Hardware System Verification (HSV)
Vertical Solutions Engineering (VSE)

LPDDR4
Palladium Memory Model
User Guide

Document Version: 3.2

Document Date: July 2018

Contents

List of Figures

List of Tables

1. General Information

The Cadence Memory Model Portfolio provides memory device models for the Cadence Palladium XP, Palladium XP II and Palladium Z1 series systems. Optimizing the acceleration and/or emulation flow on these platforms for MMP memory models may require information outside the scope of the MMP user guides and related MMP documentation.

1.1 Related Publications

For basic information regarding emulation and acceleration, please refer to the following documents:

For Palladium XP and Palladium XP II:

UXE User Guide
UXE Library Developer's Guide
UXE Known Problems and Solutions
UXE Command Reference Manual
Palladium XP Planning and Installation Guide
Palladium Target System Developer's Guide
What's New in UXE

For Palladium Z1:

VXE User Guide
VXE Library Developer's Guide
VXE Known Problems and Solutions
VXE Command Reference Manual
Palladium Z1 Planning and Installation Guide
Palladium Target System Developer's Guide
What's New in VXE

LPDDR4 Memory Model

1. Introduction

The Cadence Palladium LPDDR4 Model is based on the datasheet listed below.

Table 1: Memory Model Standard or Datasheet Revision Info

Association or Vendor	Part	Reference Datasheet	Rev.	Revision Date
JEDEC	jedec_lpddr4_*	JESD209-4B.pdf	JESD209-4B	February 2017
JEDEC	jedec_lpddr4x_*	TG426_8^20160805^1831.55^Micron^LPDDR4X_spec_addendum_edit0_2.pdf		August 2016
JEDEC	jedec_lpddr4_x8_*	JC426^20150828^1831 31^Micron^LPDDR4_Byte_Mode_Spec_Addendum.pdf		August 2015
Samsung	samsunglpddr4_*	Samsung LPDDR4 SDRAM Operations & Timing_R04_20130705_Red.pdf	0.4	July 2013
SKHynix	sklpddr4_4ch_16Gb	272b_H9HKNNNBTUMUAR_Series_Rev0p8.pdf	0.81	July 2013
Micron	mt53b128m32	200b_z00n_auto_lpddr4.pdf	Rev.C	January 2017
	mt53b384m32d2 mt53b768m32d4	200b_z0am_ddp_embedded_lpddr4.pdf	Rev.D	March 2017
	mt53b256m32d1 mt53b512m32d2 mt53b1024m32d4	200b_z01m_sdp_ddp_qdp_mobile_lpddr4.pdf	Rev.E	January 2017
	mt53b384m64d4	366b_z0am_qdp_mobile_lpddr4.pdf	Rev.D	February 2017
	mt53b256m64d2	366b_z01m_ddp_qdp_mobile_lpddr4.pdf	Rev.E	November 2016
	mt53b768m64d8	376b_j_z0aq_8dp_mobile_lpddr4.pdf	Rev.B	April 2017
	mt53b512m64d4 mt53b1024m64d8	376b_j_z11m_qdp_8dp_mobile_lpddr4.pdf	Rev.B	March 2017
	mt53d512m64d8	366b_q_z00m_8dp_mobile_lpddr4x.pdf	Rev.C	October 2016
	mt53d768m64d8	376b_j_z0aq_8dp_mobile_lpddr4x.pdf	Rev.B	April 2017
	mt53d512m64d4 mt53d1024m64d8	376b_j_z11m_qdp_8dp_mobile_lpddr4x.pdf	Rev.C	April 2017

2. Model Release Levels

All models in the Memory Model Portfolio are graded with a release level. This release level informs users of the current maturity and status of the model. All families in the library are graded at one of these levels.

The different levels give an overall indication of the amount of testing, level of quality and feature availability in the model. For details on supported features check the User Guide for that particular model family.

There are three release levels for models in the MMP release.

Release Level		Model Status	Available in Release	Listed in Catalog	Requires Beta Agreement
Mainstream Release	MR	Fully released and available in the catalog for all customers to use.	Yes	Yes	No
Emerging Release	ER	Model has successfully completed Beta engagement(s). Most, but not all features have been tested. Documentation is available.	No	Yes	Yes
Initial Release	IR	Model has completed initial development and has been released to Beta customer(s). The model may have missing features, may not be fully tested and may not have documentation. Model may contain defects.	No	Yes	Yes

Access to Initial Release and Emerging Release versions of the models will require a Beta Agreement to be signed before the model can be delivered.

3. Features

The table below lists which features are supported and which are unsupported.

Table 2: Features List of LPDDR4 Model

FEATURE	SUPPORT	NOTE
Initialization sequence	Yes	
Activate command	Yes	
Burst Read operation	Yes	
Burst Write operation	Yes	
Write Data Mask	Yes	
Precharge/auto-precharge operation	Yes	
Refresh/self-refresh command	Yes	
Read/Write DBI function	Yes	
Burst Length 16, 32 and BL on the fly function	Yes	
Read/Write Pre-amble and Post-amble features	Yes	
FSP-WR and FSP-OP features	Yes	
Command Bus Training feature	Yes	
Write Leveling feature	Yes	
MRR data extending to first 4 UI	Yes	
MPC command and its sub commands (NOP, Read/Write DQ FIFO, Read DQ Training and ZQ Calibration)	Yes	
TRR function	Yes	
SRE and SRX commands	Yes	SRE and SRX commands are accepted for LPDDR4 model, but have no effect on the core memory
ODT feature	No	non_target DRAM is supported for LPDDR4_x8, but it has no effect on ODT feature
PPR function	No	
PASR Bank and PASR Segment functions	No	
DQS Oscillator Count	Yes	
ZQCal Reset function	No	
Timing TDQSCK/TDQSS/TDQS2DQ	Yes	The unit is half cycle of input clock. Refer to section 13 for detail.
LPDDR4X	Yes	
LPDDR4 x8 (Byte Mode)	Yes	

4. Verilog Macro Defines

The following table lists the optional Verilog macro defines the user may want to consider.

Table 3: LPDDR4 Optional Verilog Defines

`define Macro Purpose	Optional Verilog `define Values
DQS Interval Oscillator feature support; define to enable DQS Interval Oscillator. MR18 and MR19 will report the DRAM DQS Oscillator count only when this macro is defined.	MMP_DQSOSC
TDQS2DQ function support; define to enable the timing for TDQS2DQ function.	MMP_ENH_TIMING_ACCURACY

5. Configurations

The following table lists the possible configurations. Not all configurations are available from all vendors. Please consult the appropriate vendor site for details on the parts they offer.

Table 4: Model Configurations

Data Width	Size	4Gb	6Gb	8Gb	12Gb	16Gb	24Gb	32Gb
	Type	S16	S16	S16	S16	S16	S16	S16
	Channels	2	2	2	2	2	2	2
	Banks	8	8	8	8	8	8	8
x16	Row Address	R[13:0]	R[14:0] (R13=0 when R14=1)	R[14:0]	R[15:0] (R14=0 when R15=1)	R[15:0]	R[16:0] (R15=0 when R16=1)	R[16:0]
	Column Address	C[9:0]	C[9:0]	C[9:0]	C[9:0]	C[9:0]	C[9:0]	C[9:0]
x8	Row Address	R[14:0]	R[15:0] (R14=0 when R15=1)	R[15:0]	R[16:0] (R15=0 when R16=1)	R[16:0]	NA	NA
	Column Address	C[9:0]	C[9:0]	C[9:0]	C[9:0]	C[9:0]	NA	NA

Data Width	Size	2Gb	3Gb	4Gb	6Gb	8Gb		
	Type	S16	S16	S16	S16	S16		
	Channels	1	1	1	1	1		
	Banks	8	8	8	8	8		
x16	Row Address	R[13:0]	R[14:0] (R13=0 when R14=1)	R[14:0]	R[15:0] (R14=0 when R15=1)	R[15:0]		
	Column Address	C[9:0]	C[9:0]	C[9:0]	C[9:0]	C[9:0]		
x8	Row Address	R[14:0]	R[15:0] (R14=0 when R15=1)	R[15:0]	R[16:0] (R15=0 when R16=1)	R[16:0]		
	Column Address	C[9:0]	C[9:0]	C[9:0]	C[9:0]	C[9:0]		

6. Model Block Diagram

For normal LPDDR4 models, the width of CA is 6 bits, DMI is 2 bits, DQ is 16 bits, DQS and DQS_N are 2-bit buses. For LPDDR4 byte mode models, the width of CA is 6 bits, DMI is 1 bit, DQ is 8 bits, DQS and DQS_N are 1-bit buses.

There are described below a 2-channel model and 1-channel model based on JESD209-4B and Samsung specification, a 4-channel model based on an SK Hynix specification, and a 2-channel byte mode model based on JEDEC Byte Mode Spec Addendum.

An additional VCC_UP pin is used to model the power-on condition. If modeling the power-on condition is desired, then initially drive the VCC_UP pin low and then assert it high after some time to indicate power on. If it is not needed, then simply tie this pin high. ODT_CA_A and ODT_CA_B are not implemented yet; the user can simply tie these pins low.

The DQSOSC_CLK_* clock pins are optional. These pins are enabled only by defining Verilog macro “MMP_DQSOSC”. For more information, please refer to the section in this user guide related to the DQS Interval Oscillator.

