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

Toggle DDR3.0 NAND Flash
Palladium Memory Model
User Guide

Document Version: 1.1

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

Toggle DDR3.0 NAND Flash Palladium Memory Models

1. Introduction

The Cadence Palladium Toggle DDR3.0 NAND Flash Models are based on data sheet specifications and revision levels for the following Toshiba devices:

Toshiba 3D Flash Memory Toggle DDR3.0 with BiCS (Bit Cost Scaling) technology

Part Number	Reference Datasheet	Rev.	Revision Date
Vendor: Toshiba			
TH58LJT0T24BA4C	TH58LJxxT24BAxx_132BGA_D_20171024_0.2.pdf	v0.2	10/24/2017
TH58LJT1T24BA8C	TH58LJxxT24BAxx_132BGA_D_20171024_0.2.pdf	v0.2	10/24/2017
TH58LJT2T24BA8H	TH58LJxxT24BAxx_132BGA_D_20171024_0.2.pdf	v0.2	10/24/2017

Please note that some of these models support Toggle Mode DDR interface only, even though the data sheet mentions Conventional Asynchronous SDR mode also. Please check the ID Definition Tables in the data sheet for more details.

The models are available in several configurations with model sizes to match real devices manufactured by the following vendors: Toshiba.

Currently only a few different sizes are available, 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, may not have documentation. Model may contain defects.	No	Yes	Yes

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

3. Configurations

The following table lists the configurations specified in the data sheet listed above. It is possible that not all configurations are available from all vendors. Please consult the appropriate vendor site for details on the parts they offer.

Model	Density	# of Die	# of CE#	# of R/B#	# of Blocks/Die	Page size (main+spare)	Pages/Block
TH58LJT0T24BA4C*	1024Gb	2	2	2	3916	18336	1152
TH58LJT1T24BA8C*	2048Gb	4	4	4	3916	18336	1152
TH58LJT2T24BA8H*	4096Gb	8	4	4	3916	18336	1152

Notes: TLC = 3 bits/cell
Models TH58LJxxT24BAxx have 2 sets of I/O pins
(I/O pin set: ALE,CLE,DQ,DQS,RE#,WE#,WP#)
Chip, Die and LUN are used interchangeably
Target and CE# are used interchangeably
A target may have one, two, or four dies

* Please contact Cadence emulation support team or MMP product team to arrange for use of these models. These models are not in the MMP release as they require additional permission.

4. Model Parameter Descriptions

The following table provides details on the user adjustable parameters for the Palladium ToggleDDR3 Memory Model. These parameters may be modified when instantiating the model.

User Adjustable Parameters	Default Values	Description
nb	16 or 32	Number of blocks per LUN
INIT_BANNER_ON	1	Debug Display initial banner

The value of parameter nb is passed into the LUN core module's BLK_IN_MEMORY parameter. It specifies the number of blocks per LUN. This parameter can be adjusted when the model is instantiated by assigning a value to the nb parameter. The maximum value should limit the total address bits to 30. Total address bits = column address bits + page address bits + block address bits, as described in the vendor data sheet. For example, 16K page size requires 15 bits (14 bits for main data and 1 bit for spare area), 1536 pages per block adds 11 bits, leaving only 4 bits or 16 blocks.

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

Not User Adjustable Parameter	Default Value	Description
data_bits	8	Width of DQ bus

