cādence®

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

Consumer FCRAM
Palladium Memory Model
User Guide

Document Version: **2.1**

Document Date: **July 2018**

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.

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. 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

Consumer FCRAM Memory Model

1. Introduction

The Cadence Palladium Consumer FCRAM Memory models are available with model sizes matching real Consumer FCRAM manufactured by Fujitsu Semiconductor Limited, Japan. Different sizes are available; please consult the memory model catalog for the current available list.

Available Consumer FCRAM Memory models:

- mb81eds516545_0.vp : Fujitsu 2M word x 64bit x 4banks organization Consumer FCRAM with output timing No.0
- mb81eds516545_1.vp : Fujitsu 2M word x 64bit x 4banks organization Consumer FCRAM with output timing No.1

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. Features

General Features:

- 2 M word × 64 bit × 4 banks organization
- DDR Burst Read/Write Access Capability
- Unidirectional READ Data Strobe per 2 byte
- Unidirectional WRITE Data Strobe per 2 byte
- Burst Length: 2, 4, 8, 16
- CAS latency: 2, 3, 4
- Clock Stop capability during idle periods
- Auto Precharge option for each burst access
- Auto Refresh and Self Refresh Modes
- Deep Power Down Mode
- Multi Bank Active (MACT)
- Multi Bank Precharge (MPRE)
- Background Refresh (BREF)
- Additional RDQS Toggle (ART)

4. Device Supported

The current models support for all the Consumer FCRAM families of Fujitsu. Please consult the appropriate vendor site for details on the parts they offer.

***Note:** Please refer to the FCRAM specification for additional details about I/O and timing.*

5. Model Block Diagram

5.1 Interface Diagram

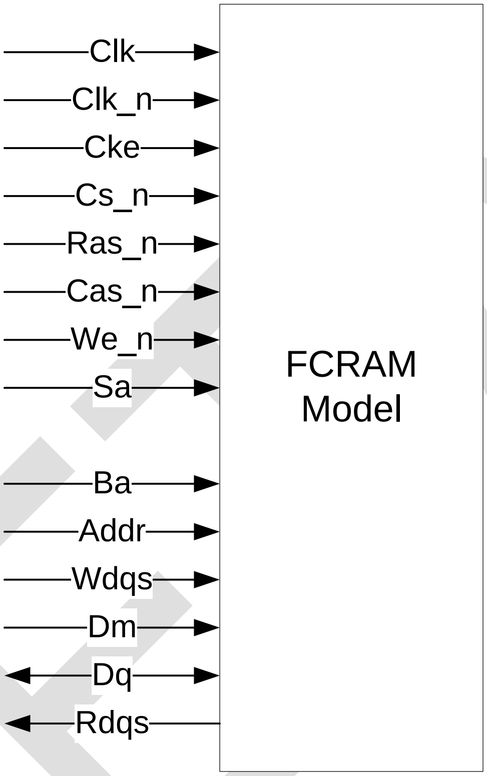


Figure 1. FCRAM Model Interface

Signal name

Signal	Description	I/O
Clk, Clk_n	Differential Clock	Input
Cke	Clock Enable	Input
Cs_n	Chip Select	Input
Ras_n	Row Address Strobe	Input
Cas_n	Column Address Strobe	Input
We_n	Write Enable	Input
Sa	Select Area Enable	Input
Ba	Bank Address Inputs	Input
Addr	Address Inputs	Input
Wdqs	Write Data Strobe	Input
Dm	Input Data Mask Enable	Input
Dq	Bus Data Input/Output	Inout
Rdqs	Read Data Strobe	Output

5.2 Connection Diagram

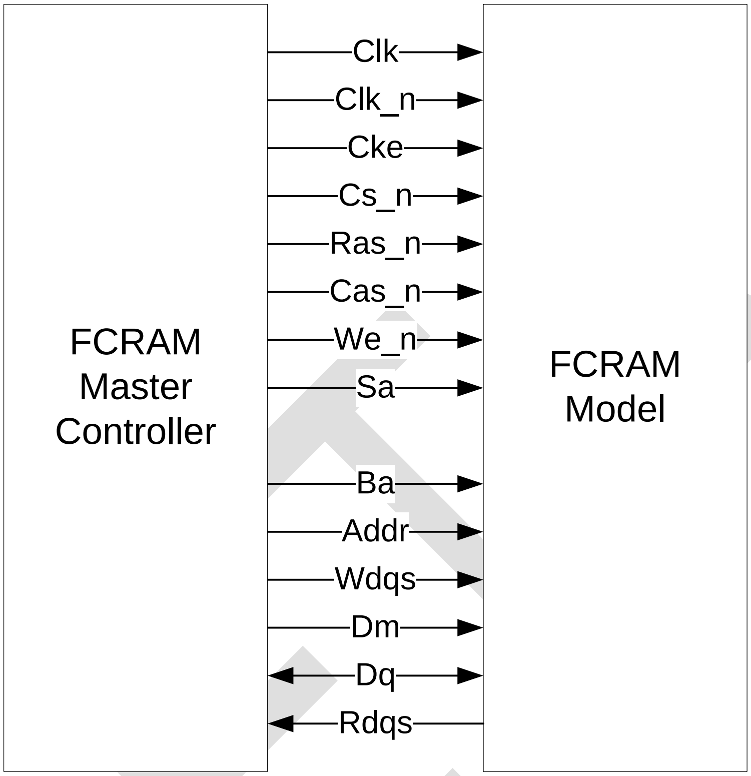


Figure 2. FCRAM Model Connection Diagram

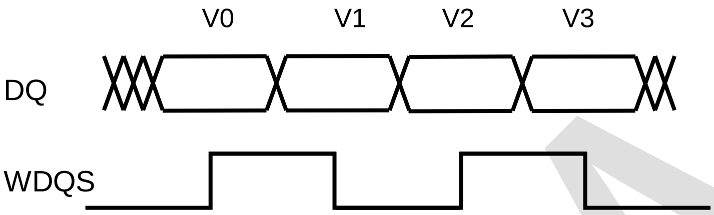
6. Limitations

- Initial Content of the FCRAM Memory
  The initial value of the FCRAM memory array is all 0's after download the model into the Palladium. If you want to use your own initial values, please import them manually using tools supplied by IXE before running.