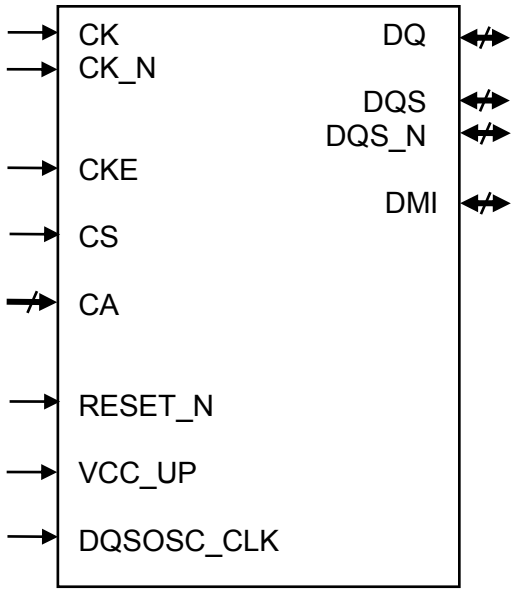


Figure 1: 1-Channel LPDDR4

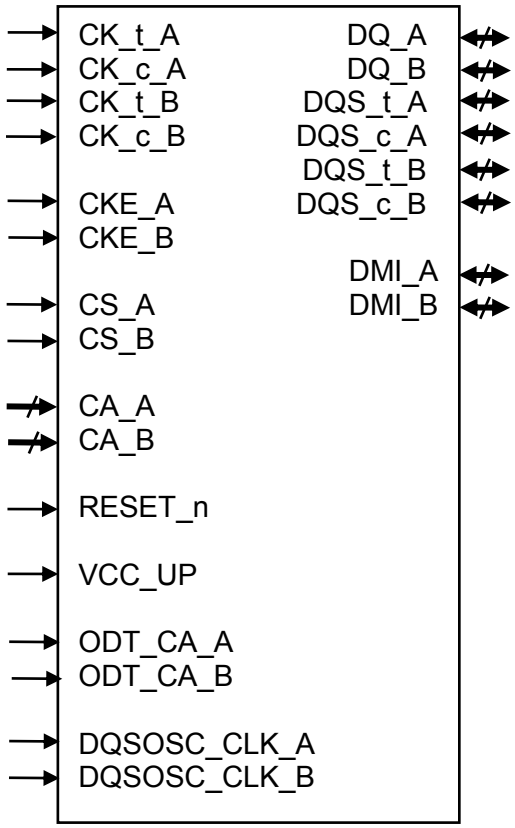


Figure 2: 2-Channel LPDDR4

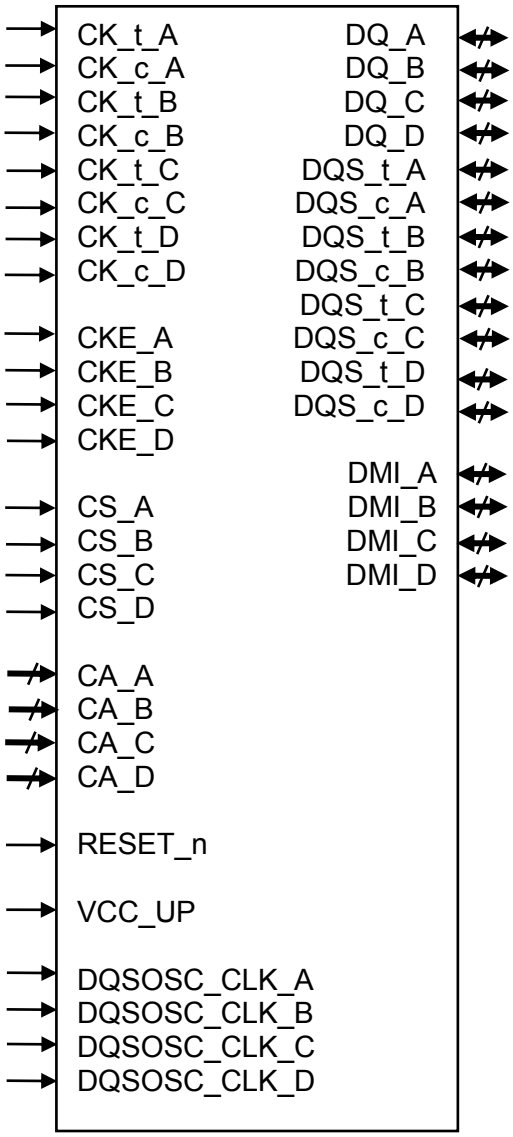


Figure 3: 4-Channel LPDDR4

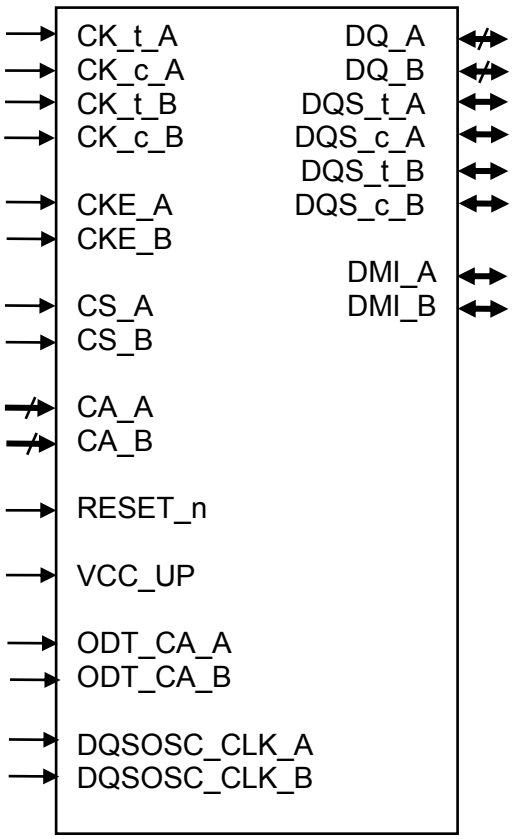


Figure 4: 2-Channel Byte Mode LPDDR4

7. I/O Signal Description

The table below lists and describes the model I/O signals.

Table 5: Model I/O Signals

NAME	TYPE	DESCRIPTION
CK_t_A CK_c_A CK_t_B CK_c_B CK_t_C CK_c_C CK_t_D CK_c_D	Input	Clock: CK_t and CK_c are differential clock inputs. All address, command, and control input signals are sampled on the crossing of the positive edge of CK_t and the negative edge of CK_c. AC timings for CA parameters are referenced to CK. Each channel has its own clock pair.
CKE_A CKE_B CKE_C CKE_D	Input	Clock Enable: CKE HIGH activates and CKE LOW deactivates the internal clock circuits, input buffers, and output drivers. Power-saving modes are entered and exited via CKE transitions. CKE is part of the command code. Each channel has its own CKE signal.
CS_A CS_B CS_C CS_D	Input	Chip Select: CS is part of the command code. Each channel has its own CS signal.
CA_A[5:0] CA_B[5:0] CA_C[5:0] CA_D[5:0]	Input	Command/Address Inputs: Provide the Command and Address inputs according to the Command Truth Table. Each channel has its own CA signals.
DQ_A[15:0] DQ_B[15:0] DQ_C[15:0] DQ_D[15:0]	I/O	Data Inputs/Outputs: Bi-directional data bus
DQS_t_A[1:0] DQS_c_A[1:0] DQS_t_B[1:0] DQS_c_B[1:0] DQS_t_C[1:0] DQS_c_C[1:0] DQS_t_D[1:0] DQS_c_D[1:0]	I/O	Data Strobe: DQS_t and DQS_c are bi-directional differential signals used to strobe data during a READ or WRITE. The Read Strobe is generated by the DRAM for a READ and is edge aligned with Data. The Write Strobe is generated by the SOC Memory Controller for a WRITE and is trained to proceed data. Each byte of data has a Read Strobe signal pair. Each channel has its own DQS strobes.
ODT_CA_A ODT_CA_B ODT_CA_C ODT_CA_D	Input	Not supported in the model.
RESET_n	Input	RESET: When asserted LOW, the RESET pin resets both channels of the die.
DMI_A[1:0] DMI_B[1:0] DMI_C[1:0] DMI_D[1:0]	I/O	Data Mask Inversion: DMI is a bi-directional signal which is driven HIGH when the data on the data bit is inverted, or driven LOW when the data is in its normal state. Data Inversion can be disabled via a setting. Each byte of data has a DMI signal. Each channel has its own DMI signals. DMI can be used for data mask when data inversion is disabled.

NAME	TYPE	DESCRIPTION
DQSOSC_CLK_A DQSOSC_CLK_B DQSOSC_CLK_C DQSOSC_CLK_D	Input	DQS Oscillator Clock: DQSOSC_CLK is the clock used to drive the DQS interval Oscillator. This clock input is optional. It is enabled only by defining Verilog macro “MMP_DQSOSC”.

8. Model Parameter Descriptions

The following table provides details on the user adjustable parameters for the Palladium LPDDR4 Memory Model. These parameters may be modified when instantiating an LPDDR4 wrapper or, if necessary, by modifying the HDL parameter declarations and default values which are exposed for access and debug visibility.

Table 6: User Adjustable Parameters

User Adjustable Parameter	Default Value	Description
bank_addr_width	3	Bank address width
row_addr_width	14	Row address width
	15	Row address width (Byte Mode)
col_addr_width	10	Column address width
flag_for_6Gb_12Gb	0	Flag for 6Gb and 12Gb size 1: memory size if 6Gb or 12Gb or 24Gb 0: for 4Gb, 8Gb or 16Gb or 32Gb
tDQSCK	1	Integer part of the timing value TDQSCK. The unit is cycle of input CK.(1)
tDQSCK_half	1	Fractional part of the timing value TDQSCK. The unit is half-cycle of input CK.
tDQSS	1	Integer part of the timing value TDQSS. The unit is cycle of input CK.(2)
tDQSS_half	0	Fractional part of the timing value TDQSS. The unit is half-cycle of input CK.
tDQS2DQ	0	Integer part of the timing value TDQS2DQ. The unit is cycle of input CK.(3)
tDQS2DQ_half	0	Fractional part of the timing value TDQS2DQ. The unit is half-cycle of input CK.

