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

ONFI 4.0 Flash
Palladium Memory Model
User Guide

Document Version: 2.6

Document Date: July 2018

Contents

GENERAL INFORMATION **4**

1.1 RELATED PUBLICATIONS 4

ONFI 4.0 FLASH PALLADIUM MEMORY MODELS **5**

1. INTRODUCTION 5
2. MODEL RELEASE LEVELS 6
3. CONFIGURATIONS 7
4. MODEL PARAMETER DESCRIPTIONS 8
5. MODEL BLOCK DIAGRAM 9

5.1. I/O Signal Description *12*

6. ADDRESS MAPPING 12
7. FEATURE ADDRESS DEFINITIONS 15
8. ID OPERATIONS 16

8.1. READ ID *16*
8.2. READ PARAMETER PAGE *16*
8.3. READ UNIQUE ID *16*

9. FEATURES 17
10. MMP AND ECC (ERROR CORRECTING CODE) 20

10.1. NAND FLASH and ECC (Error Correcting Code) *20*

11. INITIALIZATION SEQUENCE 21
12. MODEL SIZE 23
13. LARGE MEMORY SUPPORT 24

12.1 Preloading Multiple Arrays *24*

14. LIMITATIONS 25
15. TIMING PARAMETER TRHOH 25
16. COMPILE AND EMULATION 26

16.1. Model file list *27*

17. MODEL CLOCKING 27

18. EXTENDABLE BUSY TIMING 27
19. DEBUGGING 29
20. REVISION HISTORY 31

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

ONFI 4.0 Flash Palladium Memory Models

1. Introduction

The Cadence Palladium ONFI 4.x-compliant Flash Models are based on data sheet specifications of the following Micron devices:

Size / Spec. level	Reference Datasheet	Revision	Revision Date
MT29F Flash memory with TLC technology			
384 Gb per die, ONFI 4.0 compliant	B0KB_384Gb_768Gb_1HTb_3Tb_6Tb_Sync_NAND_Datasheet.pdf	Rev. A	2/12/2015
512 Gb per die, ONFI 4.0 compliant	B17A_FortisFlash_512Gb_1Tb_4Tb_8Tb_Async_Sync_NAND_Datasheet_20170706.pdf	Rev. D	07/06/17
256 Gb per die, ONFI 4.0 compliant	B16A_Fortis_Flash_256Gb_512Gb_1Tb_Async_Sync_NAND_Datasheet_20170906.pdf	Rev. I	09/06/17
MT29F Flash memory with MLC technology			
256Gb per die, ONFI 4.0 compliant	L06B_256Gb_512Gb_1Tb_Aync_Sync_NAND_Datasheet.pdf	Rev. H	04/01/16
256Gb per die, ONFI 4.0 compliant	L06B_Fortis_Flash_256Gb_512Gb_1Tb_2Tb_4Tb_Async_Sync_NAND_Datasheet.pdf	Rev. F	04/01/16

These models support asynchronous, synchronous NV-DDR, and synchronous NV-DDR2 I/O interfaces.

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

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 and 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 tables list the configurations specified in the data sheets 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 (ONFI 4.0)	Density	Level	# of Die	# of CE#	# of R/B#	I/O	Interface
MT29F256G08CBCBB	256Gb	MLC	1	1	1	Common	Sync/Async
MT29F512G08CECBB	512Gb	MLC	2	2	2	Sep – 2 CH	Sync/Async
MT29F512G08CFCBB	512Gb	MLC	2	2	2	Common	Sync/Async
MT29F1T08CMCBB	1Tb	MLC	4	4	4	Sep – 2 CH	Sync/Async
MT29F256G08CBHBB	256Gb	MLC	1	1	1	Common	Sync/Async
MT29F512G08CEHBB	512Gb	MLC	2	2	2	Sep – 2 CH	Sync/Async
MT29F1T08CMHBB	1Tb	MLC	4	4	4	Sep – 2 CH	Sync/Async
MT29F2T08CUHBB	2Tb	MLC	8	4	4	Sep – 2 CH	Sync/Async
MT29F4T08CTHBB	4Tb	MLC	16	4	4	Sep – 2 CH	Sync/Async
MT29F384G08EBCBB	384Gb	TLC	1	1	1	Common	Sync/Async
MT29F768G08EECBB	768Gb	TLC	2	2	2	Sep – 2 CH	Sync/Async
MT29F1HT08EKCBB	1536Gb	TLC	4	2	2	Sep – 2 CH	Sync/Async
MT29F1HT08EMCBB	1536Gb	TLC	4	4	4	Sep – 2 CH	Sync/Async
MT29F3T08EUCBB	3072Gb	TLC	8	4	4	Sep – 2 CH	Sync/Async
MT29F6T08ETCBB	6144Gb	TLC	16	8	4	Sep – 2 CH	Sync/Async
MT29F512G08EBHAF	512Gb	TLC[1]	1	1	1	Common	Sync/Async
MT29F1T08EEHAF	1Tb	TLC[1]	2	2	2	Sep – 2 CH	Sync/Async
MT29F2T08EMHAF	2Tb	TLC[1]	4	4	4	Sep – 2 CH	Sync/Async
MT29F4T08EUHAF	4Tb	TLC[1]	8	4	4	Sep – 2 CH	Sync/Async
MT29F8T08EWHAF	8Tb	TLC[1]	16	4	4	Sep – 2 CH	Sync/Async
MT29F256G08EBHAF	256Gb	TLC[2]	1	1	1	Common	Sync/Async

