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

GDDR6
Palladium Memory Model
User Guide

Document Version: **1.3**

Document Date: **July 2018**

Contents

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.

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

GDDR6 Memory Model

1. Introduction

The Cadence Palladium GDDR6 Model is based on “Ballot of a Proposed GDDR6 SGRAM Full Specification Update Item 1836.99A for JEDEC committee JC-42.3C (JUN 2017).”

The model is available in several configurations with model.

Different sizes from 8Gb up to 16Gb are available, 24Gb up to 32Gb are TBD, please consult the memory model catalog for the current available list.

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

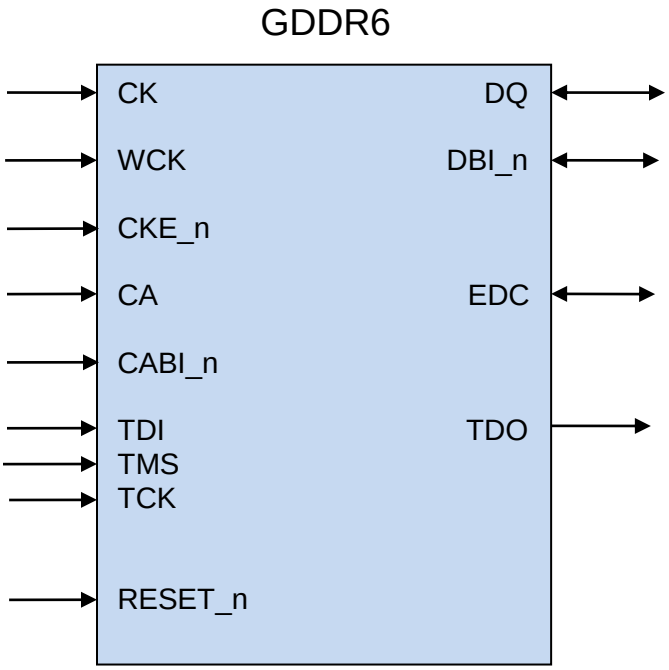
Access to Initial and Emerging Release versions of the models will require a Beta Agreement to be signed before the model can be delivered.

3. 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.

Memory Density	**8Gb**		**12Gb**		**16Gb**		**24Gb(TBD)**		**32Gb(TBD)**	
Device Organization	**x16 mode**	**x8 mode**	**x16 mode**	**x8 mode**	**x16 mode**	**x8 mode**	**x16 mode**	**x8 mode**	**x16 mode**	**x8 mode**
Number channels	2		2		2		2		2	
Channel density	4Gb		6Gb		8Gb		12Gb		16Gb	
Array pre-fetch (bits, per channel)	256	128	256	128	256	128	256	128	256	128
Bank address (per channel)	BA[3:0]		BA[3:0]		BA[3:0]		TBD		TBD	
Number Banks (per channel)	16		16		16		TBD		TBD	
Row address (per channel)	R[13:0]	R[13:0]	R[13:0]	R[14:0]	R[13:0]	R[14:0]	TBD	TBD	TBD	TBD
Number of Rows (per channel)	16384	16384	12288	24576	16384	32768	TBD	TBD	TBD	TBD
Column Address (per channel)	C[5:0]	C[6:0]	C[6:0]	C[6:0]	C[6:0]	C[6:0]	TBD	TBD	TBD	TBD
Page Size (per channel)	2K	2K	4K	2K	4K	2K	TBD	TBD	TBD	TBD