Note:

(1) The value of timing TDQSCK=tDQSCK*1 + tDQSCK_half*0.5 (Unit of CK)
(2) The value of timing TDQSCK=tDQSS*1 + tDQSS_half*0.5 (Unit of CK)
(3) The value of timing TDQSCK=tDQS2DQ*1 + tDQS2DQ_half*0.5 (Unit of CK)

The following table provides some information about exposed localparams that are NOT user adjustable. On rare occasion the user may find one of these localparam needs adjusting for their configuration. If this case arises, please contact Cadence emulation or MMP support.

Table 7: Visible Non-User-Adjustable Localparam

Localparam	Default Value	Description
data_bits	16	Data bus width
	8	Data bus width (Byte Mode)
addr_bits	6	CA bus width
byte_width	8	Byte bits
mode_reg_width	8	Mode register address width
num_bytes	2	Data width in bytes
	1	Data width in bytes (Byte Mode)
total_addr_bits		Memory capacity address width
banks		Number of banks
rows		Number of rows
cols		Number of columns
tDQSS	1	Timing parameter tDQSS
tRP	4	Timing parameter tRP
tMRD	10	Timing parameter tMRD
dqfifo_bl	16	BL for DQ fifo
dqfifo_num	5	Number of DQ fifo
BL_MWR	16	BL for Mask write
BL_MRR	16	BL for Mode register read
BL_DQFIFO	16	BL for DQ fifo read/write
BL_DQTRAIN	16	BL for DQ training read

Note that there are additional exposed localparams in the model HDL that are not described here nor intended to be described here. These additional localparams are exposed for debugging purposes only and will not be described herein.

9. Address mapping

The array of the LPDDR4 model is mapped into the internal memory of the Palladium system. This array is a single two dimensional array. The mapping of bank, row and column addresses to the internal model array is as follows:

ARRAY_ADDR = (BA*rows + ROW)*cols + COL;

(rows: the maximum number of rows per bank; cols is the maximum number of columns per row.)

This information is required if the memory needs to be preloaded with user data.

The array name in the model hierarchy is: memcore
For multi-channel LPDDR4 models, the core memory array is inside file lpddr4_pd*.vp
For single-channel LPDDR4 models, the core memory array is inside file jedec_lpddr4_single_ch_*Gb.vp

If {row,ba,col} addressing is needed instead of the default {ba,row,col} addressing, add +define+MMP_RBC to the vlan invocation (IXCOM flow) or to the appropriate HDL-ICE synthesis (Classical flow) command. No value is required for MMP_RBC – only the compile phase define. This option is applicable when using the <model>.vp file. In MMP_RBC mode, the mapping of bank, row and column addresses to the internal model array is as follows:

ARRAY_ADDR = (ROW*banks+BA)*cols + COL;

10. Register Definitions

In the LPDDR4 specification there are up to 64 registers defined. The LPDDR4 Palladium model implements the following registers.

Table 8: Registers Implemented in Samsung and SK Models

MR#	MA <5:0>	Function	Access	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
0	00_H	Device Info	R	CATR	RFU	PPR	RZQI		RFU		
1	01_H	Device Feature 1	W	PST0	nWR0		RD-PRE0	WR-PRE0	BL		
				PST1	nWR1		RD-PRE1	WR-PRE1			
2	02_H	Device Feature 2	W	WR Lev	WL Select0	WL0			RL0		
					WL Select1	WL1			RL1		
3	03_H	I/O Config-1	W	DBI-WR0	DBI-RD0	PDDS0			RFU		PU-CAL0
				DBI-WR1	DBI-RD1	PDDS1					PU-CAL0
4	04_H	Refresh Rate	R/W	TUF	RFU		PPRE	TRR	Refresh Rate		
5	05_H	Basic Config-1	R	Manufacturer ID							
6	06_H	Basic Config-2	R	Revision ID-1							
7	07_H	Basic Config-3	R	Revision ID-2							
8	08_H	Basic Config-4	R	I/O width		Density			Type		
9	09_H	Test Mode	W	Vendor-Specific							
10	$0A_H$	IO Calibration	W	RFU							ZQ-RESET
11	$0B_H$	ODT Feature	W	ODTE-CA0	CA ODT0			ODTE-DQ0	DQ ODT0		
				ODTE-CA1	CA ODT1			ODTE-DQ1	DQ ODT1		
12	$0C_H$	VREF(ca) Setting/Range	R/W	RFU	VR-CA0	VREF0(ca)					
				RFU	VR-CA1	VREF1(ca)					
13	$0D_H$	CBT,VRO,DME,FSP-WR,FSP-OP	W	FSP-OP	FSP-WR	DM_DIS	RFU	VRCG	VRO	RFU	CBT
14	$0E_H$	VREF(dq) Setting/Range	R/W	RFU	VR-DQ0	VREF0(dq)					
				RFU	VR-DQ1	VREF1(dq)					
15	$0F_H$	Lower-Byte Invert for DQ Calibration	W	Lower Byte Invert Register for DQ Calibration[7:0]							
16	10_H	PASR Bank	W	PASR Bank Mask							
17	11_H	PASR Segment	W	PASR Segment Mask							
18	12_H	IT-LSB	R	DQS Oscillator Count-LSB[7:0]							
19	13_H	IT-MSB	R	DQS Oscillator Count-LSB[15:8]							
20	14_H	Upper-Byte Invert for DQ Calibration	W	DQ Calibration Invert Mask [15:8]							
22	16_H	Controller ODT	W	Vendor Specific				ODTE-CTRL0	CTRL ODT0		
				Vendor Specific				ODTE-CTRL1	CTRL ODT1		
32	20_H	DQ Calibration Pattern	W	DQ Calibration Pattern [7:0]							
40	28_H	DQ Calibration Pattern	W	DQ Calibration Pattern [15:8]							

	Applied when FSP = 0
	Applied when FSP = 1

Table 9: Registers Implemented in JEDEC and Micron Models

MR#	MA <5:0>	Function	Access	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
0	00_H	Device Info	R	CATR	RFU	RFU	RZQI		RFU	Latency Mode	Refresh mode
1	01_H	Device Feature 1	W	RPST0	nWR0		RD-PRE0	WR-PRE0	BL		
				RPST1	nWR1		RD-PRE1	WR-PRE1			
2	02_H	Device Feature 2	W	WR Lev	WL Select0	WL0			RL0		
					WL Select1	WL1			RL1		
3	03_H	I/O Config-1	W	DBI-WR0	DBI-RD0	PDDS0			PPR P0	WR-PST0	PU-CAL0
				DBI-WR1	DBI-RD1	PDDS1			PPR P1	WR-PST1	PU-CAL0
4	04_H	Refresh Rate	R/W	TUF	Thermal Offset		PPRE	SR Abort	Refresh Rate		
5	05_H	Basic Config-1	R	Manufacturer ID							
6	06_H	Basic Config-2	R	Revision ID-1							
7	07_H	Basic Config-3	R	Revision ID-2							
8	08_H	Basic Config-4	R	I/O width		Density				Type	
9	09_H	Test Mode	W	Vendor-Specific							
10	$0A_H$	IO Calibration	W	RFU							ZQ-RESET
11	$0B_H$	ODT Feature	W	ODTE-CA0	CA ODT0			ODTE-DQ0	DQ ODT0		
				ODTE-CA1	CA ODT1			ODTE-DQ1	DQ ODT1		
12	$0C_H$	VREF(ca) Setting/Range	R/W	RFU	VR-CA0	VREF0(ca)					
				RFU	VR-CA1	VREF1(ca)					
13	$0D_H$	CBT,VRO,DME,FSP-WR,FSP-OP	W	FSP-OP	FSP-WR	DM_DIS	RRO	VRCG	VRO	RPT	CBT
14	$0E_H$	VREF(dq) Setting/Range	R/W	RFU	VR-DQ0	VREF0(dq)					
				RFU	VR-DQ1	VREF1(dq)					
15	$0F_H$	Lower-Byte Invert for DQ Calibration	W	Lower Byte Invert Register for DQ Calibration[7:0]							
16	10_H	PASR Bank	W	PASR Bank Mask							
17	11_H	PASR Segment	W	PASR Segment Mask							
18	12_H	IT-LSB	R	DQS Oscillator Count-LSB[7:0]							
19	13_H	IT-MSB	R	DQS Oscillator Count-LSB[15:8]							
20	14_H	Upper-Byte Invert for DQ Calibration	W	DQ Calibration Invert Mask [15:8]							
22	16_H	Controller ODT	W	ODTD for x8_2ch mode		ODTD-CA	ODTE-CS	ODTE-CK	SOC ODT		
				ODTD for x8_2ch mode		ODTD-CA	ODTE-CS	ODTE-CK	SOC ODT		
23	17_H	DQS IT	W	DQS interval timer run time setting							
24	18_H	TRR, MAC	R/W	TRR Mode	TRR Mode BAn			Unlimited MAC	MAC Value		
25	19_H	PPR Resource	R	PPR Resource							
30	$1E_H$	Reserved for testing	W	Don't care							
32	20_H	DQ Calibration Pattern	W	DQ Calibration Pattern [7:0]							
39	27_H	Reserved for testing	W	Don't care							
40	28_H	DQ Calibration Pattern	W	DQ Calibration Pattern [15:8]							

	Applied when FSP = 0
	Applied when FSP = 1

Table 10 : Registers Implemented in JEDEC Byte Mode Models

MR#	MA <5:0>	Function	Access	OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0
0	00_H	Device Info	R	CATR	RFU	RFU	RZQI		RFU	Latency Mode	Refresh mode
1	01_H	Device Feature 1	W	RPST0	nWR0		RD-PRE0	WR-PRE0	BL		
				RPST1	nWR1		RD-PRE1	WR-PRE1			