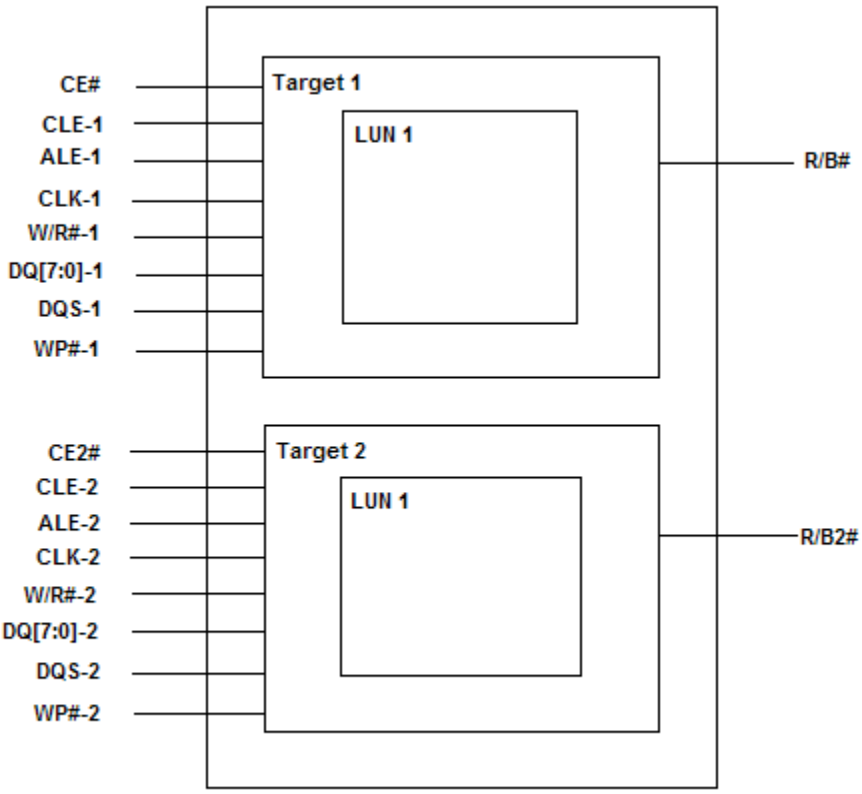
The local parameter data_bits should not be adjusted. It specifies the width for the DQ bus declaration.

5. Model Block Diagram

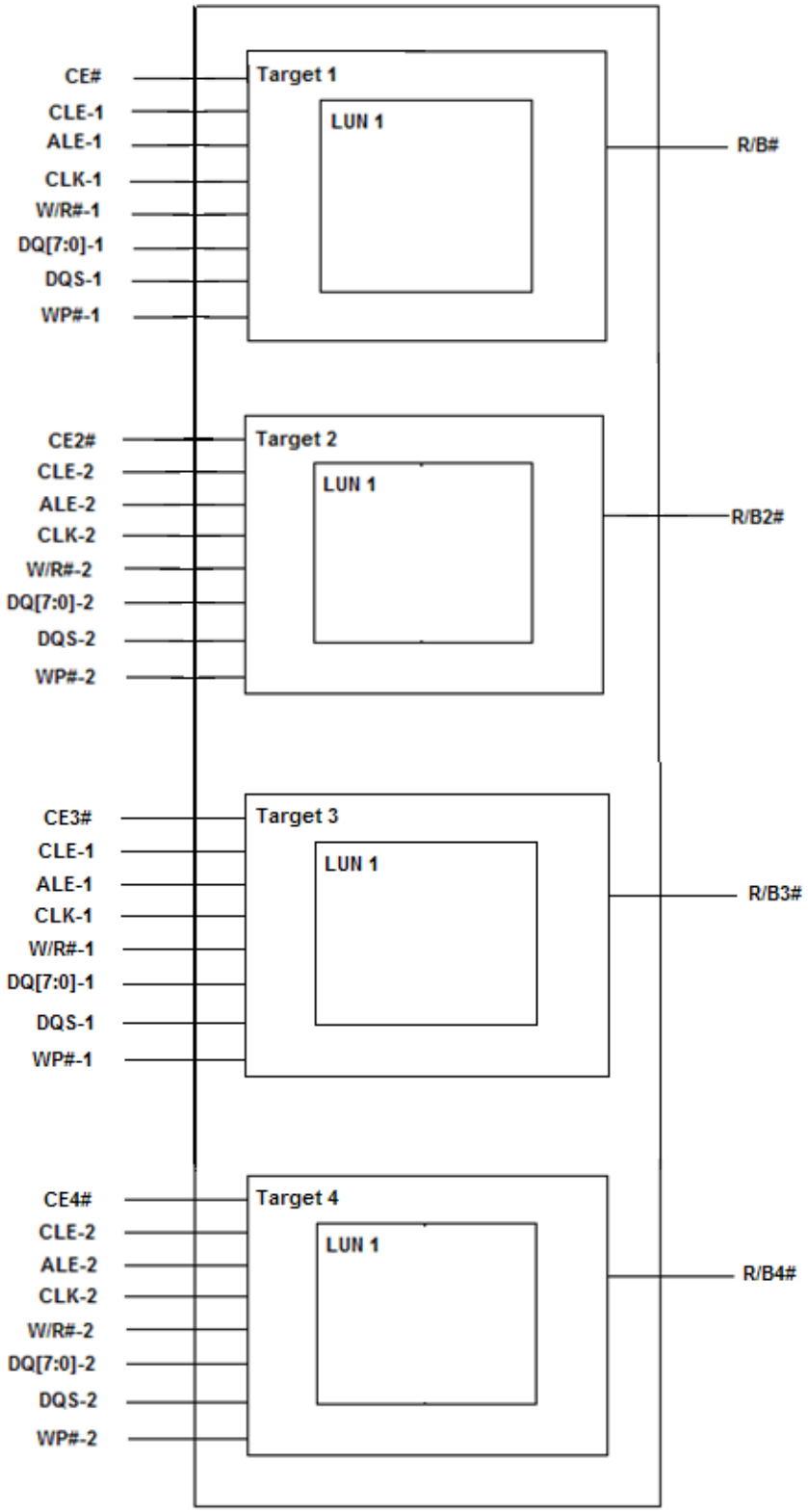
These models are implemented modularly based on the die core, which is instantiated as many times as needed within each model. The user does not need to instantiate the core directly. The user instantiate the model based on the model's I/O pin sets.