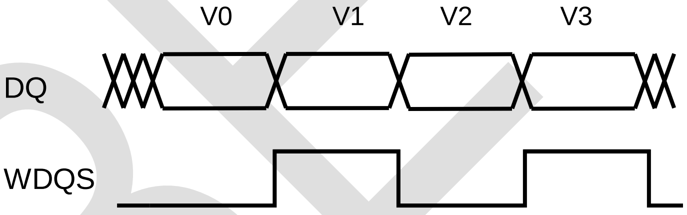
- Writing timing
  There are two kind of writing timing are supported by the FCRAM model and you can chose anyone which you want, but you should avoid that WRITE command can influence each other. For more information, please refer to the FCRAM spec.

7. Handling W/R DQS in Palladium Memory Models

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

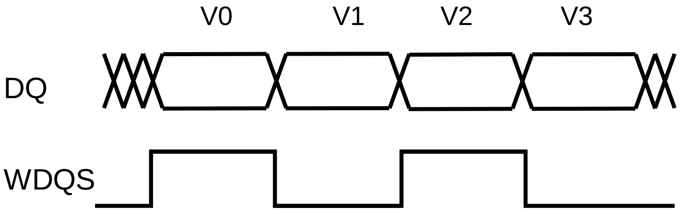


For DDR models provided by Cadence for Palladium, if the design drives DQ and WDQS signals with the above timing, the DDR memory will behave correctly. However, to obtain this timing in Palladium, the fastest design clock must toggle twice as frequently as the WDQS signal. If this faster clock is not 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 WDQS edge at the end of the corresponding DQ valid period (rather than the middle), as in the following diagram:



Note that the first WDQS 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 RDQS with the first RDQS 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-WDQS 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 WDQS will provide one-half FCLK delay, so that each delayed-WDQS edge is at the end of the corresponding data valid period.

To delay the WDQS signal, a commonly used approach is to create a special pad cell for WDQS, 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
```

8. Compile and Emulation

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

```
ixcom -ua +dut+mb81eds516545_0 \
    ./mb81eds516545_0.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 mb81eds516545_0.vp
hdlImport -full -2001 -l qtref
hdlOutputFile -add -f verilog mb81eds516545_0.vg
hdlSynthesize -memory -keepVhdlCase -keepRtlSymbol –keepAllFlipFlop
mb81eds516545_0
……

2)
vavlog mb81eds516545_0.vp

vaelab -keepRtlSymbol -keepAllFlipFlop -outputVlog mb81eds516545_0.vg
mb81eds516545_0
……
```

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.

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.

9. Model Emulation Notes

Some things need you to keep in mind when running emulation with the FCRAM model in Palladium.

- The FCRAM models are delivered as protected Verilog netlists with the top-level declaration unprotected and the memory array unprotected.
  - The main data array is called \ *.memcore*. So the addressing range of the data array is as same as of the actual part. The array simply use an identical combinational address {BA, RA, CA} to access each memory unit. The BA, the RA and the CA represent bank address, row address and column address respectively.

- The top-level module in the FCRAM modules is called "mb81eds516545_0" or "mb81eds516545_1" as needed. To use the model in your design, please instantiate the top-level module and map its ports to your actual wires. For more details of the top-level declaration, please open the netlist with a text viewer and search the string "module" to find out its exact name. Below is the port declaration of the top-level model of mb81eds516545_0:

```
module mb81eds516545_0
 (
        Clk,
        Clk_n,
        Cke,
        Cs_n,
        Ras_n,
        Cas_n,
        We_n,
        Sa,
        Ba,
        Addr,
        Wdqs,
        Dm,
        Dq,
        Rdqs
 );
```

- The initial content of FCRAM memory is all '0' at start-up. The content of FCRAM can be initialized to in Memory tab of questDebug GUI or by issuing commands in the questQel console as below:

```
QEL> memory -set {<dspath>.\ .memcore}
QEL> memory -load %pd_raw2 {<dspath>.\ .memcore} -file meminit.raw2
QEL> memory -dump %pd_raw2 {<dspath>.\ .memcore} -file meminit.raw2
```

To check the full path and exact name of the memory array, please goto to the Memory tab in questDebug, or invoke the memory command in the questQel console. Below is an example to get the memory path in the design.

```
QEL> memory
```

10. Revision History

The following table shows the revision history for this document

Date	Version	Revision
Apr 2012	1.0	Initial release
May 2012	1.1	Minor updates
Jun 2012	1.2	Minor updates
July 2014	1.3	Repaired doc property title. Updated legal.
September 2014	1.4	Remove version from UG file name. Update UXE / IXE documentation reference titles.
September 2014	1.5	Added requirement to W/RDQS timing.
March 2015	1.6	Update related publications list.
July 2015	1.7	Update Cadence naming on front page
September 2015	1.8	Add an example for compile
January 2016	1.9	Update for Palladium-Z1 and VXE
July 2016	1.10	Remove hyphen in Palladium naming
January 2018	2.0	Modify header and footer
July 2018	2.1	Update for new utility library