4. Model Block Diagram



5. I/O Signal Description

NAME	TYPE	DESCRIPTION
CK_t, CK_c	Input	Differential clock inputs shared between Channel A and B.
WCK0_t,WCK0_c, WCK1_t,WCK1_c (per channel)	Input	Write clocks: differential clocks used for WRITE data capture and READ data output. WCK0 is associated with DQ[7:0], DBI0_n and EDC0. WCK1 is associated with DQ[15:8], DBI1_n and EDC1. WCK0 and WCK1 are expected to be in phase and 4x CK frequency.
CKE_n (per channel)	Input	Clock enable: Low active.
CA[10:0] (per channel)	Input	Command and address inputs.
DQ[15:0] (per channel)	I/O	Data Input/Output: 16-bit data bus per channel.
DBI[1:0]_n (per channel)	I/O	Data Bus Inversion: DBI[0] is associated with DQ[7:0], DBI[1] is associated with DQ[15:8].
EDC[1:0] (per channel)	I/O	Error Detection Code. The calculated CRC data is transmitted on these signals. EDC[0] is associated with DQ[7:0], EDC[1] is associated with DQ[15:8]. Device ID input for x8 mode.
CABI_n (per channel)	Input	Command Address Bus Inversion.

NAME	TYPE	DESCRIPTION
TDI	Input	JTAG test data input.
TDO	Output	JTAG test data output.
TMS	Input	JTAG test mode select.
TCK	Input	JTAG test clock.
RESET_n	Input	RESET_n: Low active, asynchronously initiates a full chip reset.

5.1 Instantiation Example

```
jedec_gddr6_16gb_16 rtl_module (
    .CK_t(ck),
    .CK_c(ck_n),
    .WCK0_t_A(wck0_t_a),
    .WCK0_c_A(wck0_c_a),
    .WCK1_t_A(wck1_t_a),
    .WCK1_c_A(wck1_c_a),
    .CKE_n_A(cke_n_a),
    .CA_A(ca_a),
    .DQ_A(dq_a),
    .DBI_A(dbi_a),
    .EDC_A(edc_a),
    .CABI_n_A(cabi_n_a),

    .WCK0_t_B(wck0_t_b),
    .WCK0_c_B(wck0_c_b),
    .WCK1_t_B(wck1_t_b),
    .WCK1_c_B(wck1_c_b),
    .CKE_n_B(cke_n_b),
    .CA_B(ca_b),
    .DQ_B(dq_b),
    .DBI_B(dbi_b),
    .EDC_B(edc_b),
    .CABI_n_B(cabi_n_b),

    .TDI(tdi),
    .TDO(tdo),
    .TMS(tms),
    .TCK(tck),
    .RESET_n(rst_n)

);
```

6. Features

The following features table shows which features are currently supported or not supported.

6.1. Features Table

FEATURE	Symbol	SUPPORT	NOTE
COMMANDS			
NO OPERATION	NOP	Yes	
MODE REGISTER SET	MRS	Yes	
ACTIVATE	ACT	Yes	
READ	RD	Yes	
READ with Autoprecharge	RDA	Yes	
Load FIFO	LDFF	Yes	
READ Training	RDTR	Yes	EDC Latency tCRCRD = 0
WRITE without Mask	WOM	Yes	
WRITE without Mask with Autoprecharge	WOMA	Yes	
WRITE with double-byte mask	WDM	Yes	
WRITE with double-byte mask with Autoprecharge	WDMA	Yes	
WRITE with single-byte mask	WSM	Yes	
WRITE with single-byte mask with Autoprecharge	WSMA	Yes	
WRITE Training	WRTR	Yes	
PRECHARGE (per bank)	PRE	Yes	
PRECHARGE (all banks)	PREA	Yes	
PER-BANK REFRESH	REF	Yes	
REFRESH (all banks)	REFA	Yes	
POWER DOWN ENTRY	PDE	Partial	Command is accepted but no power down is performed
POWER DOWN EXIT	PDX	Partial	Command is accepted
SELF REFRESH ENTRY	SRE	Partial	Command is accepted but no self refresh is performed, data is retained
SELF REFRESH EXIT	SRX	Partial	Command is accepted
COMMAND ADDRESS TRAINING CAPTURE	CAT	Yes	
TRAINING			
Command Address	CADT	Yes	MR15[3:2]
WCK2CK Alignment		Yes	MR10[8]