Block diagrams of various models are shown below.
For models that have 2 CE#s, Target 1 uses pin set 1, Target 2 uses pin set 2.
For models that have 4 CE#s, Targets 1 and 3 share pin set 1, Targets 2 and 4 share pin set 2.
For models that have 8 CE#s, Targets 1,3,5,7 share pin set 1, Targets 2,4,6,8 share pin set 2.

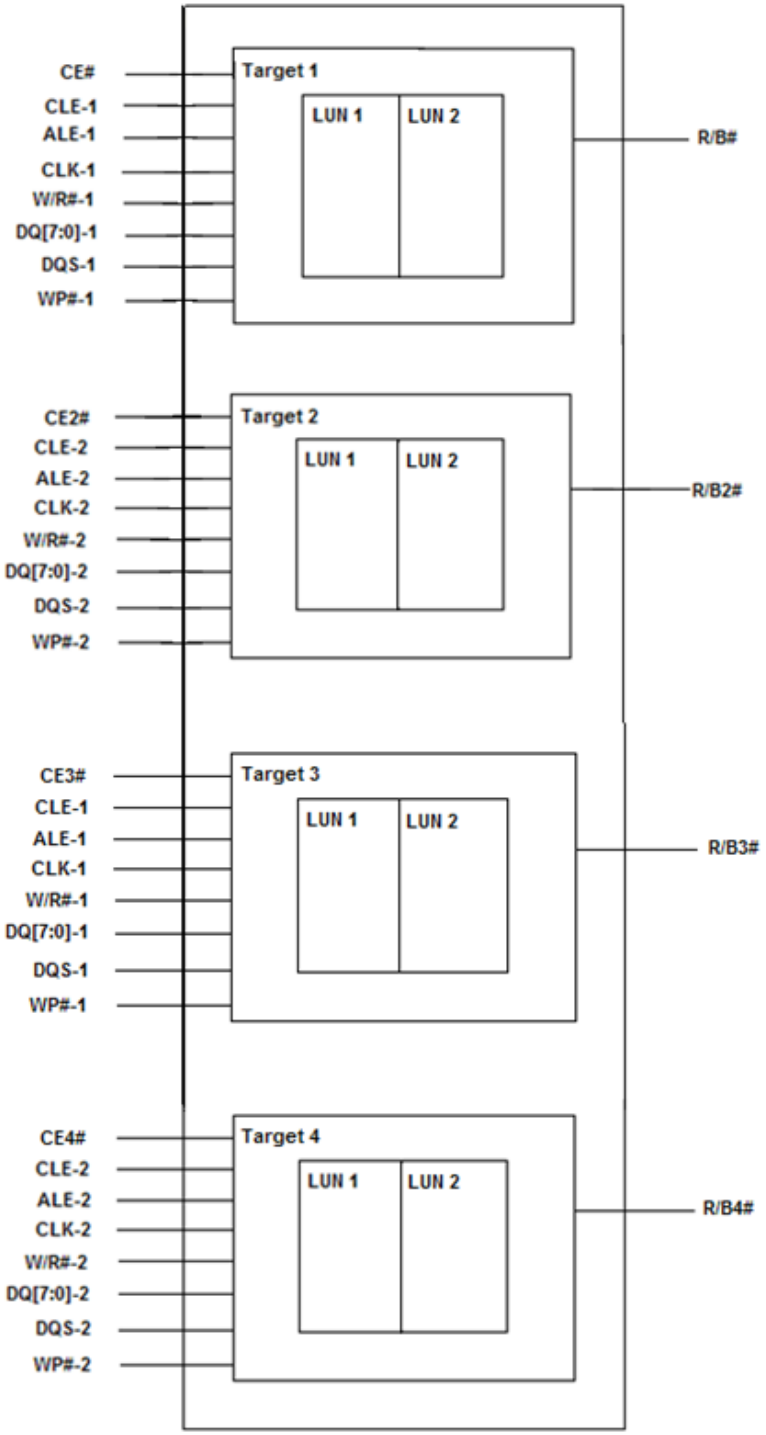
The following block diagram shows the th58ljt0t24ba4c model which has 2 LUNs, 2 CE#s, 2 R/B#s and two sets of I/O pins.



The following block diagram shows the th58ljt1t24ba8c model which has 4 LUNs, 4 CE#s, 2 R/B#s and two sets of I/O pins.



The following block diagram shows the th58ljt2t24ba8h model which has 8 LUNs, 4 CE#s, 4 R/B#s and two sets of I/O pins.



6. Instantiation example

```
th58ljt0t24ba4c u1 (
    .Dq1(io1),
    .Cle1(cle1),
    .Ale1(ale1),
    .Ce_n1(ceb1),
    .We_n1(web1),
    .Re_n1(reb1),
    .Wp_n1(wpb1),
    .Rb_n1(rbb1),
    .Dqs1(dqs1),
    .Dq2(io2),
    .Cle2(cle2),
    .Ale2(ale2),
    .Ce_n2(ceb2),
    .We_n2(web2),
    .Re_n2(reb2),
    .Wp_n2(wpb2),
    .Rb_n2(rbb2),
    .Dqs2(dqs2));
```

7. Address mapping

The array of the NAND Flash model is mapped into the internal memory of the Palladium system. This array is a single two dimensional array. The mapping of LUN, block, page and column addresses to the internal model array is as follows:

ARRAY_ADDR = {LA, BA, PA, CA}

This information is required if the memory needs to be preloaded with user data. For models with only 1 die, LA should be set to 0. To preload 3D models, each word in memory is 32 bits and the byte mapping to MSB, CSB, and LSB pages is as follows:

MEMORY_WORD[31:8] = {MSB[7:0],CSB[7:0],LSB[7:0]}

MEMORY_WORD[7:0] bits are not used in 3D models.

Here is the addressing cycle table for TLC models.

