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

Mobile DDR
Palladium Memory Model
User Guide

Document Version: 2.1

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.

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

Mobile DDR Memory Model

1. Introduction

The Cadence Palladium Mobile DDR Model is based on the JEDEC specification JESD209A.

The model is available in several configurations with model sizes to match real devices manufactured by the following vendors: Micron, Hynix and Samsung.

Different sizes from 64Mb up to 2Gb are available, please consult the memory model catalog for the current available list. The Mobile DDR model supports standard page size as well as reduced page size parts.

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, 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

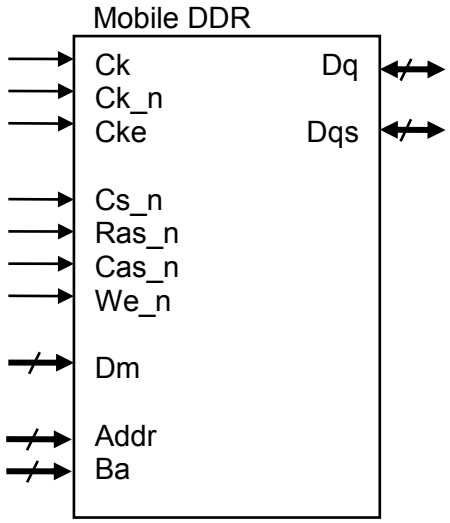
The following table lists the possible configurations. Not all configurations are available from all vendors. Please consult the appropriate vendor site for details on the parts they offer.

Data Width		64Mb	128Mb	256Mb	512Mb		1Gb		2Gb
	Banks	4	4	4	4		4		4
x16	Row Address	A[11:0]	A[11:0]	A[12:0]	A[12:0]		A[13:0]		A[13:0]
	Column Address	A[8:0]	A[8:0]	A[8:0]	A[9:0]		A[9:0]		A[11,9:0]
x32	Row Address	A[10:0]	A[11:0]	A[11:0]	A[12:0]	A[13:0]	A[12:0]	A[13:0]	A[13:0]
	Column Address	A[7:0]	A[7:0]	A[8:0]	A[8:0]	A[7:0]	A[9:0]	A[8:0]	A[9:0]

The auto pre-charge bit is A10

4. Model Block Diagram

The widths of the Addr, Ba, Dm, Dq, and Dqs buses are dependent on the density of the part being used.



5. Address mapping

The array of the mobile DDR 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

6. Register Definitions

In the mobile DDR there are three registers, the Mode Register, the Extended Mode Register and the Status Register Read.

6.1. Mode Register

The mode register is implemented in the mobile DDR model. The model supports the following parameter values.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CAS Latency			Burst Type	Burst Length		

Bit 6	Bit 5	Bit 4	CAS Latency
0	0	0	Not supported
0	0	1	Not supported
0	1	0	2
0	1	1	3
1	0	0	Not supported
1	0	1	Not supported
1	1	0	Not supported
1	1	1	Not supported

Bit 3	Burst Type	
0	Sequential	Supported
1	Interleaved	Supported

Bit 2	Bit 1	Bit 0	Burst Length
0	0	0	Not supported
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	Not supported
1	0	1	Not supported
1	1	0	Not supported
1	1	1	Not supported

6.2. Extended Mode Register

The extended mode register is not supported.

6.3. Status Register Read (SRR)

The status register read is not supported.

7. Commands

The mobile DDR model accepts the following commands:

- Deselect
- NOP
- Mode Register Set
- Activate
- Precharge
- Precharge All
- Read
- Read with AP
- Write
- Write with AP
- Burst Terminate
- Auto Refresh

8. Compile and Emulation

Each 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) and run of this model in the IXCOM flow is shown below.

```
ixcom -64 +sv -ua +dut+mt46h16m16lf \
    ./mt46h16m16lf.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 example for Palladium classic ICE synthesis:

```
1)
hdlInputFile mt46h16m16lf.vp
hdlImport -full -2001 -l qtref
hdlOutputFile -add -f verilog mt46h16m16lf.vg
hdlSynthesize -memory -keepVhdlCase -keepRtlSymbol -keepAllFlipFlop
mt46h16m16lf
……

2)
vavlog mt46h16m16lf.vp

vaelab -keepRtlSymbol -keepAllFlipFlop -outputVlog mt46h16m16lf.vg
mt46h16m16lf
……
```

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-

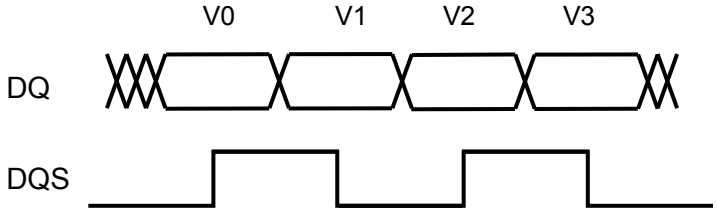
ICE can remove flops with constant inputs and merge equivalent FF. The picture above is modified a bit when ICE ATB mode (–atb) is used since then a constant input FF is only optimized out when there is no initial value for it or the initial value is the same as the constant input value.

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.

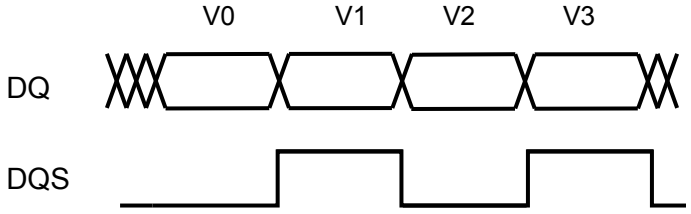
If the recommended compile script includes the aforementioned options, the user must include them to avoid affecting functionality of the design.

9. 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.

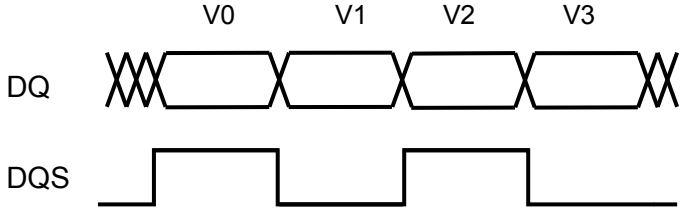


For DDR models provided by Cadence for Palladium, if the design drives DQ and DQS signals with the above timing, the DDR memory will behave correctly. However, to obtain this timing in Palladium, the fastest design clock must toggle twice as frequently as the DQS signal. If this faster clock is not needed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

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
```

Revision History

The following table shows the revision history for this document

Date	Version	Revision
April 2010	1.0	Initial version
January 2010	1.1	File renamed without spaces
June 2011	1.2	Updated with "Related Documents" information
December 2011	1.3	Minor user guide updates
July 2014	1.4	Repaired doc property title. Added revision history. Updated legal.
September 2014	1.5	Remove version from UG file name. Update UXE / IXE documentation reference titles. Added paragraph about flow independent delay cell in “Handling DQS …” section
November 2014	1.6	Remove emulation capacity info. Update related publications list.
July 2015	1.7	Update Cadence naming on front page
January 2016	1.8	Update for Palladium-Z1 and VXE
July 2016	1.9	Remove hyphen in Palladium naming
January 2018	2.0	Modify header and footer
July 2018	2.1	Update for new utility library