FEATURE	Symbol	SUPPORT	NOTE
READ	RDTR	Yes	
WRITE	WRTR	Yes	
ADVANCED FEATURES			
Read Error Detection Code	RD CRC	Yes	MR4[9]
Write Error Detection Code	WR CRC	Yes	MR4[10]
Read Data Bus Inversion	RDBI	Yes	MR1[8]
Write Data Bus Inversion	WDBI	Yes	MR1[9]
Command Address Bus Inversion	CABI	Yes	MR1[10]
EDC hold pattern		Yes	MR4[3:0]
EDC hold pattern Invert		Yes	MR4[11]
EDC hold rate		Yes	MR2[11]
EDC Hi-Z		Yes	MR8[5]
EDC mode		Yes	MR2[8]
DQ preamble		Yes	MR7[5]
Vendor ID		Yes	MR3[7:6]
WCKINV		Yes	MR10[5:4]
CRC Read Latency	CRCRL	Yes	MR4[8:7]
CRC Write Latency	CRCWL	Yes	MR4[6:4]
Read Data Strobe	RDQS	Yes	MR2[9]
x8 mode and MRE Device ID		Yes	MR15[1:0]
PLL/DLL		No	
QDR WCK clocking		No	
DDR WCK clocking		Yes	WCK = 4x CK frequency
Read Latency Extra High Frequency		Yes	MR8[8]
JTAG signals/Boundary scan		No	
Pseudo-Channel (PC) mode		Yes	Enabled by driving CA6_A and CA6_B LOW when RESET_n de-asserts

7. Address Mapping

The array of the GDDR6 model is mapped into the internal memory of the Palladium system. This array is a single two dimensional array. The mapping of bank, row, column, and prefetch addresses to the internal model array is as follows:

ARRAY_ADDR = {BA, ROW, COL, prefetch_addr[3:0]}

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

The array name in the model hierarchy is: memcore

Example: memory –load %readmemh gddr6_inst.CH_A.memcore –file mem.dat

8. Model File List

The Palladium GDDR6 Memory Model consists of the following two files:

<model>.vp – wrapper that instantiates two core module instances

gddr6_pd.vp – core module of one channel in the model

The user should instantiate the wrapper, not the core module.

9. Model Parameter Descriptions

The following table provides details on the user adjustable parameters for the Palladium GDDR6 Memory Model. These parameters may be modified when instantiating a GDDR6 wrapper or, if necessary, by modifying the HDL parameter declarations and default values which are exposed for access and debug visibility.

Table 1: User Adjustable Parameters

User Adjustable Parameter	Default Value	Description
None available		

The following table provides some information about exposed parameters and localparams that are NOT user adjustable. There are exposed parameters in the core module that are adjustable for the wrapper but not for the user. The user should instantiate the wrapper, not the core module. On rare occasion the user may find one of these parameters or localparam needs adjusting for their configuration. If this case arises, please contact Cadence emulation or MMP support.

Table 2: Visible Non-User-Adjustable Parameters & Localparams

Parameter / Localparam	Default Value	Description
ROW_ADDR_WIDTH	14	The width of row address
COL_ADDR_WIDTH	7	The width of column address
BANK_ADDR_WIDTH	4	The width of bank address
CHANNEL_DENSITY	3'h3	Density per channel

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.

10. Mode Registers

The following contents are based on the November 2016 proposed JEDEC specification mentioned in the Introduction section.

All the mode registers for this model can be accepted and set. If the Debug Display feature is enabled messaging will indicate when a MRS command is accepted. However, a number of mode register fields remain unsupported in the sense of not being functionally implemented. These unsupported fields are indicated as such below.

10.1.MR0

Bits 11, 10, 9, 8 are not supported.

