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

HBM (High Bandwidth Memory)
Palladium Memory Model
User Guide

Document Version: **3.0**

Document Date: **July 2018**

Copyright © 2013-2018 Cadence Design Systems, Inc. All rights reserved.
Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

© 2013-2018 All rights reserved.

Contents

© 2013-2018 All rights reserved.

© 2013-2018 All rights reserved.

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

© 2013-2018 All rights reserved.

HBM Memory Model

1. Introduction

The Cadence Palladium HBM Model is based on the data sheet specifications of the following JEDEC Specification:

High Bandwidth Memory (HBM) DRAM JESD 235 Specification Rev 2.10 (November 2015).

The model can be configured to support different configurations of sizes and other features to match real devices manufactured by various vendors.

© 2013-2018 All rights reserved.

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.

© 2013-2018 All rights reserved.

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.

	Channel Density Legacy Mode[4]			Channel Density Pseudo Channel Mode[4]			
	1 Gb	2 Gb	4 Gb	2 Gbit	4 Gbit	8 Gbit	8 Gbit 8-High[5]
Prefetch Size (bits)	256	256	256	256 128 for PC	256 128 for PC	256 128 for PC	256 128 for PC
Row Address	RA12:RA0	RA13:RA0	RA13:RA0	RA13:RA0	RA13:RA0	RA14:RA0	RA13:RA0
Column Address	CA5:CA0	CA5:CA0	CA5:CA0	CA5:CA0	CA5:CA0	CA5:CA0	CA5:CA0
Bank Address	BA2:BA0	BA2:BA0	BA3:BA0	BA2:BA0	BA3:BA0	BA3:BA0	SID,BA3:BA0

1. In Pseudo channel mode, an additional address bit BA4 is provided for RAS and CAS commands to direct commands either to Pseudo Channel 0 (BA4=0) or Pseudo Channel 1 (BA4=1).
2. The “8Gbit 8-High” addressing is a specific configuration that is optimized for an HBM stack using 8 DRAM dies. The stack height of all other configuration is vendor specific.
3. The stack ID (SID) acts as a bank address bit in command execution.

3.1 Pseudo Channel Mode

1. Pseudo channel mode requires that the burst length be set to 4. Both Pseudo channels share a given channel’s mode registers.
2. There can be only two imPRE ACT in one pseudo channel until the tRP time is satisfied.
3. Target Mode Refresh (TRR) mode is self-clearing. The TRR mode will be disabled automatically after the completion of defined TRR flow. At this time MR5 will be self-cleared.
4. When TRR mode is enabled, imPRE ACT is not supported.
5. While in TRR mode, the only commands allowed are ACT and PRE until the TRR mode has been completed.
6. To enable Pseudo Channel mode, the user needs to add “+define+MMP_HBM_PSEUDO_CHANNEL” in the synthesis phase of compilation as in below examples:
    a. vavlog –define MMP_HBM_PSEUDO_CHANNEL
    b. hdlInputFile -add ./hbm.vp +define+MMP_HBM_PSEUDO_CHANNEL

3.2 Stack ID (SID) Support

The user needs to define the Verilog macro MMP_HBM_SID_SUPPORT in the synthesis phase of compilation to enable SID support (8 Gbit 8-HIGH addressing) as in below examples:

a. vavlog –define MMP_HBM_SID_SUPPORT
b. hdlInputFile -add ./hbm.vp +define+ MMP_HBM_SID_SUPPORT

© 2013-2018 All rights reserved.

3.3 Error Correction Code (ECC) Support

The HBM memory model provides support for the extra interface and for the extra storage cells that are needed in HBM ECC mode. While the MMP HBM model handles the MR4 settings, supports the 16 bit signal path and ports at the DM_CB interface, and provides additional DRAM storage as described for ECC mode, the memory model does not compute ECC or detect or correct errors. The user needs to define the Verilog macro MMP_HBM_ECC_SUPPORT in the synthesis phase of compilation to enable the portion of ECC support provided in the HBM memory model, as in below examples:

a. vavlog –define MMP_HBM_ECC_SUPPORT
b. hdlInputFile -add ./hbm.vp +define+ MMP_HBM_ECC_SUPPORT

To word this another way, the MMP HBM model is capable of receiving, storing, and transmitting the ECC bits when enabled with MR4 and MMP_HBM_ECC_SUPPORT macro; however the model does not perform ECC calculations, detection, or correction.

4 Verilog Macro Defines Required and Optional

At this time there are no required (as opposed top optional) Verilog macro `define(s) for the Palladium HBM model. The following table lists the optional Verilog macro defines the user may want to consider.

Table 1: HBM Optional Verilog Defines

`define Macro purpose	Optional Verilog `define Values
Define this Verilog macro during synthesis to enable SID support (8Gbit 8-High addressing)	MMP_HBM_SID_SUPPORT
Define this Verilog macro during synthesis to enable Pseudo Channel mode support	MMP_HBM_PSEUDO_CHANNEL
Define this Verilog macro during synthesis to enable ECC support	MMP_HBM_ECC_SUPPORT

5 Model Parameter & Localparam Description

The following tables provide details on the **user adjustable** parameters and **fixed** local parameters for the Palladium HBM Model. These parameters may be modified when instantiating an HBM wrapper or, if necessary, by modifying the HDL parameter declarations and default values which are exposed for access and debug visibility.

5.1 Top Level Parameters

The following table provides details on the parameters of the Palladium HBM memory model.

Parameter	Default Value	Description
ROW_ADDR_WIDTH	15	Indicates number of rows in each bank
COL_ADDR_WIDTH	6	indicates number of columns in each row
BANK_ADDR_WIDTH	4	indicates number of banks in each channel

© 2013-2018 All rights reserved.

© 2013-2018 All rights reserved.

5.2 IEEE1500 Wrapper

The below table lists some visible localparams in the hbm_ieee1500_wrapper module in the hbm_ieee1500_wrapper.v file.

Localparam	Default Value	Description
DEV_GEN2_TEST	1'b1	Gen2 testing support, default value is 1 means GEN2 testing is supported.
DEV_ECC	1'b1	Default value is 1 means ECC is supported.
DEV_DENSITY	4'b0100	Device density, default size is 8Gbit
DEV_MANUFACTURE_ID	4'b0001	Device manufacture id, default is '1' for samsung
DEV_MANUFACTURING_LOCATION	4'b0000	Device manufacturing location, vendor specific, default is 0
DEV_MANUFACTURING_YEAR	8'b00000011	Device manufacturing year, default is 2014
DEV_MANUFACTURING_WEEK	8'b00000001	Device manufacturing week, default is week '1'
DEV_SERIAL_NUMBER	34'h123456789	Device serial number, vendor specific, default is 34'h123456789
DEV_ADDRESSING_MODE	2'b01	Device addressing mode, default is pseudo channel mode. The default behavior is the same as with defining Verilog macro MMP_HBM_PSEUDO_CHANNEL
DEV_CHANNEL_AVAILABLE	8'b11111111	Device channel available, default is for all channels to be available
DEV_STACK_HIGH	1'b1	Device stack high setting. The default is '1' for "8Gbit 8 high" This default behavior is the same as defining the Verilog macro MMP_HBM_SID_SUPPORT
DEV_MODEL_PART_NUMBER	7'b0000000	Device model part number, vendor reserved, default is 0
TEMP_VALID	1'b0	Setting for whether temperature function is valid or not, default value is '0' for 'valid'.
TEMP_VALUE	7'd25	Temperature value setting. The default value is 25'C

© 2013-2018 All rights reserved.