2	02H	Device Feature 2	W	WR Lev	WL Select0	WL0			RL0			
					WL Select1	WL1			RL1			
3	03H	I/O Config-1	W	DBI-WR0	DBI-RD0	PDDS0			PPRP 0	WR-PST0	PU-CAL0	
				DBI-WR1	DBI-RD1	PDDS1			PPRP 1	WR-PST1	PU-CAL0	
4	04H	Refresh Rate	R/W	TUF	Thermal Offset		PPRE	SR Abort	Refresh Rate			
5	05H	Basic Config-1	R	Manufacturer ID								
6	06H	Basic Config-2	R	Revision ID-1								
7	07H	Basic Config-3	R	Revision ID-2								
8	08H	Basic Config-4	R	I/O width		Density			Type			
9	09H	Test Mode	W	Vendor-Specific								
10	0AH	IO Calibration	W	RFU								ZQ-RESET
11	0BH	ODT Feature	W	DQ ODTnt0	CA ODT0			DQ ODTnt0	DQ ODT0			
				DQ ODTnt1	CA ODT1			DQ ODTnt1	DQ ODT1			
12	0CH	VREF(ca) Setting/Range	R/W	RFU	VR-CA0	VREF0(ca)						
				RFU	VR-CA1	VREF1(ca)						
13	0DH	CBT,VRO,DME,FSP-WR,FSP-OP	W	FSP-OP	FSP-WR	DM_DIS	RRO	VRCG	VRO	RPT	CBT	
14	0EH	VREF(dq) Setting/Range	R/W	RFU	VR-DQ0	VREF0(dq)						
				RFU	VR-DQ1	VREF1(dq)						
15	0FH	Lower-Byte Invert for DQ Calibration	W	Lower Byte Invert Register for DQ Calibration[7:0]								
16	10H	PASR Bank	W	PASR Bank Mask								
17	11H	PASR Segment	W	PASR Segment Mask								
18	12H	IT-LSB	R	DQS Oscillator Count-LSB[7:0]								
19	13H	IT-MSB	R	DQS Oscillator Count-LSB[15:8]								
20	14H	Upper-Byte Invert for DQ Calibration	W	DQ Calibration Invert Mask [15:8]								
22	16H	Controller ODT	W	ODTD for x8_2ch mode		ODTD-CA	ODTE-CS	ODTE-CK	SOC ODT			
				ODTD for x8_2ch mode		ODTD-CA	ODTE-CS	ODTE-CK	SOC ODT			
23	17H	DQS IT	W	DQS interval timer run time setting								
24	18H	TRR, MAC	R/W	TRR Mode	TRR Mode BAn			Unlimited MAC	MAC Value			
25	19H	PPR Resource	R	PPR Resource								
30	1EH	Reserved for testing	W	Don't care								
32	20H	DQ Calibration Pattern	W	DQ Calibration Pattern [7:0]								
39	27H	Reserved for testing	W	Don't care								
40	28H	DQ Calibration Pattern	W	DQ Calibration Pattern [15:8]								

	Applied when FSP = 0
	Applied when FSP = 1
	Vendor Specific

10.1.MR0 Device Info Register

MR0 for Samsung and SK models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CATR	RFU	PPR	RZQI		RFU		

MR0 for JEDEC models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Reserved	RFU	RFU	RZQI	RFU	Latency Mode	Refresh Mode

MR0 for JEDEC LPDDR4X models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	RFU	RFU	RZQI		RFU		Refresh Mode

Always returns a value of 0.

10.2.MR1 Device Feature 1 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PST	nWR(for AP)			RD-PRE	WR-PRE	BL	

nWR value for Samsung and SK models

Bit 6	Bit 5	Bit 4	nWR
0	0	0	6 (default)
0	0	1	10
0	1	0	14
0	1	1	20
1	0	0	24
1	0	1	28
1	1	0	32
1	1	1	36

nWR value for JEDEC models

Bit 6	Bit 5	Bit 4	nWR
0	0	0	6 (default)
0	0	1	10
0	1	0	16
0	1	1	20
1	0	0	24
1	0	1	30
1	1	0	34
1	1	1	40

nWR value for JEDEC Byte Mode models

Bit 6	Bit 5	Bit 4	nWR
0	0	0	6 (default)
0	0	1	12
0	1	0	16
0	1	1	22
1	0	0	28
1	0	1	32
1	1	0	38
1	1	1	44

Bit 7	Post-amble Length
0	RD Post-amble = 0.5*tCK (default)
1	RD Post-amble = 1.5*tCK

Bit 3	RD Pre-amble Length
0	Pre-amble = 2*tCK Static (default)
1	Pre-amble = 2*tCK Toggle

Bit 2	WR Pre-amble Length
0	Pre-amble = 1tCK (default)
1	Pre-amble = 2*tCK

Bit 1	Bit 0	Burst Length
0	0	16 Sequential (default)
0	1	32 Sequential
1	0	16 or 32 Sequential (on-the-fly)
1	1	Reserved

10.3.MR2 Device Feature 2 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WR Lev	WL Select	WL			RL		

RL and WL settings for normal models

Read Latency		Write Latency	
No DBI	w/ DBI	Set “A”	Set “B”
6	6	4	4
10	12	6	8
14	16	8	12
20	22	10	18
24	28	12	22
28	32	14	26
32	36	16	30
36	40	18	34

RL and WL settings for JEDEC Byte Mode models

Read Latency		Write Latency	
No DBI	w/ DBI	Set “A”	Set “B”
6	6	4	4
10	12	6	8
16	18	8	12
22	24	10	18
26	30	12	22
32	36	14	26
36	40	16	30
40	44	18	34

10.4.MR3 IO Config Register

MR3 for Samsung and SK models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBI-WR	DBI-RD	PDDS			RFU		PU-CAL

MR3 for JEDEC models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBI-WR	DBI-RD	PDDS			PPRP	WR PST	PU-CAL

Only the DBI-WR and DBI-RD bit are supported in LPDDR4 model.

10.5.MR4 SDRAM Refresh Register

MR4 for Samsung and SK models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TUF	RFU		PPRE	TRR	Refresh Rate		

Only TRR function is supported in LPDDR4 model. When TRR=1, MR4[6:4] means the bank for TRR.

MR4 for JEDEC models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TUF	Thermal Offset		PPRE	SR Abort	Refresh Rate		

10.6.MR5 Basic Config 1 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Manufacturer ID							

See JEDEC specification for list of manufacturer codes. The model currently supports Samsung (0x01) and Micron (0xFF).

10.7.MR6 Basic Config 2 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Revision ID 1							

Always returns a value of 0.

10.8.MR7 Basic Config 3 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Revision ID 2							

Always returns a value of 0.

10.9.MR8 Basic Config 4 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IO Width		Density				Type	

Type for LPDDR4:

Bit 1	Bit 0	Type
0	0	S16 SDRAM (16n pre. fetch)
0	1	Reserved
1	0	Reserved
1	1	Reserved

Type for LPDDR4X:

Bit 1	Bit 0	Type
0	0	S16 SDRAM (16n pre-fetch), STD VDDQ(1.1V)
0	1	S16 SDRAM (16n pre-fetch), Low VDDQ(0.6/0.4V) (default)
1	0	Reserved
1	1	Reserved

Bit 5	Bit 4	Bit 3	Bit 2	Density
0	0	0	0	4Gb per die (2Gb per channel)
0	0	0	1	6Gb per die (3Gb per channel)
0	0	1	0	8Gb per die (4Gb per channel)
0	0	1	1	12Gb per die (6Gb per channel)
0	1	0	0	16Gb per die (8Gb per channel)
0	1	0	1	24Gb per die (12Gb per channel)
0	1	1	0	32Gb per die (16Gb per channel)
All others				Not supported

Bit 7	Bit 6	IO Width
0	0	X16 (per channel)
0	1	X8 (per channel)
1	0	Reserved
1	1	Reserved

10.10. MR9 Test Mode Register

This register is implemented in the model but its content has no effect on the function of the model.

10.11. MR10 Calibration Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU							ZQ-Reset

ZQ Reset is not supported in LPDDR4 model.

10.12. MR11 ODT Feature Register

This register is implemented in the model but its content has no effect on the function of the model.

10.13. MR12 VREF(ca) Registers

This register is implemented in the model but its content has no effect on the function of the model.

10.14. MR13 Register Information

MR13 for Samsung and SK models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSP-OP	FSP-WR	DM_DIS	RFU	VRCG	VRO	RFU	CBT

MR13 for JEDEC models

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSP-OP	FSP-WR	DM_DIS	RRO	VRCG	VRO	RPT	CBT

Only FSP-OP, FSP-WR, DM_DIS and CBT are supported in LPDDR4 model.
There are two physical registers for MR1, MR2, MR3, MR11, MR12, MR14 and MR22, designated set point 0 and set point 1. The FSP-WR bit controls which register to be read and written. The FSP-OP bit controls which register to be used for command operation.

10.15. MR14 VREF(dq) Registers

This register is implemented in the model but its content has no effect on the function of the model.

10.16. MR15 Lower-Byte Invert Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Lower-Byte Invert Register for DQ Calibration(default 55_H)							

10.17. MR16/MR17 PASR Bank and PASR Segment Registers

These register are implemented in the model but their contents have no effect on the function of the model.

10.18. MR18/MR19 IT-LSB and IT_MSB Registers

If the macro “MMP_DQSOSC” is not defined, MR18 and MR19 are set to zero.

When the macro “MMP_DQSOSC” is defined, the DQS Internal Oscillator is enabled in the LPDDR5 model. MR18 reports the LSB bits of the DRAM DQS Oscillator count. MR19 reports the MSB bits of the DRAM DQS Oscillator count.

10.19. MR20 Upper-Byte Invert Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Upper-Byte Invert Register for DQ Calibration(default 55_H)							

10.20. MR22 ODT Feature Register

This register is implemented in the model but its content has no effect on the function of the model.

10.21. MR23 DQS interval timer Register (JEDEC models only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DQS interval timer run time setting							

When MR23 is OP[7:0]=8'h00, MPC command with OP[6:0]=1001101B (Stop DQS Interval Oscillator) stops DQS interval timer works. When MR23 is non-zero, this command is considered as illegal. In this case, DQS interval timer stops automatically at MR23*16th CK clock cycle after timer start. The macro of "MMP_DQSOSC" shall be defined to enable this function.