Address Cycle Table for TLC Array

Cycle	I/O 0	I/O 1	I/O 2	I/O 3	I/O 4	I/O 5	I/O 6	I/O 7
First	A0	A1	A2	A3	A4	A5	A6	A7
Second	A8	A9	A10	A11	A12	A13	A14	Low
Third	A15	A16	A17	A18	A19	A20	A21	A22
Fourth	A23	A24	A25	A26	A27	A28	A29	A30
Fifth	A31	A32	A33	A34	A35	A36	Low	Low

Notes from data sheet:
A0 ~ A14 = column address, A15 ~ A23 = page address, A24~ A35 = block address, A36 = LUN or Chip address for models with 2 LUNs per target. A37 = LUN address for models with 4 LUNs per target. The page address, block address, and LUN address are collectively called the row address.

When using the Toggle DDR interface, A0 is forced to 0 internally; one data cycle always returns one even byte and one odd byte.

Column addresses 18336 (47a0h) through 32767 (7FFFh) are invalid, out of bounds, do not exist in the device, and cannot be addressed.

A24 is the plane-select bit for devices with 2 planes.
Plane 0: A22 = 0
Plane 1: A22 = 1

A24 and A25 are the plane-select bits for devices with 4 planes:
Plane 0: A25,A24 = 00
Plane 1: A25,A24 = 01
Plane 2: A25,A24 = 10
Plane 3: A25,A24 = 11

A36 is the LUN-select bit. It is present only when two LUNs are shared on the target; otherwise, it should be held LOW.
LUN 0: A36 = 0
LUN 1: A36 = 1

A37 may be an additional LUN-select bit. It is present only when four LUNs are shared on the target; otherwise, it should be held LOW.
LUN 0: A37,A36 = 00
LUN 1: A37,A36 = 01
LUN 2: A37,A36 = 10
LUN 3: A37,A36 = 11

8. Feature Address Definition

Since the feature address is 8 bits there can be up to 256 feature addresses defined – a 256 x 32 array. However addresses 02h, 10h, 30h are the only parameters that can be set by the SET FEATURE (EFh) command, therefore these settings are not supported in the Palladium models. The following table shows Feature Address definitions.

Feature Address Definitions

Address	Description
00h~01h	Reserved
02h	High Speed setting
03h~0fh	Reserved
10h	Driver Strength setting
11h~2fh	Reserved
30h	External Vpp setting
31h ~ FFh	Reserved

9. ID Operations

9.1. READ ID

The READ ID parameters for addresses 00h and 40h have been hardcoded into each model. Therefore user data file is not required.

9.2. READ Device ID Table

The data for the device id table is provided in the <model_name>.dat file. This data file should be preloaded into all LUNs in the model if the user wants to read Device ID information from the model. The instance names are L<CE><LUN>. Using the th58ljt2t24ba8h model as an example, there are 4 CEs and 2 LUNs per CE. The path to each LUN's device id table or parameter page can be viewed in dbFiles/xcva_top_et5mpart file.

```
memory –load %readmemh <path.to.model.inst>.L11.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L12.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L21.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L22.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L31.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L32.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L41.param_page –file <model>.dat
memory –load %readmemh <path.to.model.inst>.L42.param_page –file <model>.dat
```

10. Runtime Memory Commands

The main data array may be initialized to 0xff by the memory -set command, or preloaded with a data file using the memory -load command as shown above. The path to the data array can be viewed in dbFiles/xcva_top_et5mpart file.

```
memory -set <path_to_model_inst>.L11.mem_array
memory -set <path_to_model_inst>.L12.mem_array
memory -set <path_to_model_inst>.L21.mem_array
memory -set <path_to_model_inst>.L22.mem_array
memory -set <path_to_model_inst>.L31.mem_array
memory -set <path_to_model_inst>.L32.mem_array
memory -set <path_to_model_inst>.L41.mem_array
memory -set <path_to_model_inst>.L42.mem_array
```

11. Features

The following table shows a list of features and support status for the ToggleDDR3 model.