5.3 HBM_Memory

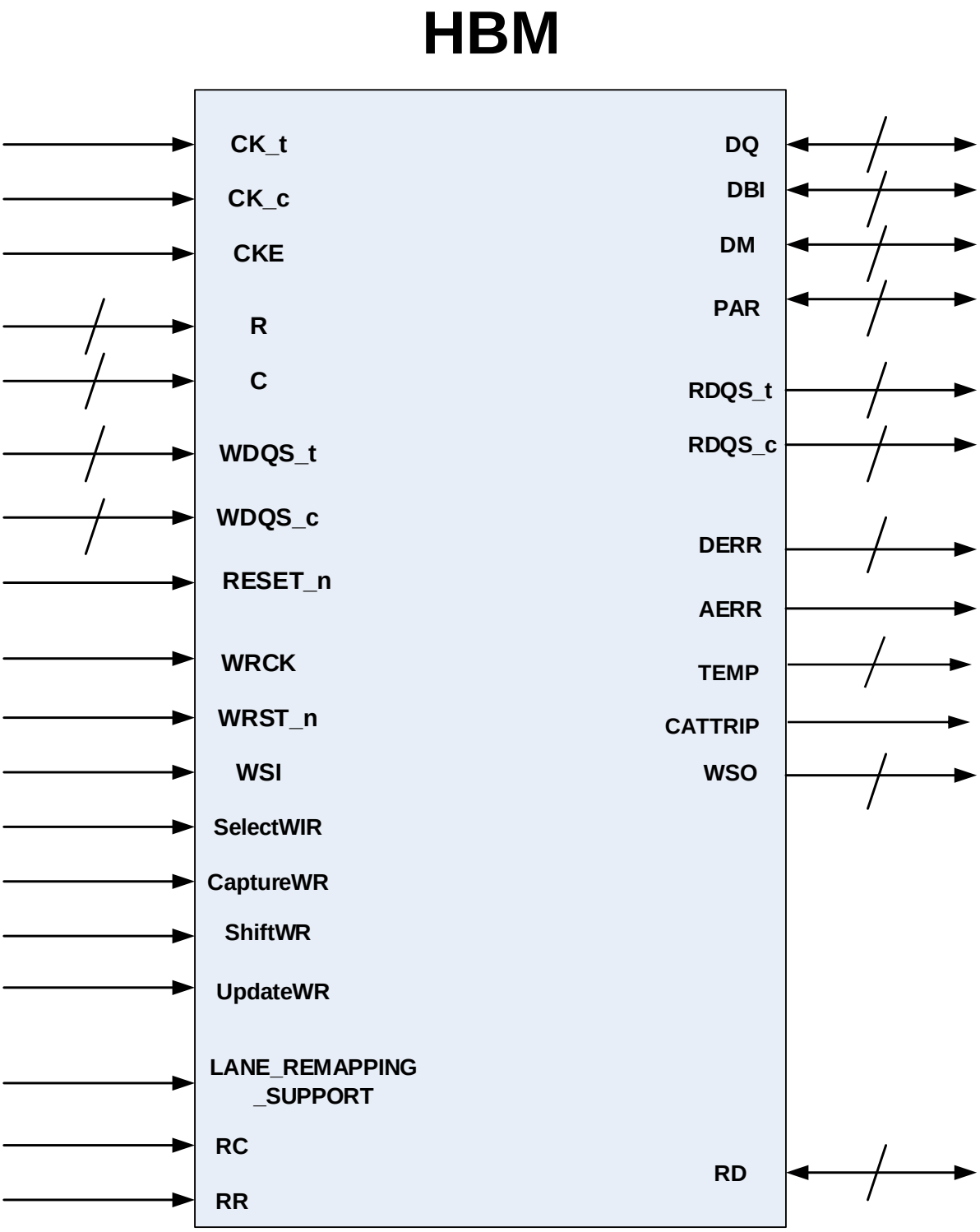
The below table lists some visible localparams in the hbm_mem module in the hbm_mem.v file.

Loalparam	Default Value	Description
tRP	12	Precharge command period
tMOD	12	MODE REGISTER SET command update delay
tCCDS	1	RD/WR bank A to RD/WR bank B command delay different bank groups
tCCDL	2	RD/WR bank A to RD/WR bank B command delay same bank group
tPARAC	2	ADD/CMD parity error output delay
tPARDQ	2	Write data parity error output delay

© 2013-2018 All rights reserved.

6 Model Block Diagram

The diagram below shows all the ports of HBM Model. The widths of the Addr, Ba, Dm, Dq, and Dqs buses are dependent on the density of the part being used.



© 2013-2018 All rights reserved.

7 I/O Signal Description

In addition to the standard IO pins for the HBM as per the JEDEC specification there are several other IO on the HBM Palladium memory model that are there to provide flexibility and the ability to model additional features.

NAME	TYPE	DESCRIPTION
CK_t, CK_c	Input	Clock
CKE	Input	Clock enable
C[7:0]	Input	Column command and address: the command code, bank and column address for Write and Read operations and the Mode Register address and code to be loaded with Mode Register Set commands are received on the C[7:0] inputs.
R[5:0]	Input	Row command and address: the command code, bank and row address for Activate, Precharge and Refresh commands are received on the R[5:0] inputs.
DQ[127:0]	I/O	Data Input/Output: 128-bit data bus
DBI[15:0]	I/O	Data Bus Inversion: DBI[0] is associated with DQ[7:0], DBI[1] is associated with DQ[15:8], ... , and DBI[15] is associated with DQ[127:120].
DM[15:0]	I/O	Data Mask: DM[0] is associated with DQ[7:0], DM[1] is associated with DQ[15:8], ... , and DM[15] is associated with DQ[127:120].
PAR[3:0]	I/O	Data Parity: one data parity bit per DWord. PAR[0] is associated with DQ[31:0], PAR[1] is associated with DQ[63:32], PAR[2] is associated with DQ[95:64], PAR[3] is associated with DQ[127:96].
DERR[3:0]	Output	Data parity error: one data parity error bit per DWord. DERR[0] is associated with DQ[31:0], DERR[1] is associated with DQ[63:32], DERR[2] is associated with DQ[95:64], DERR[3] is associated with DQ[127:96].
AERR	Output	Address parity error. One address parity error bit for row and column address and command per channel.
WDQS_t[3:0] WDQS_c[3:0]	Input	Write Data Strobe: WDQS_t and WDQS_c are differential strobe inputs. Write input data are latched on the rising and falling edges of WDQS_t,WDQS_c. One WDQS pair per DWord. WDQS_t[0] and WDQS_c[0] is associated with DQ[31:0], WDQS_t[1] and WDQS_c[1] is associated with DQ[63:32], WDQS_t[2] and WDQS_c[2] is associated with DQ[95:64], WDQS_t[3] and WDQS_c[3] is associated with DQ[127:96].
RDQS_t[3:0] RDQS_c[3:0]	Output	Read Data Strobe: RDQS_t and RDQS_c are differential strobe inputs. Read output data are sent on the rising and falling edges of RDQS_t,RDQS_c. One RDQS pair per DWord. RDQS_t[0] and RDQS_c[0] is associated with DQ[31:0], RDQS_t[1] and RDQS_c[1] is associated with DQ[63:32], RDQS_t[2] and RDQS_c[2] is associated with DQ[95:64], RDQS_t[3] and RDQS_c[3] is associated with DQ[127:96].

© 2013-2018 All rights reserved.