10.22. MR24 TRR and MAC Register (JEDEC models only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRR Mode	TRR Mode BAn			Unlimited MAC	MAC Value		

The MAC function is not supported in LPDDR4 model.

10.23. MR25 PPR Resource Register (JEDEC models only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PPR Resource							

This register is implemented in the model but its content has no effect on the function of the model.

10.24. MR32 DQ Calibration Pattern A Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DQ Calibration Pattern[7:0] (default is $5A_H$)							

10.25. MR40 DQ Calibration Pattern B Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DQ Calibration Pattern[15:8] (default is $3C_H$)							

11. Commands

The LPDDR4 model accepts the following commands:

- Deselect
- MPC
- Mode Register Write (MRW-1 following MRW-2)
- Mode Register Read (MRR-1 following CAS-2)
- Activate (ACT-1 following ACT-2)
- Precharge (Per Bank, All Bank)

- Refresh (Per Bank, All Bank)
- Read (RD-1 following CAS-2)
- Read with AP (RD-1 following CAS-2)
- Write (WR-1 following CAS-2)
- Write with AP (WR-1 following CAS-2)
- Mask Write (MWR-1 following CAS-2)
- Self Refresh Entry
- Self Refresh Exit

The following table shows the command encoding for LPDDR4.

Table 11: LPDDR4 Command Encoding

SDRAM Command	SDR Command Pins			SDR CA Pins (6)						
	CKE		CS	CA0	CA1	CA2	CA3	CA4	CA5	CK_t Edge
	CK_t(n-1)	CK_t(n)								
Maintain PD	L	L	X	X						R1
Enter Power Down (PDE)	H	L	L	V						R1
	L		L	V						R2
Exit Power Down	L	H	L	V						R1
	H		L	V						R2
Deselect (DES)	H	H	L	X						R1
Multi Purpose Command (MPC)	H	H	H	L	L	L	L	L	OP6	R1
			L	OP0	OP1	OP2	OP3	OP4	OP5	R2
Precharge (PRE)	H	H	H	L	L	L	L	H	AB	R1
			L	BA0	BA1	BA2	V	V	V	R2
Refresh (REF)	H	H	H	L	L	L	H	L	AB	R1
			L	BA0	BA1	BA2	V	V	V	R2
Self Refresh Entry(SRE)	H	H	H	L	L	L	H	H	V	R1
			L	V						R2
Write-1 (WR-1)	H	H	H	L	L	H	L	L	BL	R1
			L	BA0	BA1	BA2	V	C9	AP	R2
Self Refresh Exit(SRX)	H	H	H	L	L	H	L	H	V	R1
			L	V						R2
Masked Write-1 (MWR-1)	H	H	H	L	L	H	H	L	BL	R1
			L	BA0	BA1	BA2	V	C9	AP	R2
RFU	H	H	H	L	L	H	H	H	V	R1
			L	V						R2
Read-1 (RD-1)	H	H	H	L	H	L	L	L	BL	R1
			L	BA0	BA1	BA2	V	C9	AP	R2
CAS-2	H	H	H	L	H	L	L	H	C8	R1
			L	C2	C3	C4	C5	C6	C7	R2
RFU	H	H	H	L	H	L	H	L	V	R1
			L	V						R2
RFU	H	H	H	L	H	L	H	H	V	R1
			L	V						R2
Mode Register Write 1 (MWR-1)	H	H	H	L	H	H	L	L	OP7	R1
			L	MA0	MA1	MA2	MA3	MA4	MA5	R2
Mode Register Write 2 (MWR-2)	H	H	H	L	H	H	L	H	OP6	R1
			L	OP0	OP1	OP2	OP3	OP4	OP5	R2
Mode Register Write 1 (MWR-1)	H	H	H	L	H	H	H	L	V	R1
			L	MA0	MA1	MA2	MA3	MA4	MA5	R2
RFU	H	H	H	L	H	H	H	L	V	R1
			L	V						R2
Activate-1 (ACT-1)	H	H	H	H	L	R12	R13	R14	R15	R1
			L	BA0	BA1	BA2	R16	R10	R11	R2
Activate-2 (ACT-2)	H	H	H	H	H	R6	R7	R8	R9	R1
			L	R0	R1	R2	R3	R4	R5	R2

MPC sub-commands:

Table 12: MPC Sub-Commands

Function	Operand	Data
Training Modes	OP[4:0]	XXXX0B: NOP 00001B: Read DQ FIFO 00011B: Read DQ Training 00101B: RFU 00111B: Write DQ FIFO 01001B: RFU 01011B: Start Interval Timer 01101B: Stop Interval Timer 01111B: ZQCal Start 10001B: ZQCal Latch 10011B-11111B: RFU
Reserved	OP[6:5]	XXB: RFU

12. DBI and DM function

The LPDDR4 Model supports Data Mask function for Write operation and Data Bus Inversion (DBIdc) function for Write and Read operation. LPDDR4 supports DM and DBIdc function with a byte granularity. MR3 OP7 and OP6 define DBI-WR and DBI-RD enable bits, and MR13 OP5 defines DM-DIS bit. There are eight possible combinations for LPDDR4 model with DM and DBIdc function. Mask Write command only supports BL16. MRR support DBI function as normal read operation.

The below table describes the functional behavior for all combinations.

Table 13: Functional Behavior for LPDDR4 DM and DBIdc

DM function	Write DBIdc Function	Read DBIdc Function	DMI Signal during Write Command	DMI Signal during Masked Write Command	DMI Signal During Read Command
Disable	Disable	Disable	Note:1	Note:1,3	Note:2
Disable	Enable	Disable	Note:4	Note:3	Note:2
Disable	Disable	Enable	Note:1	Note:3	Note:5
Disable	Enable	Enable	Note:4	Note:3	Note:5
Enable	Disable	Disable	Note:6	Note:7	Note:2
Enable	Enable	Disable	Note:4	Note:8	Note:2
Enable	Disable	Enable	Note:6	Note:7	Note:5
Enable	Enable	Enable	Note:4	Note:8	Note:5

Notes:
1) DMI input signal is a don't care. DMI input receivers are turned OFF.
2) DMI output drivers are turned OFF.
3) Masked Write Command is not allowed and is considered an illegal command as DM function is disabled.
4) DMI signal is treated as DBI signal and it indicates whether DRAM needs to invert the Write data received on DQs within a byte. The LPDDR4 device inverts Write data received on the DQ inputs in case DMI was sampled HIGH, or leaves the Write data non-inverted in case DMI was sampled LOW.
5) The LPDDR4 DRAM inverts Read data on its DQ outputs associated within a byte and drives DMI signal HIGH when the number of ' 1' data bits within a given byte lane is greater than four; otherwise the DRAM does not invert the read data and drives DMI signal LOW.
6) The LPDDR4 DRAM does not perform any mask operation when it receives Write command. During the Write burst associated with Write command, DMI signal is a don't care and ignored by DRAM.
7) The LPDDR4 DRAM requires an explicit Masked Write command for all masked write operations. DMI signal is treated as DM signal and it indicates which bit time within the burst is to be masked. When DMI signal is HIGH, DRAM masks that bit time across all DQs associated within a byte. All DQ input signals within a byte are don't

care (either HIGH or LOW) when DMI signal is HIGH. When DMI signal is LOW, the LPDDR4 DRAM does not perform mask operation and data received on DQ input is written to the array.
8) The LPDDR4 DRAM requires an explicit Masked Write command for all masked write operations. The LPDDR4 device masks the Write data received on the DQ inputs if the total count of '1' data bits on DQ[2:7] or DQ[10:15] (for Lower Byte or Upper Byte respectively) is equal to or greater than five, and DMI signal is LOW. Otherwise the LPDDR4 DRAM does not perform mask operation and treats it as a legal DBI pattern; DMI signal is treated as DBI signal and data received on DQ input is written to the array.

13. DQ Training function

The Read DQ FIFO and Write DQ FIFO use the MPC command with operands to enable this special mode of operation. Up to 5 consecutive Write DQ FIFO commands with user defined patterns may be issued to the SDRAM to store up to 80 values (BL16*5) per pin that can be read back via the Read DQ FIFO command.
Read DQ FIFO is non-destructive to the data captured in the FIFO, so data may be read continuously until it is disturbed by another command to the SDRAM. The Read DQ FIFO pointer is reset internally to FIFO[0] in the DRAM anytime a command other than Read DQ FIFO or CAS-2 is received. The Write DQ FIFO pointer is reset internally to FIFO[0] only when SRE command or reset issued.
Read DQ FIFO and Write DQ FIFO will use DBI function as normal read and write command.
Read DQ Training command will not use DBI function.

14. DQS Interval Oscillator

The LPDDR4 model supports the DQS Interval Oscillator feature. This feature is optional and can be enabled by defining the Verilog macro "MMP_DQSOSC" during the compile phase. It is expected that using the DQS Interval Oscillator feature will slow down the emulation if it is modeled accurately.

The DQS Interval Oscillator is started by issuing a MPC [Start Interval Timer] command with OP[7:0] set as described in the Table 12: MPC Sub-Commands table.

The DQS Interval Oscillator may be stopped by issuing a MPC [Stop Interval Timer] command with OP[7:0] set as described in the Table 12: MPC Sub-Commands table, or (JEDEC models only) the controller may instruct the SDRAM to count for a specific number of clocks and then stop automatically (See MR23 for more information). If MR23 is set to automatically stop the DQS Oscillator, then MPC command should not be used(illegal).

When the DQS Oscillator is stopped by either method, the result of the oscillator counter is automatically stored in MR18 & MR19.