Feature Support

FEATURE	SUPPORT	NOTE
COMMANDS		
Page Read	Yes	
Last Page Cache Read	Yes	
Random Cache Read	Yes	
Page Program	Yes	
Cache Program	Yes	
Block Erase	Yes	
Random Data Input	Yes	
Random Data Output	Yes	
Set Feature	Partial	See Feature Address Definition section
Get Feature	Yes	
Read ID	Yes	
Read Status	Yes	
Reset	Yes	
Reset LUN	Yes	
Read LUN Status	Yes	
Device Identification Table Read	Yes	
LSB Page Select	Yes	3D models
CSB Page Select	Yes	3D models
MSB Page Select	Yes	3D models
Full Sequence Page Program	Yes	3D models
SPECIAL OPERATIONS		
Multi-Plane Operations	Yes	
Multi-LUN or Interleaving Operations	Yes	
ODT (On die termination)	No	
OTP (One-time programmable operations)	No	
Error Management	No	Spare area is available
SDR Interface	No	

The following table shows the Basic Command Sets as described in Toshiba data sheet.

Basic Command Sets

Function	Primary or secondary	1st Set	Address Cycles	2nd Set	Acceptable while Accessed LUN is busy	Acceptable while Other LUNs are busy
LSB Page Select	-	01h	-	-		
CSB Page Select	-	02h	-	-		
MSB Page Select	-	03h	-	-		
Full Sequence Page Program	N/A	80h-1Ah	5	80h-10h		Y
Page Read	Primary	00h	5	30h		Y
Read Start for Last Page Cache Read	Primary	3Fh	-	-		Y
Random Cache Read	Primary	00h	5	31h		Y
Page Program	Primary	80h	5	10h		Y
Cache Program	Primary	80h	5	15h		Y
Block Erase	Primary	60h	3	D0h		Y
Random Data Input (1)	Primary	85h	2	-		Y
Random Data Output (1)	Primary	05h	5	E0h		Y
Set Feature	Primary	EFh	1	-		
Get Feature	Primary	EEh	1	-		
Read ID	Primary	90h	1	-		Y
Read Status	Primary	70h	-	-	Y	Y
Reset	Primary	FFh	-	-	Y	Y
Reset LUN	-	FAh	3	-	Y	Y

Notes from data sheet:
1. Random Data Input/Output can be executed in a page.

The following table shows the Extended Command Sets as described in Toshiba data sheet.

Extended Command Sets

Function	Primary or secondary	1st Set	Address Cycles for 1st Set	2nd Set	Address Cycles for 2nd Set
Multi-Plane Page Read/ Multi-Plane Page Cache Read	Primary	00h-32h	5	00h-30h	5
Multi-Plane Random Cache Read	Primary	00h-32h	5	00h-31h	5
Multi-Plane Random Data Output (1)	Primary	05h-E0h	5	O5h-E0h	5
Multi-Plane Full Sequence Program	N/A	80h-11h	5	80h-1Ah or 80h-10h	5
Multi-Plane Cache Full Sequence Program	Primary	80h-11h (2)	5	80h-1Ah or 80h-15h or 80h-10h	5
Multi-Plane Block Erase	Primary	60h	3	60h-D0h	-
Device Identification Table Read	Primary	ECh-	1	-	-
Read status enhanced	Primary	78h-	3	-	-
Read LUN#0 Status	Secondary	F1h	-	-	-
Read LUN#1 Status	Secondary	F2h	-	-	-

Notes from data sheet:
1. Multi-Plane Random Data out must be used after Multi-Plane Page Read or Multi-Plane Cache Read operation.
2. Any command between 11h and 80h/85h is prohibited except 70h/78h/F1h/F2h and FFh.

12. MMP and ECC (Error Correcting Code)

MMP models do not support Error Correcting Code (ECC) functionality. ECC functions, if they are present in a memory device, are typically found in the NAND and DDRx families. The MMP product does not have any plans to provide such functions in the models. MMP models are provided as system level emulation models and not as verification IP. The below sections discuss work-arounds that enable the user to deal with some ECC scenarios. Note that ECC means different things to different device families.

12.1. NAND FLASH and ECC (Error Correcting Code)

MMP NAND models do not support Error Correcting Code (ECC) functionality. There will not be an ECC error in a Palladium MMP flash model; the data stored in a MMP model should not need to be corrected because the model does not degrade over time like the real device. The data returned is always correct. The paragraphs below provide details about host ECC in relation to MMP NAND Flash.