NAME	TYPE	DESCRIPTION
RESET_n	Input	RESET_n Low asynchronously initiates a full chip reset of the HBM device.
TEMP[2:0]	Output	Temperature Report. Not supported, tied to 0.
CATTRIP	Output	Catastrophic Temperature Report. Not supported, tied to 0.
WRCK	Input	IEEE-1500 Wrapper Serial Port Clock
WRST_n	Input	IEEE-1500 Wrapper Serial Port RESET
WSI	Input	IEEE-1500 Wrapper Serial Port Instruction Register Select
SelectWIR	Input	IEEE-1500 Wrapper Serial Port Shift
CaptureWR	Input	IEEE-1500 Wrapper Serial Port Capture
ShiftWR	Input	IEEE-1500 Wrapper Serial Port Update
UpdateWR	Input	IEEE-1500 Wrapper Serial Port Data
WSO[7:0]	Output	IEEE-1500 Wrapper Serial Port Data OutChannels [7:0]
LANE_REMAPPING_SUPPORT	Input	Force this port to 1 to enable lane_remapping support
RD	I/O	Redudant microbumps in DWORD, for lane remapping
RC	Input	Redundant Column microbump in AWORD, for lane remapping
RR	Input	Redudant Row microbump in AWORD, for lane remapping

7.1 Instantiation Example

```
hbm_top rtl_module (
    .CH0_CK_t(ck),
    .CH0_CK_c(ck_n),
    .CH0_CKE(cke),
    .CH0_RESET_n(rst_n),
    .CH0_R(R),
    .CH0_C(C),
    .CH0_RDQS_t(rdqs),
    .CH0_RDQS_c(rdqs_n),
    .CH0_WDQS_t(wdqs),
    .CH0_WDQS_c(wdqs_n),
    .CH0_DQ     (real_dq),
    .CH0_DBI    (real_dbi),
    .CH0_DM_CB  (real_dm_cb),
    .CH0_PAR(par),
    .CH0_DERR(derr),
    .CH0_AERR(aerr),
    .CH0_RD(real_rdata),
    .CH0_RC(RC),
    .CH0_RR(RR),

    .LANE_REMAPPING_SUPPORT(1'b1),
    .CH1_CK_t(),
    .CH1_CK_c(),
    .CH1_CKE(),
    .CH1_RESET_n(),
    .CH1_R(),
```


© 2013-2018 All rights reserved.

```
.CH1_C(),
.CH1_RDQS_t(),
.CH1_RDQS_c(),
.CH1_WDQS_t(),
.CH1_WDQS_c(),
.CH1_DQ(),
.CH1_DBI(),
.CH1_DM_CB(),
.CH1_PAR(),
.CH1_DERR(),
.CH1_AERR(),

.CH2_CK_t(),
.CH2_CK_c(),
.CH2_CKE(),
.CH2_RESET_n(),
.CH2_R(),
.CH2_C(),
.CH2_RDQS_t(),
.CH2_RDQS_c(),
.CH2_WDQS_t(),
.CH2_WDQS_c(),
.CH2_DQ(),
.CH2_DBI(),
.CH2_DM_CB(),
.CH2_PAR(),
.CH2_DERR(),
.CH2_AERR(),

.CH3_CK_t(),
.CH3_CK_c(),
.CH3_CKE(),
.CH3_RESET_n(),
.CH3_R(),
.CH3_C(),
.CH3_RDQS_t(),
.CH3_RDQS_c(),
.CH3_WDQS_t(),
.CH3_WDQS_c(),
.CH3_DQ(),
.CH3_DBI(),
.CH3_DM_CB(),
.CH3_PAR(),
.CH3_DERR(),
.CH3_AERR(),

.CH4_CK_t(),
.CH4_CK_c(),
.CH4_CKE(),
.CH4_RESET_n(),
.CH4_R(),
```


© 2013-2018 All rights reserved.

```
.CH4_C(),
.CH4_RDQS_t(),
.CH4_RDQS_c(),
.CH4_WDQS_t(),
.CH4_WDQS_c(),
.CH4_DQ(),
.CH4_DBI(),
.CH4_DM_CB(),
.CH4_PAR(),
.CH4_DERR(),
.CH4_AERR(),

.CH5_CK_t(),
.CH5_CK_c(),
.CH5_CKE(),
.CH5_RESET_n(),
.CH5_R(),
.CH5_C(),
.CH5_RDQS_t(),
.CH5_RDQS_c(),
.CH5_WDQS_t(),
.CH5_WDQS_c(),
.CH5_DQ(),
.CH5_DBI(),
.CH5_DM_CB(),
.CH5_PAR(),
.CH5_DERR(),
.CH5_AERR(),

.CH6_CK_t(),
.CH6_CK_c(),
.CH6_CKE(),
.CH6_RESET_n(),
.CH6_R(),
.CH6_C(),
.CH6_RDQS_t(),
.CH6_RDQS_c(),
.CH6_WDQS_t(),
.CH6_WDQS_c(),
.CH6_DQ(),
.CH6_DBI(),
.CH6_DM_CB(),
.CH6_PAR(),
.CH6_DERR(),
.CH6_AERR(),

.CH7_CK_t(),
.CH7_CK_c(),
.CH7_CKE(),
.CH7_RESET_n(),
.CH7_R(),
```


© 2013-2018 All rights reserved.

```
    .CH7_C(),
    .CH7_RDQS_t(),
    .CH7_RDQS_c(),
    .CH7_WDQS_t(),
    .CH7_WDQS_c(),
    .CH7_DQ(),
    .CH7_DBI(),
    .CH7_DM_CB(),
    .CH7_PAR(),
    .CH7_DERR(),
    .CH7_AERR(),

    //test IOs
    .WRCK(WRCK),
    .WRST_n(WRST_n),
    .WSI(WSI),
    .SelectWIR(SelectWIR),
    .CaptureWR(CaptureWR),
    .ShiftWR(ShiftWR),
    .UpdateWR(UpdateWR),
    .WSO(WSO)
);
```

8 Commands

The HBM commands are rising/falling sample mode. The Row Active command requires two cycles.

8.1 Row Commands

Function	Symbol	Clock Cycle	CKE		R[0]	R[1]	R[2]	R[3]	R[4]	R[5]
			(n-1)	n						
Row No Operation	RNOP	Rising	H	H	H	H	H	V	V	V
		Falling			V	V	PAR	V	V	V
Activate	ACT	Rising	H	H	L	H	RA14/ SID	BA0	BA1	BA2
		Falling			RA11	RA12	PAR	RA15/ BA4	RA13	BA3
		Rising	H	H	RA5	RA6	RA7	RA8	RA9	RA10
		Falling			RA0	RA1	PAR	RA2	RA3	RA4
Precharge	PRE	Rising	H	H	H	H	L	BA0	BA1	BA2

© 2013-2018 All rights reserved.

		Falling			V	V/SID	PAR	BA4	L	BA3
Precharge All	PREA	Rising	H	H	H	H	L	V	V	V
		Falling			V	V	PAR	BA4	H	V
Single Bank Refresh	REFSB	Rising	H	H	L	L	H	BA0	BA1	BA2
		Falling			V	V/SID	PAR	BA4	L	BA3
Refresh	REF	Rising	H	H	L	L	H	V	V	V
		Falling			V	V	PAR	BA4	H	V
Power Down Entry	PDE[1]	Rising	H	L	H	H	H	V	V	V
		Falling			V	V	PAR	V	V	V
Self Refresh Entry	SRE[1]	Rising	H	L	L	L	H	V	V	V
		Falling			V	V	PAR	V	V	V
Power Down & Self Refresh Exit	PDX/ SRX[1]	Rising	L	H	H	H	H	V	V	V
		Falling			V	V	V	V	V	V

Notes:
1. The Palladium module accepts these analog commands at input but does not support these analog actions.

8.2 Column Commands

Function	Symbol	Clock Cycle	C[0]	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]
Column No Operation	CNOP	Rising	H	H	H	V	V	V	V	V
		Falling	V	V	PAR	V	V	V	V	V
Read	RD	Rising	H	L	H	L	BA0	BA1	BA2	BA3
		Falling	CA0/ V/SID	CA1	PAR	CA2	CA3	CA4	CA5	CA6/ BA4

© 2013-2018 All rights reserved.