The LPDDR4 model requires the user to provide an external dedicated independent clock to drive the DQS Interval counter inside the LPDDR4 model through the port of DQSOSC_CLK. The valid frequency should be within the range of 625MHz (= 1/(2*tDQS2DQmax)=1/(2*800ps)) to 2500MHz (= 1/(2*tDQS2DQmin)=1/(2*200ps)).

15. Enhanced Timing Parameter Accuracy

15.1. Enhanced Timing Parameter Accuracy for TDQSCK/TDQSS/TDQS2DQ

The LPDDR4 model supports in a very limited manner some enhanced timing parameter accuracy. The enhanced timing parameter accuracy affects TDQSCK, TDQSS, and TDQS2DQ. The unit is one half-cycle. The user is responsible for setting the model's parameters tDQSCK and tDQSCK_half, tDQSS and tDQSS_half, tDQS2DQ and tDQS2DQ_half for the three timing parameters.

The timing for TDQSCK and TDQSS function is enabled by default. On the other hand, the timing for TDQS2DQ function needs to be enabled by defining the Verilog macro "MMP_ENH_TIMING_ACCURACY". Below are some considerations related to the TDQS2DQ function:

- There will also be an extra 2x clock input needed if the user wants to use the timing parameter TDQS2DQ. This 2x clock is used to delay the input DQS bus and then to latch the DQ bus data.
- The user should note that if the timing TDQS2DQ function is enabled, and thus using the extra input 2x clock, the model performance in Palladium will be decreased. The other two timing parameters, TDQSCK and TDQSS, will not affect model performance.
- There is a limitation when using the timing parameter TDQS2DQ. The 2x clock is used to delay the DQS, but the phase between 2x clock and DQS is not fixed, so the delayed DQS may not latch the DQ correctly in the below CASE02. In order to latch the DQ correctly, the user needs to add another half cycle of CK to the value of TDQS2DQ and set the model parameter tDQS2DQ and tDQS2DQ_half accordingly.

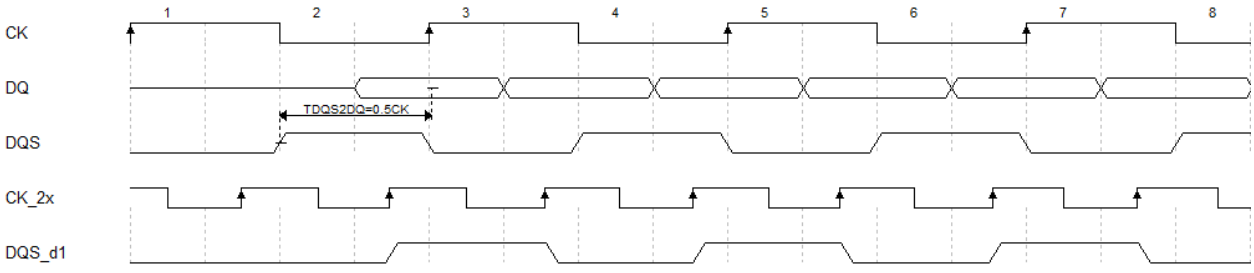


Figure 5: Timing for Latching DQ Using Delayed DQS (CASE01)

As shown in the figure above, the TDQS2DQ is 0.5CK, so the model will use DQS_d1(delayed DQS one cycle of CK_2x) to latch the DQ bus according to the user configured model parameter tDQS2DQ=0 and tDQS2DQ_half=1.

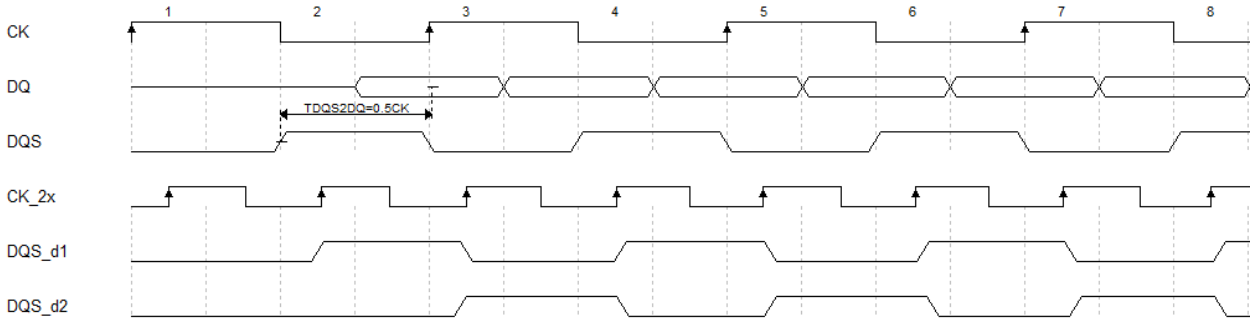


Figure 6: Timing for Latching DQ Using Delayed DQS (CASE02)

As shown in the figure above, the TDQS2DQ is 0.5CK, so the model will use DQS_d1 (delayed DQS one cycle of CK_2x) to latch the DQ bus when user configured tDQS2DQ=0 and tDQS2DQ_half=1. As shown, with the phase relationship between CK_2x and DQS, the DQS_d1 can't latch the DQ bus correctly; however the DQS_d2 (delayed DQS two cycle of CK_2x) can latch the DQ bus correctly. So in this case, although the TDQSCK is 0.5CK, the user should configure the model parameter tDQS2DQ=1 and tDQS2DQ_half=0. Then the model will use the DQS_d2 to latch the DQ bus.

15.2. Runtime Control of Timing Parameters

The following mechanism allows users to change the timing parameters TDQSCK/TDQSS/TDQS2DQ at runtime.

Parameter name	Register name for runtime control
tDQSCK	tDQSCK_reg
tDQSCK_half	tDQSCK_half_reg
tDQSS	tDQSS_reg
tDQSS_half	tDQSS_half_reg
tDQS2DQ	tDQS2DQ_reg
tDQS2DQ_half	tDQS2DQ_half_reg

Step1: Users must add "keepNet" during compilation for all the registers listed below.

```
keepNet -add {tb.inst0.lpddr4_chA.tDQSCK_reg }
keepNet -add {tb.inst0.lpddr4_chB.tDQSCK_reg }
keepNet -add {tb.inst0.lpddr4_chA.tDQSCK_half_reg }
keepNet -add {tb.inst0.lpddr4_chB.tDQSCK_half_reg }
keepNet -add {tb.inst0.lpddr4_chA.tDQSS_reg }
keepNet -add {tb.inst0.lpddr4_chB.tDQSS_reg }
keepNet -add {tb.inst0.lpddr4_chA.tDQSS_half_reg }
keepNet -add {tb.inst0.lpddr4_chB.tDQSS_half_reg }
keepNet -add {tb.inst0.lpddr4_chA.tDQS2DQ_reg }
keepNet -add {tb.inst0.lpddr4_chB.tDQS2DQ_reg }
keepNet -add {tb.inst0.lpddr4_chA.tDQS2DQ_half_reg }
keepNet -add {tb.inst0.lpddr4_chB.tDQS2DQ_half_reg }
```

Step2: After download and during runtime, use the 'force' command to change the values of timing parameter.

```
force tb.inst0.lpddr4_chA.tDQSCK_reg 2
force tb.inst0.lpddr4_chB.tDQSCK_reg 2
force tb.inst0.lpddr4_chA.tDQSCK_half_reg 1
force tb.inst0.lpddr4_chB.tDQSCK_half_reg 1
force tb.inst0.lpddr4_chA.tDQSS_reg 1
force tb.inst0.lpddr4_chB.tDQSS_reg 1
force tb.inst0.lpddr4_chA.tDQSS_half_reg 0
force tb.inst0.lpddr4_chB.tDQSS_half_reg 0
force tb.inst0.lpddr4_chA.tDQS2DQ_reg 0
force tb.inst0.lpddr4_chB.tDQS2DQ_reg 0
force tb.inst0.lpddr4_chA.tDQS2DQ_half_reg 0
force tb.inst0.lpddr4_chB.tDQS2DQ_half_reg 0
```

16. Initialization Sequence

The LPDDR4 model requires that the memory controller follows the initialization sequence as documented in the specification. The sequence basically entails the following steps:

1. RESET_n asserted.
2. RESET_n de-asserted and then CKE goes high.
3. At least one DES command after CKE goes active.
4. (Optional)Issue MRR or MRW to set some parameters.
5. (Optional)ZQ Calibration: issue ZQCal start command and then ZQCal Latch command to finish ZQ Calibration.
6. (Optional)Command Bus Training: issue MRW to enable CBT and enter CBT mode, and then do command bus training, and then issue MRW to disable CBT and exit CBT mode.
7. Write Leveling: issue MRW to enable write leveling and then do write leveling, and then issue MRW to disable write leveling and initial done.

Initialization sequence state-machine Init_STATE signal decoding in the model:

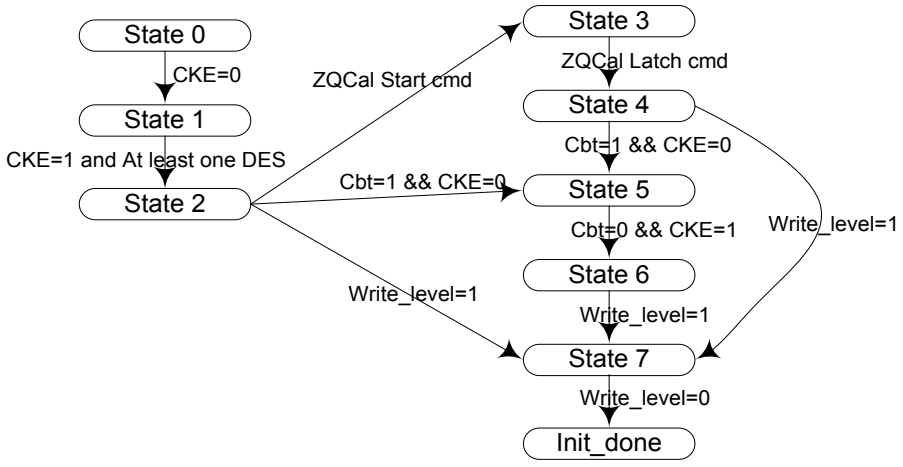


Figure 7: Initialization Sequence

Notes:
0. Init_STATE = 0: initial state, Reset_n de-asserted.
1. Init_STATE= 1: wait CKE asserted; and at least one DES command issued.
2. Init_STATE = 2: wait MRW or MRR command to set some parameters or write to MR13 OP[0] to enable CBT or write to MR2 OP[7] to enable write leveling.
3. Init_STATE = 3: process ZQ Calibration operation, and wait ZQCal latch command to finished ZQ Calibration.
4. Init_STATE = 4: wait MRW command to enable CBT. If cbt=1 and CKE=0, enter command bus training mode.
5. Init_STATE = 5: process Command bus training operation, wait MRW command to disable CBT; if cbt=0 and CKE=1, exit command bus training mode.
6. Init_STATE = 6: wait MRW command to enable write leveling.
7. Init_STATE = 7; process Write Level operation, wait MRW command to disable write leveling; if write_level = 0, exit write leveling mode and finished initialization. The signal init_done will be set to 1.