For NAND Flash devices, ECC means that the internal engine in the Flash device calculates the ECC when programming and writes the resulting value into the spare array. The low level details of this operation are in the device specification. The Flash then re-calculates the ECC on reads and compares with the value stored in the spare array. If non-equivalence is found, bit error is indicated, and the device corrects and/or flags an error. There are several cases:

- If the controller relies on ECC generation internal to the Flash device, then the model needs to do nothing. This has worked for all users so far.
    - To support this scenario, model parameters can be modified to indicate that ECC is enabled. The controller is then happy. NOTE: the model will NOT actually do the ECC calculation.

- If the controller uses its own ECC and manually writes to the spare array in the device, then again the MMP model does not need to do anything.
    - There is a spare area is implemented in the NAND model for the host to store ECCs. This spare area allows the host to do data correction.

- If the controller relies on ECC generation internal to the memory device AND the controller reads and examines the spare calculation itself, then the MMP model will not work.
    - There is no MMP plan to enhance NAND FLASH MMP models to support this case. It is a large effort.

Occasionally, an issue may be seen due to the parameter page setting for the available number of bits of ECC correction. According to the ONFI standard this setting is handled by byte 112 of the parameter page. See the figure below for an example entry from the standard. Problems may occur with some controllers when byte 112 of the parameter page is set to the value ‘0’. If the controller requires some positive value for the *Number of bits ECC correctability*, then the user may need to change the setting to a value of ‘1’.

Table 13: Parameter Page Data Structure (Continued)

Byte	Description	Device	Values
112	Number of bits ECC correctability	–	0Ch
113	Number of interleaved address bits	–	01h

13. Initialization Sequence

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

1. The RESET (FFh) command must be the first command issued to all targets (CE#s). The RESET busy time can be monitored by polling R/B#.
2. When R/B# is high the model is now initialized and ready for normal operation.

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.

14. Model Size

To reduce memory utilization each LUN has only 32 blocks but the actual device has 3916 blocks. If larger size is needed please contact Customer Support.

15. Limitations

- Set Feature and Get Feature commands are partially supported.
- Model does not check illegal sequence of command cycles.
- Model does not check for attempts to program a bit to 1, user should make sure the block is erased before program.
- OTP is not supported.
- ECC is not supported.
- Vref, /DQS, RE, DQS cycle latency, Driver Strength, Ext Vpp , are not supported.
- Conventional Asynchronous SDR mode is not supported in some models.

16. Compile and Emulation

The memory models are currently provided in one format: an encrypted RTL file(s) (*.vp) that targets use in either the IXCOM flow or in the ICE flow. The encrypted RTL (*.vp) file(s) must be synthesized along with other design code prior to acceleration / emulation.

An example of the command for compilation (including synthesis) of this model in IXCOM flow is shown below:

```
ixcom -64bit +sv -ua +dut+th58ljt0t24ba4c \
../tb.v \
```

```
../src/ttd3_64.vp \
../src/th58ljt0t24ba4c.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
```

Note that +sv switch is needed.

ICE flow synthesis commands:

```
vavlog ../src/ttd3_64.vp ../src/th58ljt0t24ba4c.vp

vaelab –keepRtlSymbol –keepAllFlipFlop –outputVlog th58ljt0t24ba4c.vgp
th58ljt0t24ba4c
```

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.

16.1. Model file list

ttd3_64.vp - LUN or die module that is instantiated by each Toshiba model.
<model_name>.vp - model wrapper that instantiates one or more LUNs.

17. Model Clocking

fclk – Fastest clock in design. It is used to increment the address while data is copied between page register and data array, and during block erase operation. Typically fclk should be at least 4x Dqs frequency.

18. Extendable Busy Timing

As stated above fclk (fastest clock in design) is used to increment the address during read, program and erase operations, at two cycles per address. The time to program or read a page depends on page size or number of addresses per page, and the time to erase a block depends on block size. Depending on fclk frequency it is possible that the model may complete the operation too quickly. The user may extend the program and erase busy time by defining one or more of the following macros.