Read w/AP	RDA	Rising	H	L	H	H	BA0	BA1	BA2	BA3
		Falling	CA0/ V/SID	CA1	PAR	CA2	CA3	CA4	CA5	CA6/ BA4
Write	WR	Rising	H	L	L	L	BA0	BA1	BA2	BA3
		Falling	CA0/ V/SID	CA1	PAR	CA2	CA3	CA4	CA5	CA6/ BA4
Write w/AP	WRA	Rising	H	L	L	H	BA0	BA1	BA2	BA3
		Falling	CA0/ V/SID	CA1	PAR	CA2	CA3	CA4	CA5	CA6/ BA4
Mode register set	MRS	Rising	L	L	L	OP7	BA0	BA1	BA2	BA3
		Falling	OP0	OP1	PAR	OP2	OP3	OP4	OP5	OP6

© 2013-2018
All rights reserved.

9 Read/Write Operation Notes

9.1 Read

HBM drives Odd byte signals to Low 2 tCK prior to Read data and Even byte signals to Low 1 tCK prior to Read data.

The HBM device must store the last beat of all DQ. Also the RDQS should be issued for the last beat data.

RDQS is 1 tCK prior to read data.

9.2 Write

WDQS is 1 tCK prior to the center of write data.

10 DBIac function

10.1 Odd/Even Bytes Group

DQ Signals	DBI Signal	DM/CB Signal	Byte Type
DQ[7:0]	DBI0	DM0/CB0	Even Byte
DQ[15:8]	DBI1	DM1/CB1	Odd Byte
DQ[23:16]	DBI2	DM2/CB2	Even Byte
DQ[31:24]	DBI3	DM3/CB3	Odd Byte
DQ[39:32]	DBI4	DM4/CB4	Even Byte
DQ[47:40]	DBI5	DM5/CB5	Odd Byte
DQ[55:48]	DBI6	DM6/CB6	Even Byte
DQ[63:56]	DBI7	DM7/CB7	Odd Byte
DQ[71:64]	DBI8	DM8/CB8	Even Byte
DQ[79:72]	DBI9	DM9/CB9	Odd Byte
DQ[87:80]	DBI10	DM10/CB10	Even Byte
DQ[95:88]	DBI11	DM11/CB11	Odd Byte
DQ[103:96]	DBI12	DM12/CB12	Even Byte
DQ[111:104]	DBI13	DM13/CB13	Odd Byte
DQ[119:112]	DBI14	DM14/CB14	Even Byte
DQ[127:120]	DBI15	DM15/CB15	Odd Byte

10.2 DBIac Reset

1) When reset signal de-assertion.
2) When MRS
3) READ/READA after WRITE command
4) Self Refresh exit

© 2013-2018 All rights reserved.

11 Address Mapping

The array of the HBM 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, ROW, COL}

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

The array name in the model hierarchy is: memcore

© 2013-2018 All rights reserved.

12 Features List and Mode Registers

• The High Bandwidth Memory (HBM) Palladium memory model is based upon JESD235 at the specification level referenced in Section Introduction.

12.1 Features List for the HBM Memory Model

Feature	Support
MR0	
Test Mode	Partially supported
Address, Command Bus Parity	YES
DQ Bus Write Parity	YES
DQ Bus Read Parity	YES
TCSR (Temperature Compensated Self Refresh)	NO
Write DBIac	YES
Read DBIac	YES
MR1	
Drive Strength	NO
Setting write recovery delay	YES
MR2	
Setting read latency	YES
Setting write latency	YES
MR3	
Setting burst length	YES
Bank group	YES
Setting RAS	YES
MR4	
Parity latency	YES
Data mask	YES
ECC	YES[1]
MR5	
TRR mode including BANK setting and pseudo channel select	YES
MR6	
Setting imPRE tRP	YES
MR7	
CATTRIP	NO
Test mode control	Partially supported

© 2013-2018 All rights reserved.

	(only OP[2:1]=2'b11 is NOT supported)

Note 1: The memory model does not compute ECC or detect or correct errors.

12.2 Instructions List for Test Mode

Feature	Support
BYPASS	YES
EXTEST_RX	YES
EXTEST_TX	YES
INTEST_RX	NO (it's shown vendor specific and optional in spec)
INTEST_TX	NO (it's shown vendor specific and optional in spec)
HBM_RESET	NO (it's shown optional in spec)
MBIST	NO (WDR is shown vendor specific in spec)
SOFT_REPAIR	NO (WDR is shown vendor specific in spec)
HARD_REPAIR	NO (WDR is shown vendor specific in spec)
DWORD_MISR	YES
AWORD_MISR	YES
CHANNEL_ID	YES
MISR_MASK	YES
AWORD_MISR_CONFIG	YES
DEVICE_ID	YES
TEMPERATURE	YES
MODE_REGISTER_DUMP_SET	YES
READ_LFSR_COMPARE_STICKY	YES
SOFT_LANE_REPAIR	YES
HARD_LANE_REPAIR	YES

12.3 Mode Registers

The following tables describe each field of the mode registers in detail.

© 2013-2018 All rights reserved.

12.3.1 MR0

Field	Bit	Description
Test Mode	OP[7]	0 – normal operation 1 – test mode (vendor specific)
Address, Command Bus Parity for Row, Column Bus Enable	OP[6]	0 – disable 1 – enable
DQ Bus Write Parity Enable	OP[5]	0 – disable 1 – enable
DQ Bus Read Parity Enable	OP[4]	0 – disable 1 – enable
Write DBIac Enable	OP[1]	0 – disable 1 – enable
Read DBIac Enable	OP[0]	0 – disable 1 – enable

12.3.2 MR1

Field	Bit	Description
Write Recovery WR (for Auto-Precharge Only)	OP[4:0]	00000 – Reserved 00001 – Reserved 00010 – Reserved 00011 – 3 nCK 00100 – 4 nCK 00101 – 5 nCK 00110 – 6 nCK 00111 – 7 nCK 01000 – 8 nCK …… 111111 – 31 nCK

12.3.3 MR2

Field	Bit	Description
Read Latency	OP[7:3]	00000 – Reserved 00001 – 3 nCK (minimum) 00010 – 4 nCK

© 2013-2018 All rights reserved.

		00011 – 5 nCK 00100 – 6 nCK 00101 – 7 nCK 00110 – 8 nCK 00111 – 9 nCK …… 111111 – 33 nCK
Write Latency	OP[2:0]	000 – 1 nCK (minimum) 001 – 2 nCK 010 – 3 nCK 011 – 4 nCK 100 – 5 nCK 101 – 6 nCK 110 – 7 nCK 111 – 8 nCK

© 2013-2018 All rights reserved.

12.3.4 MR3

Field	Bit	Description
BL (Burst Length)	OP[7]	0 – BL2 1 – BL4
Bank Group Enable	OP[6]	0 – disable 1 – enable
Activate to Precharge RAS	OP[5:0]	000000 – Reserved 000001 – Reserved 000010 – Reserved 000011 – 3 nCK 000100 – 4 nCK 000101 – 5 nCK 000110 – 6 nCK 000111 – 7 nCK …… 1111111 – 63 nCK

Table 11 — MR3 - Burst Type and Burst Order Definition - BL2

Burst Type	Burst Length	Read/Write	Burst Order
Sequential	2	Read	0, 1
		Write	0, 1

Table 12 — MR3 - Burst Type and Burst Order Definition - BL4

Burst Type	Burst Length	Read/Write	Starting Column Address CA0	Burst Order	Note
Sequential	4	Read	0	0, 1, 2, 3	1
			1	2, 3, 0, 1	1
		Write	0	0, 1, 2, 3	1
			1	2, 3, 0, 1	1

Burst re-order via address bit CA0 is supported in legacy mode only. The burst order is fixed in Pseudo channel mode.

