 is successful. 1 – The training engine performs read training.

Data Type	Union Member	Description
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 given latency setting, the training engine increments the latency setting by 1 and retries write training. 0 – The training engine programs the trained write DQ/DBI delay and trained write-latency attributes to the PHY, and checks that write and read FIFO access are equal. 1 – The training engine performs write training.
unsigned int	ca_vref_training_en	Enables V_{REF} training on the command-and-address bus as part of the CBT. V_{REF} training is embedded in the CBT flow, where the Graphical User Interface (GUI) values are read in as starting values. 0 – V_{REF} training is not included in the CBT routine. 1 – The training engine performs command-and-address V_{REF} training as part of the CBT.
unsigned int	mc_vref_training_en	Enables V_{REF} training on the memory controller as part of the read training. V_{REF} training is embedded in the training flow, where the V_{REF} values in the GUI are read in as starting values. 0 – Disables V_{REF} training during read training. 1 – The training engine performs memory controller V_{REF} training as part of the read training.
unsigned int	mem_vref_training_en	Enables V_{REF} training on the SDRAM memory as part of the write training. V_{REF} training is embedded in the training flow, where the V_{REF} values in the GUI are read in as starting values. 0 – V_{REF} training is not included in write training. 1 – The training engine performs SDRAM memory V_{REF} training as part of the write training.
unsigned int	trn_op_rsvd1	Reserved.

4.9. Union and Struct status_reg_t

Refer to the LPDDR4 Memory Controller for Nexus Devices (FPGA-IPUG-02127) user guide for the register description.

Table 4.9. Status_reg_t Parameters

Data Type	Union Member	Description
Struct	fields	Use this variable to access the status_reg_t register members.
unsigned int	reg	Use this variable to access the status_reg_t register.
Data Type	**fields struct Members**	**Description**
unsigned int	phy_ready	Asserts when the Physical Layer (PHY) initialization completes and the PHY is ready for operation.
unsigned int	cbt_done	Asserts when CBT completes.
unsigned int	write_lvl_done	Asserts when write leveling completes.
unsigned int	read_trn_done	Asserts when read training completes.
unsigned int	write_trn_done	Asserts when write training completes.
unsigned int	in_self_refresh	Asserts when a memory is in self-refresh. - 2'b00: no rank in self-refresh - 2'b01: rank 0 is in self-refresh - 2'b10: rank 1 is in self-refresh - 2'b11: rank 0 and rank 1 are in self-refresh
unsigned int	sts_rsvd1	Reserved.
unsigned int	cbt_err	Asserts when a failure occurs during CBT.
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 read_trn_done is low (STATUS_REG[3]=0), the training failed to find a setting that captures the read DQS burst. If read_trn_done is high (STATUS_REG[3]=1), the training failed to find an optimal read DQS-DQ delay for capturing read data.
unsigned int	write_trn_err	Asserts when a failure occurs during write training.
unsigned int	err_on_rank	Indicates which rank has an error. - 2'b00: no error - 2'b01: error on rank 0 - 2'b10: error on rank 1 - 2'b11: error on both rank 0 and rank 1
unsigned int	sts_rsvd2	Reserved.
unsigned int	refresh_rate	Reflects the refresh-rate value set in LPDDR4 MR4 and specifies the required refresh period based on the device temperature. 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 completes for rank 0.
unsigned int	rank1_done	Asserts when training completes for rank 1.
unsigned int	sts_rsvd3	Reserved.

The following APIs are reserved for internal use:

- Union and Struct trn_patrn_reg_t
- Union and Struct trained_dqsbuf_wrlvl_0_t
- Union and Struct trained_dqsbuf_wrlvl_1_t
- Union and Struct trained_val_1_t
- Union and Struct trained_val_2_t
- Union and Struct trained_val_3_t