The model requires that these steps are performed in the correct sequence in order to complete initialization. The model will not respond to any others commands unless this sequence is completed.

17. Compile and Emulation

The model is provided as a protected RTL file(s) (*.vp). The file(s) need to be synthesized prior to the back-end Palladium compile. An example of the command for compilation (including synthesis) and run of this model in the IXCOM flow is shown below.

```
ixcom -64 +sv -ua +dut+jedec_lpddr4_4Gb \
    ./lpddr4_pd.vp \
    ./jedec_lpddr4_4Gb.v \
    -incdir ../../../utils/cdn_mmp_utils/sv \
    ../../../utils/cdn_mmp_utils/sv/cdn_mmp_utils.sv \
    ……

xeDebug -64 --ncsim \
    -sv_lib ../../../utils/cdn_mmp_utils/lib/64bit/libMMP_utils.so -- \
    -input auto_xedebug.tcl
```

The script below shows two examples for Palladium classic ICE synthesis:

```
1)
hdlInputFile lpddr4_pd.vp
hdlInputFile jedec_lpddr4_4Gb.v
hdlImport -full -2001 -l qtref
hdlOutputFile -add -f verilog jedec_lpddr4_4Gb.vg
hdlSynthesize -memory -keepVhdlCase -keepRtlSymbol -keepAllFlipFlop
jedec_lpddr4_4Gb
……

2)
vavlog lpddr4_pd.vp jedec_lpddr4_4Gb.v

vaelab -keepRtlSymbol -keepAllFlipFlop -outputVlog jedec_lpddr4_4Gb.vg
jedec_lpddr4_4Gb
……
```

For multi-channel LPDDR4 models, the wrapper (<model_name>.v) and core file (lpddr4_pd*.vp) are needed for compile.
For single-channel LPDDR4 models, only the file jedec_lpddr4_single_ch_*Gb.vp is needed for compile.

NOTE: It is common for Palladium flows to require –keepallFlipFlop since it removes optimizations that are in place by default. For example, without –keepAllFlipFlop, HDL-ICE can remove flops with constant inputs and merge equivalent FF. The picture above is modified a bit when ICE ATB mode (–atb) is used since then a constant input FF is only optimized out when there is no initial value for it or the initial value is the same as the constant input value.

It is also common for Palladium flows to require –keepRtlSymbol. This option enables the HDL Compiler to keep original VHDL RTL symbols, such as ".", whenever possible. In other words, it maps VHDL RTL signal name a.b to the netlist entry, \a.b. Without this modifier, the signal name would otherwise be converted to a_b in the netlist.

If the recommended compile script includes the aforementioned options, the user must include them to avoid affecting functionality of the design.

18. SWI Smart Memory Interface

This LPDDR4 core model (lpddr4_pd.vp) supports a fixed, Verilog macro controlled interface, or "hooks", to one of the Smart Memory components in Cadence's Software Integrator (SWI) product. Software Integrator (SWI) is a support library that is part of Cadence's Hybrid Solution--a multi-tool system level solution that runs in a hybrid PXP + IES simulation mode and targets the increasing number of SoC performance conscious projects requiring the booting of commercial operating systems and the running of complex software-driven tests against an accurate model of the SoC prior to tapeout.

For additional details about the SWI product at large please consult the SWI product documentation which includes a user guide. This documentation can be accessed via support.cadence.com and is located on the Product Pages/Product Manuals link where SWI 13.1 is located with other Functional Verification products.

The user of the SWI solution who is integrating this core model to the corresponding SWI Smart Memory component side should define the Verilog macro "MMP_SM" to enable the Smart Memory interface. This will enable the portion of the interface that resides in the MMP model core, thus completing access to the implemented SWI Smart Memory functionality.

Table 14 : SWI Smart Memory Verilog Define

`define Macro purpose	Possible `define Values
Set the Smart Memory interface to compile "in" as part of the memory model	MMP_SM

The SWI Smart Memory interface includes the signals shown in the table below. It is outside the MMP scope to treat the integration of the MMP model into a hybrid solution. For additional details, please consult the SWI documentation and other Hybrid Solution documentation.

Table 15: SWI Smart Memory Interface Signals

NAME	DECLARATION	DESCRIPTION
sm_raddr	output [(total_addr_bits-1):0] sm_raddr	Smart Memory read address
sm_re	output sm_re	Smart Memory read enable
sm_raddr_en	output sm_raddr_en	Smart Memory read address enable
sm_dout	input [data_bits-1:0] sm_dout	Smart Memory read data
sm_waddr	output [(total_addr_bits-1):0] sm_waddr	Smart Memory write address
sm_we	output sm_we	Smart Memory write enable
sm_waddr_en	output sm_waddr_en	Smart Memory write address enable
sm_din	output [data_bits-1:0] sm_din	Smart Memory write data
sm_bw	output [data_bits-1:0] sm_bw	Smart Memory data mask
verbosity	input [3:0] verbosity	Smart Memory debug info display level control

The Smart Memory interface includes a user adjustable parameter that passes user data from wrapper layers that are external to this MMP model into MMP model. The single 64-bit parameter is subdivided by field to accommodate data as shown in the table below. This parameter defaults to a value of '0' and is managed by the SWI product Smart Memory component. For additional details, please consult the SWI documentation and other Hybrid Solution documentation.

Table 16: SWI Smart Memory User Adjustable Parameters

PARAMETER	DESCRIPTION
parameter [63:0] SMConfigSpecific_UserData	Smart Memory User Data Passing
FIELD	**DESCRIPTION**
[2:0]	Used for specifying device/channel/subpart
[63:61]	Extension value of total_addr_bits

The Smart Memory interface includes non-user adjustable localparams and parameters as shown in the table below. Note that a localparam of the same name (total_addr_bits) but of different size is part of the standard MMP core model.

Table 17: SWI Smart Memory Non-User Adjustable Parameters

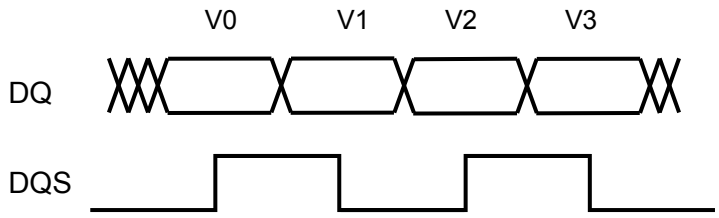
LOCALPARAM	VALUE	DESCRIPTION
total_addr_bits	bank_addr_width+row_addr_width +col_addr_width + SMConfigSpecific_UserData[63:61]	Memory capacity width

The SWI Smart Memory interface has dependencies on the inclusion of external file(s). For additional details about purpose and content of any Smart Memory related external files, please consult the SWI documentation and other Hybrid Solution documentation.

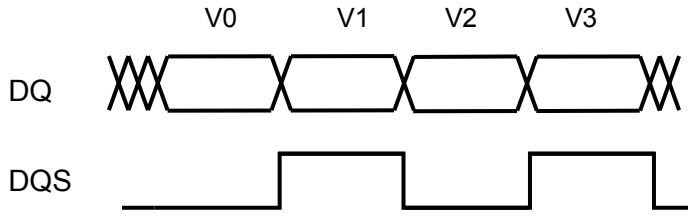
```
`ifdef MMP_SM
   `include "cdn_sm_mapDRBCToLinAdr.vh"
`endif
```

19. Handling DQS in Palladium Memory Models

For writes to a DDR memory, industry datasheets show each DQS edge centered within the corresponding valid period (v0, v1, v2, etc.) of DQ, as in the following diagram.

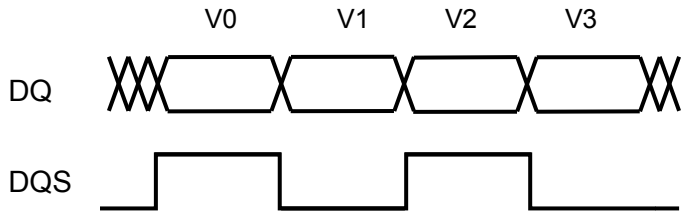


For DDR models provided by Cadence for Palladium, if the design drives DQ and DQS signals with the above timing, the DDR memory will behave correctly. However, to obtain this timing in Palladium, the fastest design clock must toggle twice as frequently as the DQS signal. If this faster clock is not needed for any other reason, the presence of the faster clock will usually cause an unnecessary 2X slowdown in emulation speed. To eliminate the need for a faster clock, you can have the design generate each DQS edge at the end of the corresponding DQ valid period (rather than the middle), as in the following diagram:



Note that the first DQS edge is at the *end* of first valid DQ, not at the beginning.