12.3.5 MR4

Field	Bit	Description
PL	OP[3:2]	00 – 0 nCK 01 – 1 nCK 10 – 2 nCK 11 – 3 nCK
DM Disable	OP[1]	0 – enable

© 2013-2018 All rights reserved.

		1 – disable NOTE: DM and ECC cannot be enabled simultaneously. i.e., OP[1:0] = 01 is not allowed.
ECC	OP[0]	0 – enable 1 – disable NOTE: DM and ECC cannot be enabled simultaneously. i.e., OP[1:0] = 01 is not allowed. Note: the HBM memory model does not compute ECC or detect or correct errors.

12.3.6 MR5

Field	Bit	Description
TRR Mode Enable	OP[7]	0 – disable 1 – enable
TRR – Pseudo Channel Select	OP[6]	0 – enable TRR mode for PS 0 1 – enable TRR mode for PS 1
TRR- BAn	OP[3:0]	0000 – BANK 0 0001 – BANK 1 0010 – BANK 2 …… 1111 – BANK 15

© 2013-2018 All rights reserved.

12.3.7 MR6

Field	Bit	Description
imPRE tRP value	OP[7:3]	00000 – 2 nCK (minimum) 00001 – 3 nCK 00010 – 4 nCK …… 11111 – 33 nCK

12.3.8 MR7

Field	Bit	Description
DWORD MISR Control	OP[5:3]	Only applicable if Loopback is enabled in OP[0] 000 – preset. The DWORD MISR/LFSRs are set to 0xAAAAAh, and the DWORD LFSR_COMPARE_STICKY bits are set to all zeroes. 001 - LFSR mode (read direction) 010 - Register mode (read and write directions) DWORD writes are captured directly into the MISR registers without compression. The MISR registers will contain the most recent write data. 011 - MISR mode (write direction) 100 - LFSR Compare mode (write direction)
DWORD Read Mux Control	OP[2:1]	Only applicable if Loopback is enabled in OP[0] 00 - Reserved 01 - Return data from MISR registers (default) 10 - Return data from Rx path sampler 11 - Return LFSR_COMPARE_STICKY (optional)
Loopback Enable	OP[0]	0 - Disable 1 - Enable; Enables Link testing circuitry. All Writes and Reads will be to/from the MISR. (Does not require any additional Activation to

© 2013-2018 All rights reserved.

		Write/Reads - Column addresses are ignored in this mode.)

13 Initialization Sequence

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

1. Assert RESET
2. De-assert RESET
3. Start clocks
4. Wait for CKE to asserted
5. At least one NOP is issued after CKE goes high
6. Set all MRS value which will be used, at least one MRS

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

© 2013-2018 All rights reserved.

14 Lane Remapping

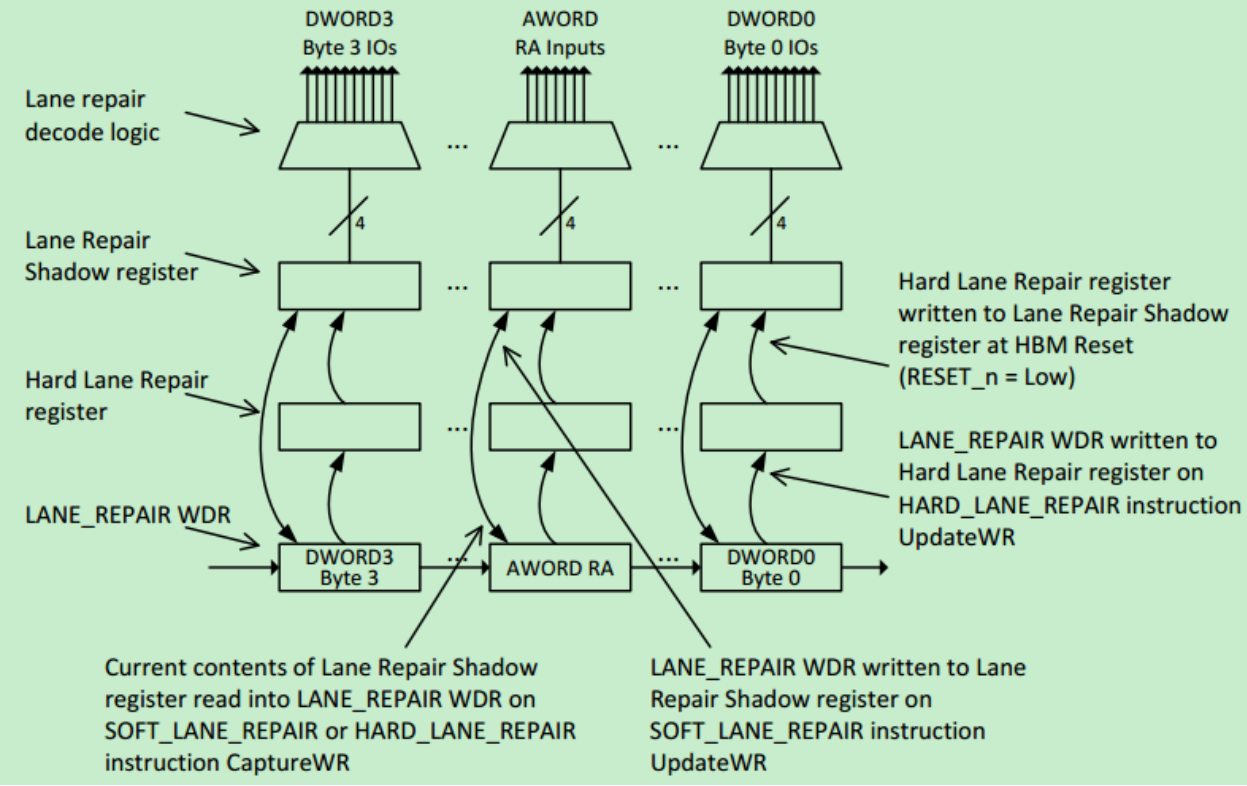
The HBM DRAM supports interconnect lane remapping to help improve DRAM yield and recover functionality of the HBM stack. LANE_REPAIR WDR is used to perform lane remapping. Lane remapping is independent for each channel.

The IEEE-1500 instructions for SOFT_LANE_REPAIR and HARD_LANE_REPAIR are used to convey lane remapping and repair information. Both instructions use the same LANE_REPAIR WDR.

When SOFT_LANE_REPAIR is the current instruction, the UpdateWR event will load the lane remapping data from the shift stage of the WDR into the lane repair shadow register and force the I/O lanes to be remapped accordingly. This remapping is non-persistent; it will be lost when RESET_n is pulled low or the device loses power. Pulling WRST_n low does not reset the lane repair shadow register.

When HARD_LANE_REPAIR is the current instruction, the UpdateWR event will load the lane remapping data from the shift stage of the WDR into the hard lane repair register. The controller must wait to allow the device to complete this operation and permanently store the repair vector.

The flow of the two instructions is shown below.



In the Palladium HBM model, the following ports are used for lane remapping (section 7 of UG):

- **LANE_REMAPPING_SUPPORT**: Force this port to 1 to enable lane_remapping support
- **RD**: Redundant microbumps in DWORD, for lane remapping

© 2013-2018 All rights reserved.