Field	Bit	Description
Write Recovery (WR)	MR8 OP[9],OP[11:8]	00000 – 4 00001 – 5 00010 – 6 00011 – 7 00100 – 8 00101 – 9 00110 – 10 00111 – 11 01000 – 12 01001 – 13 01010 – 14 01011 – 15 01100 – 16 01101 – 17 01110 – 18 01111 – 19 10000 – 20 10001 – 21 … 11110 – 34 11111 - 35
Test Mode	OP[7]	0 – normal mode 1 – not supported
Read Latency (RLmrs)	OP[6:3]	0000 – RFU 0001 – RFU 0010 – RFU 0011 – RFU 0100 – 9

		0101 - 10 0110 – 11 0111 – 12 1000 – 13 1001 – 14 1010 – 15 1011 – 16 1100 – 17 1101 – 18 1110 – 19 1111 - 20
Write Latency (WLmrs)	OP[2:0]	000 – 8 001 – RFU 010 – RFU 011 – RFU 100 – RFU 101 – 5 110 – 6 111 - 7

10.2.MR1

Bits 11, 7, 6, 5, 4, 3, 2, 1, 0 are not supported.

Field	Bit	Description
PLL/DLL Reset	OP[11]	0 – No 1 – Yes
CABI	OP[10]	0 – On 1 - Off
Write DBI	OP[9]	0 – On 1 – Off
Read DBI	OP[8]	0 – On 1 – Off
PLL/DLL	OP[7]	0 – Off 1 – On
Calibration Update	OP[6]	0 – On 1 – Off
CA Termination	OP[5:4]	00 – CKE_n value at RESET 01 – 60 ohm 10 – 120 ohm 11 – Disabled

Data Termination	OP[3:2]	00 – Disabled 01 – 60 ohm 10 – 120 ohm 11 – 48 ohm
Driver Strength	OP[1:0]	00 – Auto Calibration On (60/40) 01 – Auto Calibration On (48/40) 10 – vendor specific 11 – vendor specific

10.3.MR2

Bits 10, 7, 6, 5, 4, 3, 2, 1, 0 are not supported.

Field	Bit	Description
EDC Hold Rate (EDC HR)	OP[11]	0 – Full Rate 1 – Half Rate
CADT SRF	OP[10]	0 – Off 1 - On
RDQS	OP[9]	0 – Off 1 – On
EDC mode	OP[8]	0 – Full Rate 1 – Half Rate
Self Refresh	OP[7:6]	00 – 32 ms 01 – vendor specific 10 – vendor specific 11 – temperature controlled
OCD Pullup Driver Offset	OP[5:3]	000 – 0 001 - +1 010 - +2 011 - +3 100 - -4 101 - -3 110 - -2 111 - -1
OCD Pulldown Driver Offset	OP[2:0]	000 – 0 001 – +1 010 - +2 011 - +3 100 - -4 101 - -3

		110 - -2 111 - -1

10.4.MR3

Bits 11, 10, 9, 8, 5, 4, 3, 2, 1, 0 are not supported.

Field	Bit	Description
Bank Groups	OP[11:10]	0X – off / tCCDL = 2 tCK 10 – on / tCCDL = 4 tCK 11 – on / tCCDL = 3 tCK
WR Scaling	OP[9:8]	00 – 1x (default) 01 – 2x / optional 10 – 3x / optional 11 – RFU
DRAM Info	OP[7:6]	00 – off 01 - Vendor ID (ID1) 10 – Temperature Readout 11 - Vendor ID (ID2)
CA Termination Offset	OP[5:3]	000 – 0 001 - +1 010 - +2 011 - +3 100 - -4 101 - -3 110 - -2 111 - -1
Data and WCK Termination Offset	OP[2:0]	000 - 0 001 - +1 010 - +2 011 - +3 100 - -4 101 - -3 110 - -2 111 - -1

10.5.MR4

Bits 11 to 0 are currently supported.