5. API Enum

5.1. Enum LPDDR_RET

Table 5.1. LPDDR_RETVariables

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 Macros

Table 6.1. API Macros Description

Macro	Description
#define LPDDR4_DRV_VER	LPDDR4 driver version.
#define FEATURE_CTRL_REG	Address of the Feature Control Register.
#define RESET_REG	Address of the Reset Register.
#define SETTINGS_REG	Address of the Settings Register.
#define INT_STATUS_REG	Address of the Interrupt Status Register.
#define INT_ENABLE_REG	Address of the Interrupt Enable Register.
#define INT_SET_REG	Address of the Interrupt Set Register (for debug purposes only).
#define CLK_CHANGE_REG	Address of the Clock Change Register.
#define TRN_OP_REG	Address of the Training Operation Register.
#define STATUS_REG	Address of the Status Register.
#define TRN_PATRN_REG	Reserved for internal use.
#define WAIT_TIMEOUT_REG	Reserved for internal use.
#define TIMER_RELOAD_REG	Reserved for internal use.
#define TRAINED_DQSBUF_RDCLKSEL	Reserved for internal use.
#define TRAINED_VAL_0	Reserved for internal use.
#define TRAINED_DQSBUF_WRLVL_0	Reserved for internal use.
#define TRAINED_DQSBUF_WRLVL_1	Reserved for internal use.
#define TRAINED_VAL_1	Reserved for internal use.
#define TRAINED_VAL_2	Reserved for internal use.
#define TRAINED_VAL_3	Reserved for internal use.
#define SCRATCH_0_REG	Reserved for internal use.
#define SCRATCH_1_REG	Reserved for internal use.
#define ERROR_LOG_REG	Reserved for internal use.
#define LATENCY_MASK	Reserved for internal use.
#define RD_LATENCY_POS	Reserved for internal use.
#define DDR_FREQ_POS	Reserved for internal use.
#define DDR_WIDTH_POS	Reserved for internal use.
#define BYTE_MASK	Reserved for internal use.
#define CLK_DELAY_MASK	Reserved for internal use.
#define DDR_WIDTH_X32	Reserved for internal use.
#define DDR_WIDTH_X64	Reserved for internal use.
#define STATUS_CBT_DONE	Reserved for internal use.
#define STATUS_WRLVL_DONE	Reserved for internal use.
#define STATUS_RD_TRN_DONE	Reserved for internal use.
#define STATUS_WR_TRN_DONE	Reserved for internal use.
#define STATUS_CBT_ERR	Reserved for internal use.
#define STATUS_WRLVL_ERR	Reserved for internal use.
#define STATUS_RD_TRN_ERR	Reserved for internal use.
#define STATUS_WR_TRN_ERR	Reserved for internal use.
#define RANK0_TRN_DONE	Reserved for internal use.
#define RANK1_TRN_DONE	Reserved for internal use.
#define VREF_DIFF_THRESHOLD	Reserved for internal use.
#define SUCCESS	Success.
#define FAILURE	Failure.
#define IS_NULL	Null value.

Macro	Description
#define OUT_OF_RESET	Out-of-reset value used to release the training engine and CPU from reset.
#define LPDDR_DONE_BITS	LPDDR initialization done status bits.
#define LPDDR_ERR_DONE_BITS	LPDDR initialization error-done status bits.
#define PLL_LOCK_BIT	LPDDR PLL lock status bit.
#define TRN_EN	LPDDR training enable status bits.
#define TWO_US_DELAY	Delay in microseconds.
#define TPAT_DQS_GRP_VREF_POS	Reserved for internal use.
#define TPAT_MEM_CA_VREF_POS	Reserved for internal use.
#define TPAT_MEM_DQ_VREF_POS	Reserved for internal use.
#define TPAT_DQS_GRP_VREF_MASK	Reserved for internal use.
#define TPAT_MEM_CA_VREF_MASK	Reserved for internal use.
#define TPAT_MEM_DQ_VREF_MASK	Reserved for internal use.

References

- LPDDR4 Memory Controller IP Core for Nexus Devices User Guide (FPGA-IPUG-02127)
- LPDDR4 Memory Controller IP Release Notes (FPGA-RN-02057)
- Lattice Propel Builder User Guide (FPGA-UG-02243)
- Lattice Solutions IP Cores web page
- Lattice Propel Design Environment web page
- Lattice Radiant Software 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, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.2, June 2026

Section	Change Summary
All	• Renamed API identifiers, parameter names, and the structure name to lowercase, such as, *LPDDR_init()* to *lpddr4_init()*, *InstancePtr* to *instance_ptr*, and *LPDDR4* structure to *lpddr4* structure. • Minor editorial fixes.
Abbreviations in This Document	Added the following abbreviations: DDR, GUI, PHY, and PLL.