- **RC**: Redundant Column microbump in AWORD, for lane remapping
- **RR**: Redundant Row microbump in AWORD, for lane remapping

14.1 AWORD Remapping (Row Command Bus)

Description	Register Encoding	Rx0	Rx1	Rx2	Rx3	Rx4	Rx5	RRx
Repair Lane 0	0000	XX	Rx0	Rx1	Rx2	Rx3	Rx4	Rx5
Repair Lane 1	0001	Rx0	XX	Rx1	Rx2	Rx3	Rx4	Rx5
Repair Lane 2	0010	Rx0	Rx1	XX	Rx2	Rx3	Rx4	Rx5
Repair Lane 3	0011	Rx0	Rx1	Rx2	XX	Rx3	Rx4	Rx5
Repair Lane 4	0100	Rx0	Rx1	Rx2	Rx3	XX	Rx4	Rx5
Repair Lane 5	0101	Rx0	Rx1	Rx2	Rx3	Rx4	XX	Rx5
Reserved	0110 to 1110	Rx0	Rx1	Rx2	Rx3	Rx4	Rx5	RFU
Default - No Repair	1111	Rx0	Rx1	Rx2	Rx3	Rx4	Rx5	RFU

© 2013-2018 All rights reserved.

14.2 AWORD Remapping (Column Command Bus)

Description	Register Encoding	Cx0	Cx1	Cx2	Cx3	Cx4	Cx5	Cx6	Cx7	RCx
Repair Lane 0	0000	XX	Cx0	Cx1	Cx2	Cx3	Cx4	Cx5	Cx6	Cx7
Repair Lane 1	0001	Cx0	XX	Cx1	Cx2	Cx3	Cx4	Cx5	Cx6	Cx7
Repair Lane 2	0010	Cx0	Cx1	XX	Cx2	Cx3	Cx4	Cx5	Cx6	Cx7
Repair Lane 3	0011	Cx0	Cx1	Cx2	XX	Cx3	Cx4	Cx5	Cx6	Cx7
Repair Lane 4	0100	Cx0	Cx1	Cx2	Cx3	XX	Cx4	Cx5	Cx6	Cx7
Repair Lane 5	0101	Cx0	Cx1	Cx2	Cx3	Cx4	XX	Cx5	Cx6	Cx7
Repair Lane 6	0110	Cx0	Cx1	Cx2	Cx3	Cx4	Cx5	XX	Cx6	Cx7
Repair Lane 7	0111	Cx0	Cx1	Cx2	Cx3	Cx4	Cx5	Cx6	XX	Cx7
Reserved	1000 to 1110	Cx0	Cx1	Cx2	Cx3	Cx4	Cx5	Cx6	Cx7	RFU
Default - No Repair	1111	Cx0	Cx1	Cx2	Cx3	Cx4	Cx5	Cx6	Cx7	RFU

© 2013-2018 All rights reserved.

14.3 DWORD Remapping

Two modes are provided to remap the data bus:

- **Mode 1:** In Mode 1 it is allowed to remap one lane per byte. No redundant pin is allocated in this mode, and DBI functionality is lost for that byte only. However, other bytes continue to support the DBI function as long as the Mode Register setting for the DBI function is enabled. If Data Parity function is enabled in the Mode Register and a lane is remapped, both DRAM and host assume DBI input as “0” for parity calculation for Read and Write operation in this mode. In Mode 1 each byte is treated independently.

- **Mode 2:** In Mode 2 the user is allowed to remap one lane per double byte. One redundant pin (RD) per double byte is allocated in this mode, and DBI functionality is preserved as long as the Mode Register setting for DBI function is enabled. The use of Mode 2 has no impact on the Data Parity function. In Mode 2 two adjacent bytes (e.g. DQ[15:0]) are treated as a pair (double byte), but each double byte is treated independently.

The two modes are distinguished by the use of the “1110b” encoding as shown in the table below. The use of Mode 1 is assumed when a remapping code other than “1110b” is used. For Mode 2 it is required to program “1110b” for the intact byte within the double byte while the remapping for the broken lane in the other byte is encoded according to the table.

© 2013-2018 All rights reserved.

Description	Register Encoding	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	DBIx0	RDx0
Repair Lane 0	0000	XX	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 1	0001	DMx0	XX	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 2	0010	DMx0	DQx0	XX	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 3	0011	DMx0	DQx0	DQx1	XX	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 4	0100	DMx0	DQx0	DQx1	DQx2	XX	DQx3	DQx4	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 5	0101	DMx0	DQx0	DQx1	DQx2	DQx3	XX	DQx4	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 6	0110	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	XX	DQx5	DQx6	DQx7	RFU/ DBIx0
Repair Lane 7	0111	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	XX	DQx6	DQx7	RFU/ DBIx0
Repair Lane 8	1000	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	XX	DQx7	RFU/ DBIx0
Repair Lane 9	1001	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	XX	RFU/ DBIx0
Reserved	1010 to 1101	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	DBIx0	RFU
Repair in other byte (Mode 2 only)	1110	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	DBIx0	RFU
Default - No Repair	1111	DMx0	DQx0	DQx1	DQx2	DQx3	DQx4	DQx5	DQx6	DQx7	DBIx0	RFU

15 IEEE1500 MISR Features and Behavior

This section addresses some features and behaviors that generally apply to the MISR modes.

a) **Entering the MISR modes** - MISR modes may be entered any time after completing the initialization (see Initialization). DWORD MISR modes are controlled via Mode Register 7, while AWORD MISR modes are controlled via the IEEE 1500 port AWORD_MISR_CONFIG instruction. AWORD and DWORD MISR modes cannot be used simultaneously since the DWORD MISR modes are driven via READ and WRITE commands on the AWORD bus.

b) **Entering and exiting AWORD MISR modes** - The sequence for entering AWORD MISR modes, and then exiting back to normal operation is as follows:

- At any time after initializing the HBM enters the all banks idle state.
- Enter either the Precharge Power-Down state or the Self-Refresh state. CKE=0 while in these states.

© 2013-2018 All rights reserved.

- Stop toggling CK (CK_t=0, CK_c=1).
- Enable/enter and operate the AWORD MISR modes (AWORD_MISR_CONFIG Enable = 1-On). Finish these operations with CK stopped (CK_t=0, CK_c=1) and CKE=0.
- Disable the AWORD MISR modes and follow the Power-Down (PDE, PDX) or Self-Refresh (SRE, SRX) exit procedures.

c) Entering and exiting DWORD MISR modes - The sequence for entering DWORD MISR modes, and then exiting back to normal operation is as follows:

- At any time after initializing the HBM enter the all banks idle state.
- Set all configuration mode registers as needed for use in the DWORD MISR modes.
- Set MR7 DWORD Loopback Enable = 1, and then wait tMOD.
- Enter either precharge power-down or self-refresh. CKE = 0 while in these states.
- Select and operate the DWORD MISR modes via MR7 settings and sending RD, WR, and CNOP commands. After completing DWORD MISR operations, send CNOP commands.
- Follow the power-down exit (PDX) or self-refresh exit (SRX) procedures,
- Set MR7 DWORD Loopback Enable = 0 - Disable, and then wait tMOD before continuing normal operation.

d) Command decode is disabled in AWORD MISR modes - When AWORD MISR modes are enabled the traffic sent on the AWORD bus is not limited to valid commands. To prevent undefined states and operations, when AWORD MISR modes are enabled (AWORD_MISR_CONFIG Enable = 1 - On), command decoding is disabled.

e) With lane repairs the MISR bit positions remain with their logical signals - The MISR bits are associated with their logic signals, not the physical microbumps. For example, if DQ3 has been repaired (which routes the DQ3 data to the DQ4 microbump) the data received on the DQ4 microbump is routed to the DQ3 Rise and Fall MISR bits. Effectively, the behaviors for all MISR modes are unchanged when Mode 2 lane repairs are active - all 10 bits of the byte are captured in the MISR in the same bit locations, as if no lane repair were active.

f) HBM DBI, Write Mask, and ECC logic circuits are not functional in the DWORD MISR modes - The DBI and DM signals are treated as pure data signals. Their raw values are captured, compared, or sent without regard to their normal bus inversion or masking/ECC functional meaning.

g) AWORD and DWORD write parity checking- In AWORD and DWORD Register mode, MISR mode, and LFSR Compare mode the HBM parity evaluation logic is **INVALID**.