Field	Bit	Description
EDC Hold Pattern Invert for EDC0_B + EDC1_A	OP[11]	0 – EDC hold pattern not inverted 1 – EDC hold pattern inverted
Write CRC	OP[10]	0 – On 1 – Off
Read CRC	OP[9]	0 – On 1 – Off
CRC Read Latency (CRCRL)	OP[8:7]	00 – 4 01 – 1 10 – 2 11 – 3
CRC Write Latency (CRCWL)	OP[6:4]	000 – 15 001 – 16 010 – RFU 011 – 10 100 – 11 101 – 12 110 – 13 111 - 14
EDC Hold Pattern	OP[3:0]	A background pattern transmitted on the EDC signals.

10.6.MR5

All bits are currently not supported.

Field	Bit	Description
RAS	OP[11:6]	000000 – RFU 000001 – 1 000010 – 2 000011 – 3 000100 – 4 000101 – 5 000110 – 6 000111 – 7 001000 – 8 001001 – 9 001010 – 10 001011 – 11 001100 – 12

		001101 – 13 001110 – 14 001111 – 15 … 111111 - 63
PLL/DLL Bandwidth	OP[5:3]	000 – vendor specific 001 – optional 010 – optional 011 – optional 100 – optional 101 – optional 110 – optional 111 - optional
Low Power Mode 3	OP[2]	0 – Off 1 - On
Low Power Mode 2	OP[1]	0 – Off 1 - On
Low Power Mode 1	OP[0]	0 – Off 1 – On

10.7.MR6

All bits are currently not supported.

Field	Bit	Description
Receiver Characteristics	OP[11:0]	MR6 and MR9 Control VREFD Levels

10.8.MR7

Bits 11, 10, 9, 8, 7, 6, 4, 3, 2, 1, 0 are not supported.

Field	Bit	Description
DCC	OP[11:10]	00 – DCC off / optional 01 – DCC start / optional 10 – RFU 11 – DCC hold / optional
VDD Range	OP[9:8]	00 – 0 01 – 1 / optional 10 – 2 / optional

		11 – 3 / optional
Half VREFD	OP[7]	0 – 0.7 * VDDQ 1 – 0.5 * VDDQ
Half VREFC	OP[6]	0 – 0.7 * VDDQ 1 – 0.5 * VDDQ
DQ Preamble	OP[5]	0 – Off 1 – On / optional
WCK2CK Auto Sync	OP[4]	0 – Off 1 – On / optional
Low Frequency Mode	OP[3]	0 – Off 1 – On / optional
PLL Delay Compensation	OP[2]	0 – Off 1 – On / optional
PLL Fast Lock	OP[1]	0 – Off 1 – On / optional
WCK2CK Alignment Point	OP[0]	0 – PD inside DRAM 1 – PD at balls

10.9.MR8

Bits 11, 10, 9, 7, 6, 4, 3, 2, 1, 0 are currently not supported.

Field	Bit	Description
CK Termination	OP[11:10]	00 – CKE_n value at RESET 01 – 60 ohm 10 – 120 ohm 11 - Disabled
WR EHF	OP[9]	0 – WR normal range (4 to 19 tCK) 1 – WR extended range (20 to 35 tCK)
RL EHF	OP[8]	0 – RLmrs normal range (5 to 20 tCK) 1 – RLmrs extended range (21 to 36 tCK)
REFpb	OP[7]	0 – REFpb 1 – REFp2b
CK Auto Calibration (CK AC)	OP[6]	0 – CK auto-calibration update during REFab enabled 1 – CK auto-calibration update during REFab disabled
EDC Hi-Z	OP[5]	0 – off 1 – on

CA Termination Override (CA TO)	OP[4]	0 – off 1 - on
CAH Termination	OP[3:2]	00 – Disabled 01 – 60 ohm 10 – 120 ohm 11 – 240 ohm
CAL Termination	OP[1:0]	00 – Disabled 01 – 60 ohm 10 – 120 ohm 11 - Reserved

10.10. MR9

All bits are currently not supported.