Macro name	Default value	Description
MMP_ADD_PROG_TIME	0	Number of fclks to extend program busy time
MMP_ADD_ERAS_TIME	0	Number of fclks to extend erase busy time
MMP_ADD_READ_TIME	0	Number of fclks to extend read busy time

Here are example calculations for these macros based on the following timing parameters.

Description	Parameter	Typical	Max.	Unit
Programming Time	t_{PROG}	4	12	ms
Block Erasing Time	t_{BERASE}	12	30	ms

Example 1 for Programming Time:
If fclk frequency is 1 GHz and page size is 18336 addresses, program operation completes in:

2 cycles per address x 18336 addresses x 1 ns = 0.000036672 seconds = 0.036672 ms

Calculating the difference between the typical, or desired, programming time and the actual programming time yields 4 .0 ms - 0.036672 ms = 3.963328 ms = 3963328 ns

Since the fclk period is 1 ns, using 3963328 ns / 1ns = 3963328 fclks can extend the programming time to 4ms by setting MMP_ADD_PROG_TIME to 3963328 fclks.

In a SSD based design there is likely to be a PCIE interface that would have a 1GHz fastest clock that would map to Palladium fclk. For the case where only the NAND flash is present, the interface speed might be, for example, 266 MHz. If this is the only clock and it can be mapped directly to the Palladium fclk the calculation above can be modified as below.

2 cycles per address x 18336 addresses x 3.75939 ns = 0.000137865 seconds = 0.137865 ms

Calculating the difference between the typical, or desired, programming time and the actual programming time yields 4 .0 ms - 0.137865 ms = 3.862135 ms = 3862135 ns

Since the fclk period is 3.75939 ns, using 3862135 ns / 3.75939 ns = 1027330 fclks can extend the programming time to 4ms by setting MMP_ADD_PROG_TIME to 1027330 fclks.

Example 2 for Erase Time:

If the block size is 128 pages, the erase operation completes in

2 cycles per address x 128 pages x 18336 addresses x 1 ns = 4.694016 ms

Calculating the difference between the typical, or desired, erase time and the actual erase time yields 12 ms – 4.694016 ms = 7.305984 ms = 7,305,984 ns

Since the fclk period is 1 ns, using 7305984 ns / 1ns = 7305984 fclks can extend the block erasing time to 12ms by setting MMP_ADD_ERAS_TIME to 7305984 fclks.

19. Debugging

The Toggle DDR 3.0 model has several debugging options techniques and tips that may assist the user in isolating a problem.

- For issues that may not be ToggleDDR3 specific please review the *Memory Model Portfolio FAQ for All Models User Guide.*

- **Golden waveform:** A waveform showing basic startup sequence - reset, read id, program page, read page, is available in the release under the toggleddr3/toshiba/golden_waveform directory. It may be useful to have a look at this waveform when using the model for the first time :

    - reset
    - get feature
    - set feature
    - full sequence program interleaving on ce0 and ce2
    - cache read interleaving on ce0 and ce2
    - multiplane full sequence program interleaving on ce0 and ce2
    - multiplane cache read interleaving on ce0 and ce2
    - multiplane full sequence program interleaving on ce1 and ce3
    - multiplane cache read interleaving on ce1 and ce3

- **Key Signals:** Some key signals to observe when debugging the model:

    - Ebar – Chip Enable should be low when the selected LUN's DQ bus is active
    - Wpbar – Write Protect should be high during normal operations
    - Rbbar – Ready/Busy, the host should wait for model to be ready before issuing new command
    - Dqs – Data Strobe should be active in DDR mode during data IO cycles
    - Cl – Command cycle
    - Al – Address cycle
    - Wbar – Command and Address strobe
    - Rbar – Data output
    - fclk – Fastest clock in design

- **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.

20. Revision History

The following table shows the revision history for this document

Date	Version	Revision
June 2018	1.0	Initial Release
July 2018	1.1	Update for new utility library