16 IEEE1500 MISR Testing

This section addresses how to test MISR features.

16.1 Test AWORD (write) MISR mode

a) Initialize the test sequence by setting MR7 DWORD Loopback Enable = 1'b1 – Enable and presetting the MISR registers by setting the DWORD MISR Control = 3'b000 - Preset.
b) Enable DWORD MISR mode by setting MR7 DWORD MISR Control = 'b011 - MISR mode.

© 2013-2018 All rights reserved.

c) The host sends one or more DWORD write cycles following the write latency and burst length setting and following the normal write protocol. The HBM clocks the received data into the DWORD MISRs.
d) The host reads the MISR content via the IEEE 1500 DWORD_MISR instruction.

16.2 Test DWORD (write) MISR mode

a) Initialize the test sequence by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and presetting the MISR registers by setting the DWORD MISR Control = 3'b000 - Preset.
b) Enable DWORD MISR mode by setting MR7 DWORD MISR Control = 'b011 - MISR mode.
c) The host sends one or more DWORD write cycles following the write latency and burst length setting and following the normal write protocol. The HBM clocks the received data into the DWORD MISRs.
d) The host reads the MISR content via the IEEE 1500 DWORD_MISR instruction.

16.3 Test AWORD (write) REGISTER mode

a) After the required HBM initialization, the host asserts either Precharge Power-Down or Self-Refresh mode (CKE = 0) and stops sending CK clocks to the HBM (CK_t = 0, CK_c = 1).
b) Initialize the AWORD MISR by setting the AWORD_MISR_CONFIG Enable = 1'b1 - On and AWORD_MISR_CONFIG MODE = 3'b000 - Preset. The Preset operation enables the preamble clock filter circuit.
c) Enable the AWORD Register mode by setting AWORD_MISR_CONFIG MODE = 3'b010 - Register mode.
d) The host sends two or more CK clock cycles and data on the AWORD signals. The first received CK clock cycle is discarded as a preamble clock by the HBM. The HBM clocks the raw received data into the AWORD MISR register without MISR compression. The ending clock state applied by the host is CK_t = 0, CK_c = 1. The last clocked DDR cycle data is retained in the AWORD MISR register.
e) The host reads the MISR content via the IEEE 1500 READ_AWORD_MISR instruction.

16.4 Test DWORD (write) REGISTER mode

a) Enable DWORD Register mode by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and DWORD MISR Control = 3'b010 - Register mode. A Preset is not required prior to using Register mode.
b) The host sends one or more DWORD write cycles following the write latency and burst length setting and following the normal write protocol. The HBM clocks the raw received data into the DWORD MISR registers without MISR. The last clocked DDR cycle data is retained in the DWORD MISR registers.
c) The host reads the MISR content via the IEEE 1500 DWORD_MISR instruction. The MISR content is also readable via the functional interface

16.5 Test DWORD (read) REGISTER mode

a) Enable the test mode and select the desired read-back register by setting MR7 DWORD Loopback Enable = 1'b1 - Enable, DWORD MISR Control = 3'b010 - Register mode, and DWORD Read Mux Control = one of the defined register sources.
b) The host sends one or more DWORD read commands. The HBM responds following the read latency and burst length setting and following the normal read protocol.

© 2013-2018 All rights reserved.

16.6 Test DWORD (read) LFSR mode

a) Initialize the test sequence by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and presetting the MISR registers by setting the DWORD MISR Control = 3'b000 - Preset. Optionally, load the DWORD MISR registers with an alternate seed value via the functional interface.
b) Enable DWORD LFSR mode by setting MR7 DWORD MISR Control = 3'b001 - LFSR mode and DWORD Read Mux Control = 2'b01 - Return data from MISR registers.
c) The host sends one or more DWORD read commands. The HBM responds following the read latency and burst length setting and following the normal read protocol, with data produced from the LFSR.

16.7 Test AWORD (write) LFSR compare mode

a) After the required HBM initialization, the host asserts either Precharge Power-Down or Self-Refresh mode (CKE = 0) and stops sending CK clocks to the HBM (CK_t = 0, CK_c = 1).
b) Initialize the AWORD MISR (LFSR) register by setting the AWORD_MISR_CONFIG Enable = 1'b1 - On and AWORD_MISR_CONFIG MODE = 3'b000 - Preset. The Preset operation also clears the AWORD per-signal sticky error bits and enables the preamble clock filter circuit. The host-side LFSR data generator should also be initialized to the same value.
c) Enable the AWORD LFSR Compare mode by setting AWORD_MISR_CONFIG MODE = 3'b100 - LFSR Compare mode.
d) The host sends two or more CK clock cycles with LFSR-generated data on the AWORD signals. The first received CK clock cycle is discarded as a preamble clock by the HBM. The HBM LFSR predicts expected AWORD data per cycle from the host, based on matching LFSR polynomials and starting seeds in the host and HBM. Any mismatches set sticky error for the respective signal. Parity is evaluated, if enabled. The ending clock state applied by the host is CK_t = 0, CK_c = 1.
e) The host reads the Sticky error bits to determine which signals failed. The bits are readable via the IEEE 1500 port READ_LFSR_COMPARE_STICKY instruction.

16.8 Test DWORD (write) LFSR compare mode

a) Initialize the DWORD LFSR (MISR) registers by setting MR7 DWORD Loopback Enable = 1'b1 - Enable and DWORD MISR Control = 3'b000 - Preset. The Preset operation also clears the DWORD per-signal sticky error bits. The host-side LFSR data generator should also be preset/initialized to the same value.
b) Enable DWORD LFSR Compare mode by setting MR7 DWORD MISR Control = 3'b100 - LFSR Compare mode.
c) The host sends one or more DWORD write cycles with LFSR-generated data on the DWORD signals following the write latency and burst length setting and following the normal write protocol. The HBM LFSRs predict expected DWORD data per cycle from the host, based on matching LFSR polynomials and starting seeds in the host and HBM. Any mismatches set sticky error for the respective signal. Parity is evaluated, if enabled.
d) The host reads the sticky error bits to determine which signals failed. The bits are readable via the IEEE 1500 port READ_LFSR_COMPARE_STICKY instruction. It's not supported to read sticky error bits via the functional interface.

© 2013-2018 All rights reserved.

17 Error Debug

The following flags in HBM memory core can be monitored to indicate bad cycles:

Flag	Description
- err_init_cmd	Command other than MRS, ZQC, NOP is issued during initialization
- err_init_mrs	MRS initialization error
- err_mrs	MRS issued without all the banks precharged
- err_ref	REF issued without all the banks precharged
- err_refa	REFA issued without the specific banks precharged
- err_act	ACT issued to a bank not precharged
- err_impre_act	Number of IMPRE ACT is larger than 2
- err_trr	TRR sequence errors
- err_wr/err_rd	WRITE/READ issued to a bank not active
- bad_rd/bad_wr	WRITE/READ doesn't satisfy tCCD

18 Synthesis and Compilation of the Model

The model is provided as 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.

```
ixcom     -64bit +sv +dut+hbm_top -ua \
          -incdir ./rtl \
          ./rtl/hbm_pd.vp \
          ./rtl/hbm_mem.vp \
          ./rtl/hbm_ieee1500_wrapper.vp \
          ./rtl/hbm_channel.vp \
          ./rtl/hbm_top.v \
          -incdir ../../../utils/cdn_mmp_utils/sv \
          ../../../utils/cdn_mmp_utils/sv/cdn_mmp_utils.sv
          ./user_tb.sv \
          +define+MMP_HBM_PSEUDO_CHANNEL \
          +ignoreNCVerCheck

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

The scripts below show two examples for Palladium classic ICE synthesis:

1)

```
hdlInputFile -add ./rtl/hbm_pd.vp +define+MMP_HBM_PSEUDO_CHANNEL
hdlInputFile -add ./rtl/hbm_mem.vp +define+MMP_HBM_PSEUDO_CHANNEL
hdlInputFile -add ./rtl/hbm_ieee1500_wrapper.vp
hdlInputFile -add ./rtl/hbm_channel.vp
```


© 2013-2018 All rights reserved.