Field	Bit	Description
Receiver Characteristics	OP[11:0]	MR6 and MR9 Control VREFD Levels

10.11. MR10

Bits 11, 10, 9, 3, 2, 1, 0 are not supported.

Field	Bit	Description
WCK Termination	OP[11:10]	00 – Disabled 01 – 60 ohm 10 – 120 ohm 11 – RFU
WCK Ratio	OP[9]	0 – Half 1 – Quarter
WCK2CK	OP[8]	0 – Off 1 – On
WCK Inv / Quad Shift Byte 1	OP [7:6]	00 – invert off / Shift 0 degrees 01 – invert off / Shift 90 degrees 10 – invert on / Shift 180 degrees 11 – invert on / Shift 270 degrees.
WCK Inv / Quad Shift Byte 0	OP[5:4]	00 – invert off / Shift 0 degrees 01 – invert off / Shift 90 degrees 10 – invert on / Shift 180 degrees 11 – invert on / Shift 270 degrees

VREFC Levels	OP[3:0]	0000 – 0 / default 0001 – +1 0010 - +2 0011 - +3 0100 - +4 0101 - +5 0110 - +6 0111 - +7 1000 – 0 1001 - -7 1010 - -6 1011 - -5 1100 - -4 1101 - -3 1110 - -2 1111 - -1

10.12. MR11

All bits are currently not supported.

Field	Bit	Description
PASR Mask (all bits)	OP[11:0]	0 – Refresh enabled (= unmasked, default) 1 – Refresh blocked (= masked) X – Don't Care for the particular segment or banks
PASR Row Segment Mask	OP[11:8]	XXX1 – Segment 0 XX1X – Segment 1 X1XX – Segment 2 1XXX – Segment 3
PASR 2-Bank Mask	OP[7:0]	XXXXXXX1 – Bank 0 and 1 XXXXXX1X – Bank 2 and 3 XXXXX1XX – Bank 4 and 5 XXXX1XXX – Bank 6 and 7 XXX1XXXX – Bank 8 and 9 XX1XXXXX – Bank 10 and 11 X1XXXXXX – Bank 12 and 13 1XXXXXXX – Bank 14 and 15

10.13. MR12

All bits are currently not supported.

Field	Bit	Description
Sub Register	OP[11:8]	To extend the register space
RFU	OP[7:0]	Reserved for Future Use

10.14. MR13

All bits are currently not supported.

Field	Bit	Description
Reserved	OP[11:0]	Vendor Specific Features

10.15. MR14

All bits are currently not supported.

Field	Bit	Description
Sub Register	OP[11:8]	Not used
Reserved	OP[7:0]	Vendor Specific Features

10.16. MR15

Bits 11, 10, 9, 8, 7, 6, 5, 4 are currently not supported.

Field	Bit	Description
Reserved	OP[11:4]	Not used
Command Address Training (CADT)	OP[3:2]	00 – Off 01 – Train CA[9:0] Rising edge of CK using CAT 10 – Train CA[9:0] Falling edge of CK using CAT 11 – Train CABI_n, CA10 using CAT
Mode Register 0-14 Enable (MRE)	OP[1:0]	00 – Both non-mirrored and mirrored (default) 01 – mirrored only 10 – non-mirrored only 11 - Reserved

11. Initialization Sequence

The GDDR6 model requires that the memory controller follows the initialization sequence as documented in the proposed specification. The sequence basically entails the following steps:

1. Assert RESET_n
2. Drive CA6_A and CA6_B both HIGH for 2 channel mode or LOW for PC (pseudo-channel) mode
3. De-assert RESET_n
4. Start clocks
5. Wait for CKE_n to assert
6. At least one NOP is issued after CKE_n goes low
7. Set at least one Mode Register value using MRS command

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.