For reads from the DDR model, the DDR model will drive DQ and DQS with the first DQS edge at the *beginning* of the first valid data, not at the end:



The DDR model behaves this way to conform to industry datasheets for DDR memories. The design reading the data from the DDR model must delay the DQS signal, and use the delayed-DQS signal to sample the DQ. A delay of one Q_FDP0B should work fine, even in CAKE 1X mode. If you are using CAKE 1X mode and the DDR clock is the fastest design clock, the DQ signal will change twice per FCLK, and the Q_FDP0B delaying DQS will provide one-half FCLK delay, so that each delayed-DQS edge is at the end of the corresponding data valid period.

To delay the DQS signal, a commonly used approach is to create a special pad cell for DQS that has a Q_FDP0B delay cell inserted on the path that leads from the DDR memory into the design.

The user may insert delays into pad cells (or elsewhere in the design) using the below code example which leverages ixc_pulse, an internal primitive that can be used to access FCLK and to create controlled delay, for IXCOM flow and leverages the Q_FDP0B primitive for delay generation in the Classic ICE flow. For more detailed information about ixc_pulse please reference the *UXE User Guide* section called *Generating Pulses*. There is no need for the user to define IXCOM_UXE for the Verilog macro; it is predefined for the user in IXCOM flow. Note that in UXE 13.1.0 and prior the equivalent pulse generating function was named axis_pulse.

```
// Flow independent delay cell
module pxp_fclk_delay (in, out_delay);
input in;
output out_delay;

reg out_delay;

`ifdef IXCOM_UXE
  wire VCC=1'b1;
  ixc_pulse #(1)(Fclk,VCC);
  always @(posedge Fclk)
    out_delay <= in;
`else
  Q_FDP0B fclk_dly (.D(in), .Q(out_delay));
`endif

endmodule
```

20. Debugging

This model has several debugging options, techniques and tips that may assist the user may use in isolating a problem.

- For issues that are may not be model specific please review the *Memory Model Portfolio FAQ for All Models User Guide.*

- **Debug signals:**

  The following signals can be monitored to examine command sequence:

  - init_done: assert when initialization sequence finished
  - mwe: write enable signal
  - maddr: read/write burst address of the core memory
  - mdin: write data
  - dout_window: read data output enable signal when tDQSCK_half_reg=0
  - dout_window_1: read data output enable signal when tDQSCK_half_reg=1
  - mdout_dbi: read data output

  The value of below four signals is valid when wr_cmd=1 or rd_cmd=1

  - BA_pre_R2: read/write command bank address of the core memory
  - bank<bank_index>_addr_reg : read/write command row address for a specific bank of the core memory
  - col_addr_rd: read command column address of the core memory
  - col_addr_wr: write command column address of the core memory

- **Golden waveform:** A package with a reference waveform is available which shows the following command sequence(s):

  (1) Initialization sequence:
  RESET_n asserted --> RESET_n de-asserted and then CKE goes high
  --> At least one DES cmd after CKE goes active --> init_done
  (2) Following commands included:
  MRW
  MRR
  ACT
  WRITE
  READ
  PRECHARGE
  MWR

- **Debug Display:** This MMP memory model has available a built-in debug methodology called MMP Debug Display that is based on the Verilog system task $display. Please see the *Palladium Memory Model Debug Display User Guide* in the release docs directory for additional information.

- **Manual Configuring of this MMP Model Family**

This MMP model supports manual configuration by accompanying the model mode register or configuration register declarations with synthesis directives, such as keep_net directives, that instruct the compiler to ensure that the relevant nets remain available for runtime forcing. For a general description of this support please see the user guide in the MMP release with path and filename *docs/MMP_FAQ_for_All_Models.pdf*.

While MMP strongly recommends following protocol-based commands to configure MMP models, MMP recognizes that the design test environment may desire to trade off the risks inherent in streamlining or circumventing the initialization sequence part of the protocol in order to better support some testing environments.

The following table lists the internal register path and naming along with the specification or datasheet naming for model mode registers or configuration registers that are accompanied by keep_net synthesis directives in support of such manual configuration. ONLY writeable configuration registers or fields are supported thusly. Please read the relevant datasheet for details about individual register behavior and mapping to fields.

Table 18: Writeable Mode Register / Configuration Register Info

Hierarchical RTL Naming for Writeable Configuration Related Registers & Signals	Specification or Vendor Datasheet Naming for Configuration Related Registers	Access
<model_name>.MR1_reg0	MR1 (set point 0)	W
<model_name>.MR1_0_set		
<model_name>.MR1_reg1	MR1 (set point 1)	W
<model_name>.MR1_1_set		
<model_name>.MR2_reg0	MR2 (set point 0)	W
<model_name>.MR2_0_set		
<model_name>.MR2_reg1	MR2 (set point 1)	W
<model_name>.MR2_1_set		
<model_name>.MR3_reg0	MR3 (set point 0)	W
<model_name>.MR3_0_set		
<model_name>.MR3_reg1	MR2 (set point 1)	W
<model_name>.MR3_1_set		
<model_name>.MR4_reg	MR4	R/W
<model_name>.MR4_set		
<model_name>.MR9_reg	MR9	W
<model_name>.MR9_set		
<model_name>.MR10_reg	MR10	W
<model_name>.MR10_set		
<model_name>.MR11_reg0	MR11 (set point 0)	W
<model_name>.MR11_0_set		
<model_name>.MR11_reg1	MR11 (set point 1)	W
<model_name>.MR11_1_set		
<model_name>.MR12_reg0	MR12 (set point 0)	R/W
<model_name>.MR12_0_set		
<model_name>.MR12_reg1	MR12 (set point 1)	R/W
<model_name>.MR12_1_set		

<model_name>.MR13_reg	MR13	W
<model_name>.MR13_set		
<model_name>.MR14_reg0	MR14 (set point 0)	R/W
<model_name>.MR14_0_set		
<model_name>.MR14_reg1	MR14 (set point 1)	R/W
<model_name>.MR14_1_set		
<model_name>.MR15_reg	MR15	W
<model_name>.MR15_set		
<model_name>.MR16_reg	MR16	W
<model_name>.MR16_set		
<model_name>.MR17_reg	MR17	W
<model_name>.MR17_set		
<model_name>.MR20_reg	MR20	W
<model_name>.MR20_set		
<model_name>.MR22_reg0	MR22 (set point 0)	W
<model_name>.MR22_0_set		
<model_name>.MR22_reg1	MR22 (set point 1)	W
<model_name>.MR22_1_set		
<model_name>.MR23_reg	MR23	W
<model_name>.MR23_set		
<model_name>.MR24_reg	MR24	R/W
<model_name>.MR24_set		
<model_name>.MR30_reg	MR30	W
<model_name>.MR30_set		
<model_name>.MR32_reg	MR32	W
<model_name>.MR32_set		
<model_name>.MR39_reg	MR39	W
<model_name>.MR39_set		
<model_name>.MR40_reg	MR40	W
<model_name>.MR40_set		
<model_name>.init_done	[Not Applicable]	1'b1 indicates initialization is complete
<model_name>.tDQSCK_reg	The register of parameter tDQSCK	
<model_name>.tDQSCK_half_reg	The register of parameter tDQSCK_half	
<model_name>.tDQSS_reg	The register of parameter tDQSS	
<model_name>.tDQSS_half_reg	The register of parameter tDQSS_half	
<model_name>.tDQS2DQ_reg	The register of parameter tDQS2DQ	
<model_name>.tDQS2DQ_half_reg	The register of parameter tDQS2DQ_half	

Note: The user needs to force the MR#_set to be 1'b1 when forcing MR#_reg.

2. Revision History

The following table shows the revision history for this document.

Date	Version	Revision
August 2013	1.0	Initial release
August 2013	1.1	Update for Samsung lpddr4 Rev0.4
August 2013	1.2	Simplify initialization sequence
November 2013	1.3	Add debug display info
January 2014	1.4	Add MRR operation DBI supported
April 2014	1.5	Update debug display info for runtime use
May 2014	1.6	Update RBC address mapping, configuration parameters
June 2014	1.7	Adding 4-channel model description
September 2014	1.8	Remove version from UG file name. Update UXE / IXE documentation reference titles. Remove Beta watermark; LPDDR4 to mainstream release level.
September 2014	1.9	Removed debug display section. Added model parameter descriptions section. Added delay cell info.
November 2014	1.10	Remove emulation capacity info. Corrected revision history numbering—1.37 -> 1.10 to not skip rev numbers.
March 2015	1.11	Update related publications list. Add section on SWI Smart Memory interface
June 2015	1.12	Adding LPDDR4 JEDEC models description
July 2015	1.13	Update Cadence naming on front page
September 2015	1.14	Adding feature table and compile section
January 2016	2.0	Update for Palladium-Z1 and VXE Adding supported timing TDQSCK/TDQSS/TDQS2DQ description
March 2016	2.1	Add LPDDR4X support
April 2016	2.2	Update SM interface sm_we
April 2016	2.3	Adding 24Gb/32Gb size support
July 2016	2.4	Remove hyphen in Palladium naming
December 2016	2.5	Add “Manual Configuring of DDRx/LPDDRx” section
January 2017	2.6	Add some suggested signals to examine for Debug
March 2017	2.7	Add LPDDR4 Byte Mode description
April 2017	2.8	Update for new SM interface
May 2017	2.9	Update “Manual Configuring of DDRx/LPDDRx” section
September 2017	2.10	Add MMP_RBC address calculation formula
September 2017	2.11	Add interval DQS oscillator feature
October 2017	2.12	Modify “bank” vs “ba” references in Address Mapping section.
November 2017	2.13	Update for JESD209-4B; move LPDDR4 x8 from BETA to non-BETA and remove BETA notation Add Micron LPDDR4 models

Date	Version	Revision
January 2018	3.0	Modify header and footer
May 2018	3.1	Update the instance name in forcing example Add section for Manual configuration
July 2018	3.2	Add addressing and compile description for multi-channel and single-channel models Update for new utility library