Introduction	• Updated Table 1.1. Driver and IP Version.
	  • Driver version updated to *26.01.00.*
	  • IP version updated to *2.7.0*.
	• Added reference to the LPDDR4 Memory Controller IP Release Notes (FPGA-RN-02057) in the Driver and IP Compatibility section.
	• Updated Table 1.2. Quick Facts of Driver Tested Environment.
	  • All tool version updated to *2025.2.*

API Description	• Added the following APIs:
	  • lpddr4_wait_pll_lock()
	  • lpddr4_bring_out_of_reset()
	  • lpddr4_start_training()
	  • lpddr4_do_training()
	  • lpddr4_GetClkChangeReg()
	  • lpddr4_GetTrainingPatternReg()
	  • lpddr4_GetTrainedDqsbufWrlvl0Reg()
	  • lpddr4_GetTrainedDqsbufWrlvl1Reg()
	  • lpddr4_GetTrainedVal1Reg()
	  • lpddr4_GetTrainedVal2Reg()
	  • lpddr4_GetTrainedVal3Reg()
	  • lpddr4_SetTrainingOperationReg()
	  • lpddr4_SetResetReg()
	  • lpddr4_SetInterruptEnableReg()
	• Updated the return type of the following APIs to return the register struct directly.
	  • lpddr4_GetFeatureControlReg()
	  • lpddr4_GetSettingReg()
	  • lpddr4_GetTrainingOperationReg()
	  • lpddr4_GetStatusReg()
	• Updated the lpddr4_start_training() write-up.
	• Removed the reg_data out-parameter in this section.

Function Call Flow Diagrams	• Updated all figure captions in this section to include *Function Call Flow.*
	• Replaced Figure 3.1. lpddr4_init() Function Flow Call to reflect the new sub-API decomposition.
	• Added function call flow diagrams for the new APIs:
	  • Figure 3.2. lpddr4_wait_pll_lock() Function Flow Call through Figure 3.5. lpddr4_do_training() Function Flow Call
	  • Figure 3.14. lpddr4_GetClkChangeReg() Function Flow Call
	  • Figure 3.17. lpddr4_GetTrainingPatternReg() Function Flow Call through Figure 3.25. lpddr4_SetInterruptEnableReg() Function Flow Call.

API Data Structures	• Reordered subsections to match the register-access sequence used by the driver.
	  • *clk_chng_t* moved from *subsection 4.3* to *subsection 4.7*.
	  • *status_reg_t* moved from *subsection 4.7* to *subsection 4.9*.
	  • *int_enable_reg_t* moved from *subsection 4.8* to *subsection 4.6*.
	• Updated Table 4.9. Status_reg_t Parameters.
	• Updated the *in_self_refresh* field description from a *1-bit assert* to the *2-bit value*

Section	Change Summary

	encoding (2'b00, 2'b01, 2'b10, 2'b11).
	• Updated the *err_on_rank* field description with the full 2-bit value encoding covering no error, rank 0, rank 1, and both ranks.
	• Listed the following APIs reserved for internal use:
	  • trn_patrn_reg_t
	  • trained_dqsbuf_wrlvl_0_t
	  • trained_dqsbuf_wrlvl_1_t
	  • trained_val_1_t
	  • trained_val_2_t
	  • trained_val_3_t.

API Variables	Removed this section.

API Macros	• Added the following macros:
	  • #define TPAT_DQS_GRP_VREF_POS
	  • #define TPAT_MEM_CA_VREF_POS
	  • #define TPAT_MEM_DQ_VREF_POS
	  • #define TPAT_DQS_GRP_VREF_MASK
	  • #define TPAT_MEM_CA_VREF_MASK
	  • #define TPAT_MEM_DQ_VREF_MASK
	• Reorganized the macros list to include internal-use macros marked *reserved for internal use*.
	• Removed macro #define RSVD_1.

References	• Updated the Lattice Propel Builder User Guide reference to FPGA-UG-02243. • Added a reference to the LPDDR4 Memory Controller IP Release Notes (FPGA-RN-02057).

Revision 1.1, February 2025

Section	Change Summary
All	Minor editorial fixes.
Introduction	Updated the following in Table 1.1. Driver and IP Version. • Driver Version: *24.02.00* • IP Version: *2.4.0*
Function Call Flow Diagrams	Updated Figure 3.1. LPDDR4_init().
API Macros	Updated the following in Table 6.1. API Macros Description. • Updated the macro #define_CLK_CHANGE to *#define_CLK_CHANGE_ REG* • Added the macro *#define_TWO_US_DELAY* • Deleted the macro *#define TRN_BIT_LVL_SWEEP_EN*

Revision 1.0, August 2024

Section	Change Summary
All	Initial release.

www.latticesemi.com