12. Compile and Emulation

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

jedec_gddr6_16gb_16.vp - wrapper or model
gddr6_pd.vp - core module instantiated by the wrapper

```
ixcom -ua +dut+jedec_gddr6_16gb_16 \
    ./jedec_gddr6_16gb_16.vp \
    ./gddr6_pd.vp \
     -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 jedec_gddr6_16gb_16.vp
hdlInputFile gddr6_pd.vp
hdlImport -full -2001 -l qtref
hdlOutputFile -add -f verilog jedec_gddr6_16gb_16.vg
hdlSynthesize -memory -keepVhdlCase -keepRtlSymbol -keepAllFlipFlop
jedec_gddr6_16gb_16
……

2)
vavlog jedec_gddr6_16gb_16.vp gddr6_pd.vp

vaelab -keepRtlSymbol -keepAllFlipFlop -outputVlog
jedec_gddr6_16gb_16.vg jedec_gddr6_16gb_16
……
```

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.

13. Runtime

If the user wishes to preload the memory array, the path to model instance can be found in dbFiles/*mpart.

Example paths:

```
26   16   2  tb_top.rtl_module.CH_A.memcore
26   16   2  tb_top.rtl_module.CH_B.memcore
```

Preloading example:

```
memory –load %readmemh tb_top.rtl_module.CH_A.memcore –file mem.dat
```

14. Limitations

Due to the lack of PLL/DLL the GDDR6 memory model only supports DDR WCK clocking. QDR WCK clocking is not supported. WCK0 and WCK1 inputs are expected to be in phase and 4x CK frequency. Other unsupported features are listed in Features Table.

15. Debugging

The GDDR6 model has several debugging options techniques and tips that may assist the user in isolating a problem.

- For issues that are may not be GDDR6 specific please review the *Memory Model Portfolio FAQ for All Models User Guide.*
- **Golden waveform:** A package with a reference waveform is available which shows the following command sequence:

  1. initialization sequence
     a. set up EDC and DBI_n signals before Reset_n goes high for x8 and mirrored modes, read vendor id and perform wck2ck training
  2. LDFF - load fifo
  3. RDTR - read training
  4. WRTR – write training
  5. RDTR – read training
  6. CAT – command address training
  7. precharge bank5
  8. precharge all
  9. refresh all

10. refresh bank5
11. activate bank5
12. write without mask
13. gapless write with single-byte mask
14. write with double-byte mask
15. write without mask with auto precharge
16. read data written without mask
17. gapless read data written with single-byte mask
18. read with auto precharge data written with double-byte mask

Key signals:
init_done – a low to high transition on init_done indicates the model went through initialization sequence and it is ready for normal operations.

init_state – a 0, 1, 2 sequence shows model is progressing through initialization sequence.

sdram_state – indicates which of the following commands the model is processing.

```
//sdram states
localparam  MRS_cmd     = 0;
localparam  REF_cmd     = 1;
localparam  REFA_cmd    = 2;
localparam  PRE_cmd     = 3;
localparam  PREA_cmd    = 4;
localparam  ACT_cmd     = 5;
localparam  NOP_cmd     = 6;
localparam  RD_cmd      = 7;
localparam  RDA_cmd     = 8;
localparam  WR_cmd      = 9;
localparam  WRA_cmd     = 10;
localparam  PDE_cmd     = 11;
localparam  PDX_cmd     = 12;
localparam  SRE_cmd     = 13;
localparam  SRX_cmd     = 14;
localparam  WDM_cmd     = 15;
localparam  WDMA_cmd    = 16;
localparam  WSM_cmd     = 17;
localparam  WSMA_cmd    = 18;
localparam  LDFF_cmd    = 19;
localparam  RDTR_cmd    = 20;
localparam  WRTR_cmd    = 21;
localparam  CAT_cmd     = 22;
```

- **Debug Display:** The Palladium GDDR6 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.