```
hdlInputFile -add ./rtl/hbm_top.v
hdlImport -atb -full -l qtref

hdlOutputFile -add -f verilog hbm_top.vg
hdlSynthesize -memory -keepRtlSymbol -keepAllFlipFlop hbm_top
……
```

2)

```
vavlog      ./rtl/hbm_pd.vp \
            ./rtl/hbm_mem.vp \
            ./rtl/hbm_ieee1500_wrapper.vp \
            ./rtl/hbm_channel.vp \
            ./rtl/hbm_top.v \
            -define MMP_HBM_PSEUDO_CHANNEL \
            ……
vaelab      -keepRtlSymbol -keepAllFlipFlop –outputVlog hbm_top.vg
hbm_top
```

NOTE: It is common for UXE 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. The -atb option is required only if user plans to use Debug Display during runtime. Please see Debugging section for more information on Debug Display feature.

It is also common for UXE 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.

18.1 Model File List

hbm_top.v – model wrapper that instantiates 8 channels
hbm_channel.vp – channel module that instantiates memory and test wrapper blocks
hbm_ieee1500_wrapper.vp –Test Wrapper block that supports Loopback Test Modes
hbm_mem.vp – memory block that instantiates one HBM core or two HBM cores to support pseudo channel mode
hbm_pd.vp – HBM core

© 2013-2018 All rights reserved.

19 Reference Waveform

A waveform showing various operations is available in the release under the golden_waveform directory. It may be useful to have a look at this waveform when using the model for the first time.

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 may not be HBM 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_state: Referencing the initialization sequence described in the user guide section on “Initialization Sequence,” the signal init_state starts at state 0 and should increment to state 1 after step 5 of the initialization sequence and should increment to state 2 after step 6.

  - init_done: The signal init_done is asserted when the initialization sequence is completed.

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

  - Waveform 1: Basic HBM functional sequence:
    - Set MISR_MASK through IEEE1500 test wrapper, R/C/DQ/DBI/DM_CB are remapped
    - Precharge ALL
    - Configurate MR0 ~ MR4
    - REFA
    - Operate on pseudo channel 0
    - Operate on pseudo channel 1
    - PS channel cross access
    - IMPRE act
    - TRR mode

  - Waveform 2: HBM Pseudo Channel mode with all commands sent in PS channel[0]:
    - HBM initialization: set WL=2, RL=3, BL=4
    - AWORD (WRITE) MISR Mode
    - DWORD (WRITE) MISR Mode
    - AWORD (WRITE) REGISTER Mode
    - DWORD (WRITE) REGISTER Mode
    - DWORD (READ) REGISTER Mode
    - normal READ/WRITE
    - MISR_MASK

© 2013-2018 All rights reserved.

  - EXTEST_RX
  - EXTEST_TX
  - AWORD_LFSR_COMPARE
  - DWORD_LFSR_COMPARE

- **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: Writeable Mode Register / Configuration Register Info

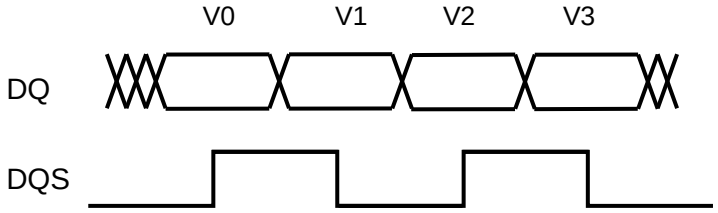
Hierarchical RTL Naming for Writeable Configuration Related Registers & Signals	Specification or Vendor Datasheet Naming for Configuration Related Registers	Access
<model_name>.MR0	MR0	W
<model_name>.MR1	MR1	W
<model_name>.MR2	MR2	W
<model_name>.MR3	MR3	W
<model_name>.MR4	MR4	W
<model_name>.MR5	MR5	W
<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

© 2013-2018 All rights reserved.

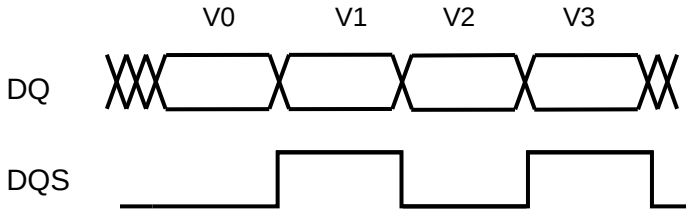
<model_name>.MR14	MR14	W
<model_name>.MR15	MR15	W
<model_name>.init_done	[Not Applicable]	1'b1 indicates initialization is complete

21 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.



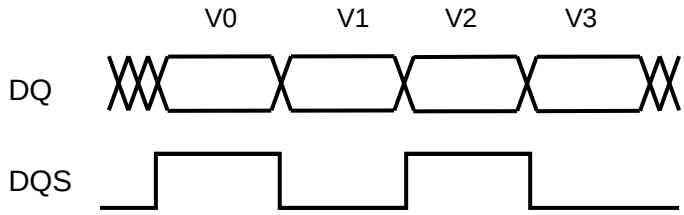
For DDR models provided by Cadence for Palladium, if the design drives DQ and DQS signals with the above timing, the DDR memory will behave correctly. However, to obtain this timing in Palladium, the fastest design clock must toggle twice as frequently as the DQS signal. If this faster clock is not needed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

© 2013-2018 All rights reserved.



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.

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
```


© 2013-2018 All rights reserved.

```
endmodule
```


© 2013-2018 All rights reserved.

22 Revision History

The following table shows the revision history for this document

Date	Version	Revision
October 2013	1.0	Initial Release
March 2014	1.1	Pseudo channel access is supported TRR mode and imPRE are supported
July 2014	1.2	Repaired doc property title.
September 2014	1.3	Remove version from UG file name. Updated SID support info. Update DQS Handling section.
November 2014	1.4	Remove emulation capacity info.
January 2015	1.5	Removed paragraph mentioning non-existent localparams. Notes added re ECC. Many tables updated for clarification of values and support level.
February 2015	1.6	Added section about lane remapping. Update related publications list.
March 2015	1.7	Update to support v1.30
July 2015	1.8	Replace ${AXIS_HOME} with ${UXE_HOME} based path
July 2015	1.9	Update Cadence naming on front page
September 2015	2.0	Move from BETA to non-BETA and place in release. Modify compile notes to reflect *.vp as sole model format
January 2016	2.1	Update for Palladium-Z1 and VXE
January 2016	2.2	Update to support v1.41
April 2016	2.3	Update to support v2.10
June 2016	2.4	Clarify ECC handling with additional words.
May 2017	2.5	Add TEMP and CATTRIP signals.
June 2017	2.6	Additional description in “Debugging” section
January 2018	2.7	Modify header and footer
May 2018	2.8	Additional note on -atb in synthesis example
June 2018	2.9	Add Manual Configuring description in Debugging section
July 2018	3.0	Update for new utility library

© 2013-2018 All rights reserved.