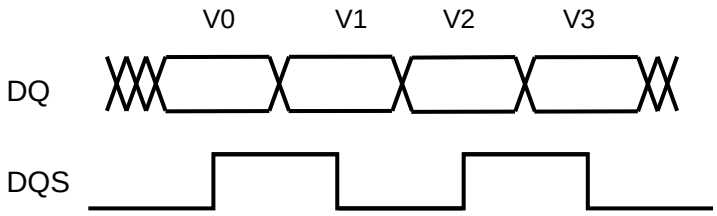
The following table lists the internal register path and naming along with the specification or datasheet naming for model mode registers or configuration registers that are accompanied by keep_net synthesis directives in support of such manual configuration. ONLY writeable configuration registers or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

Table: Writeable Mode Register / Configuration Register Info

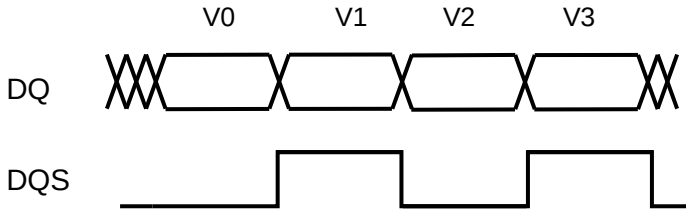
Hierarchical RTL Naming for Writeable Configuration Related Registers & Signals	Specification or Vendor Datasheet Naming for Configuration Related Registers	Access
<model_name>.MR0	MR0	W
<model_name>.MR1	MR1	W
<model_name>.MR2	MR2	W
<model_name>.MR3	MR3	W
<model_name>.MR4	MR4	W
<model_name>.MR5	MR5	W
<model_name>.MR6	MR6	W
<model_name>.MR7	MR7	W
<model_name>.MR8	MR8	W
<model_name>.MR9	MR9	W
<model_name>.MR10	MR10	W
<model_name>.MR11	MR11	W
<model_name>.MR12	MR12	W
<model_name>.MR13	MR13	W
<model_name>.MR14	MR14	W
<model_name>.MR15	MR15	W
<model_name>.init_done	[Not Applicable]	1’b1 indicates initialization is complete

16. 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.

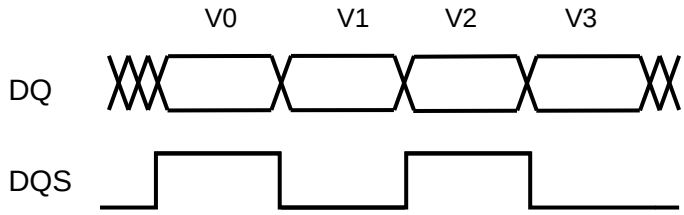


For DDR models provided by Cadence for Palladium, if the design drives DQ and DQS signals with the above timing, the DDR memory will behave correctly. However, to obtain this timing in Palladium, the fastest design clock must toggle twice as frequently as the DQS signal. If this faster clock is not needed 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 with 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.

17. Clocking

The GDDR6 memory model only supports DDR WCK clocking. QDR WCK clocking is not supported. WCK0 and WCK1 are expected to be in phase and 4x CK frequency.

18. Revision History

The following table shows the revision history for this document.

Date	Version	Revision
January 2017	0.1	Initial Release
February 2017	0.2	Add support for Read, Write, Command Address training, EDC hold pattern, and Debug Display
April 2017	0.3	Add support for CABI, DBI, WCK2CK, WCKINV, x8 mode, MRE, Vendor ID, REDC, CRCRL, and DQ Preamble
July 2017	0.4	Add support for CE bit in Pseudo-Channel (PC) mode, update MR4 and MR8 per v0.16r2 spec
September 2017	0.5	Update Features Table PDE, PDX, SRE, SRX support from No to Partial
October 2017	1.0	Remove watermark; Move model and user guide from BETA level to release
January 2018	1.1	Modify header and footer
June 2018	1.2	Add Manual Configuring description to Debugging section
July 2018	1.3	Update for new utility library