MT29F512G08EEHAF	**512Gb**	**TLC[2]**	**2**	**2**	**2**	**Sep – 2 CH**	**Sync/Async**
MT29F1T08EMHAF	**1Tb**	**TLC[2]**	**4**	**4**	**4**	**Sep – 2 CH**	**Sync/Async**

Notes: TLC = 3bits/cell, 1536 pages per block, 2192 blocks per die
TLC[1] = 3bits/cell, 2304 pages per block, 2016 blocks per die (B17A data sheet)
TLC[2] = 3bits/cell, 2304 pages per block, 1008 blocks per die (B16A data sheet)
MLC = 2bits/cell, 1024 pages per block, 2192 blocks per die
Separate I/O = 2 sets of pins
Separate I/O 2 CH = 2 sets of pins
Separate I/O 4 CH = 4 sets of pins
Common I/O = 1 set of pins
(pin set: ALE,CLE,DQ,DQS,RE#,WE#,WP#)
Die and LUN are used interchangeably
Target and CE# are used interchangeably
ONFI 4.0 models have 16KB (16384+2208 bytes) per page
TLC models support SLC mode via Set Feature address 91h

.

4. Model Parameter Descriptions

The following table provides details on the user adjustable parameters for the Palladium ONFI 4.x Memory Model. These parameters may be modified when instantiating the model.

User Adjustable Parameter	Default Value	Description
INIT_BANNER_ON	1	Debug Display Banner
nb	16	Number of blocks per LUN

The Debug Display Banner is turned on by default. If there are multiple instances of the model the user may want to turn it off by passing in a value of 0 for all except the first instance so that the banner is displayed only once. Please refer to Debug Display user guide for more information on Debug Display feature.

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. In order to support more than 16 blocks large memory support needs to be enabled. Please see large memory support section for more details.

The following table provides some information about exposed 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.

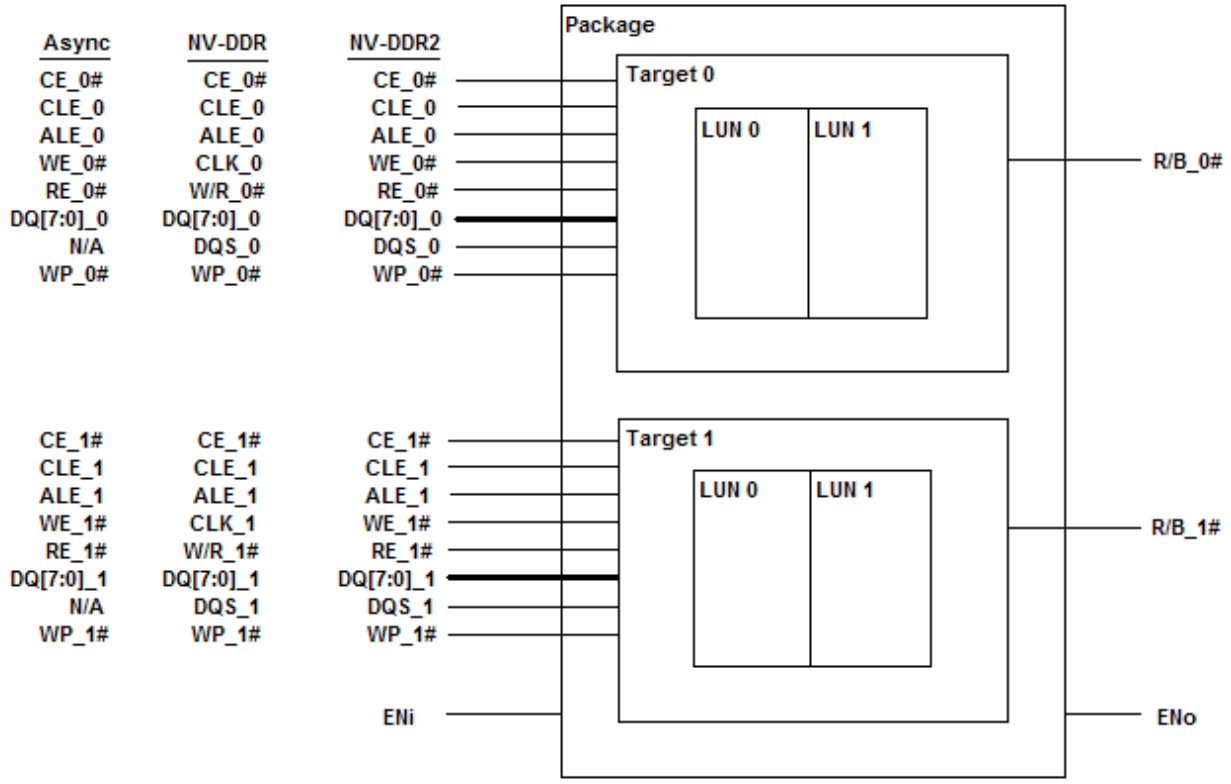
User Not Adjustable localparam	Default Value	Description
data_bits	8	Width of DQ bus

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

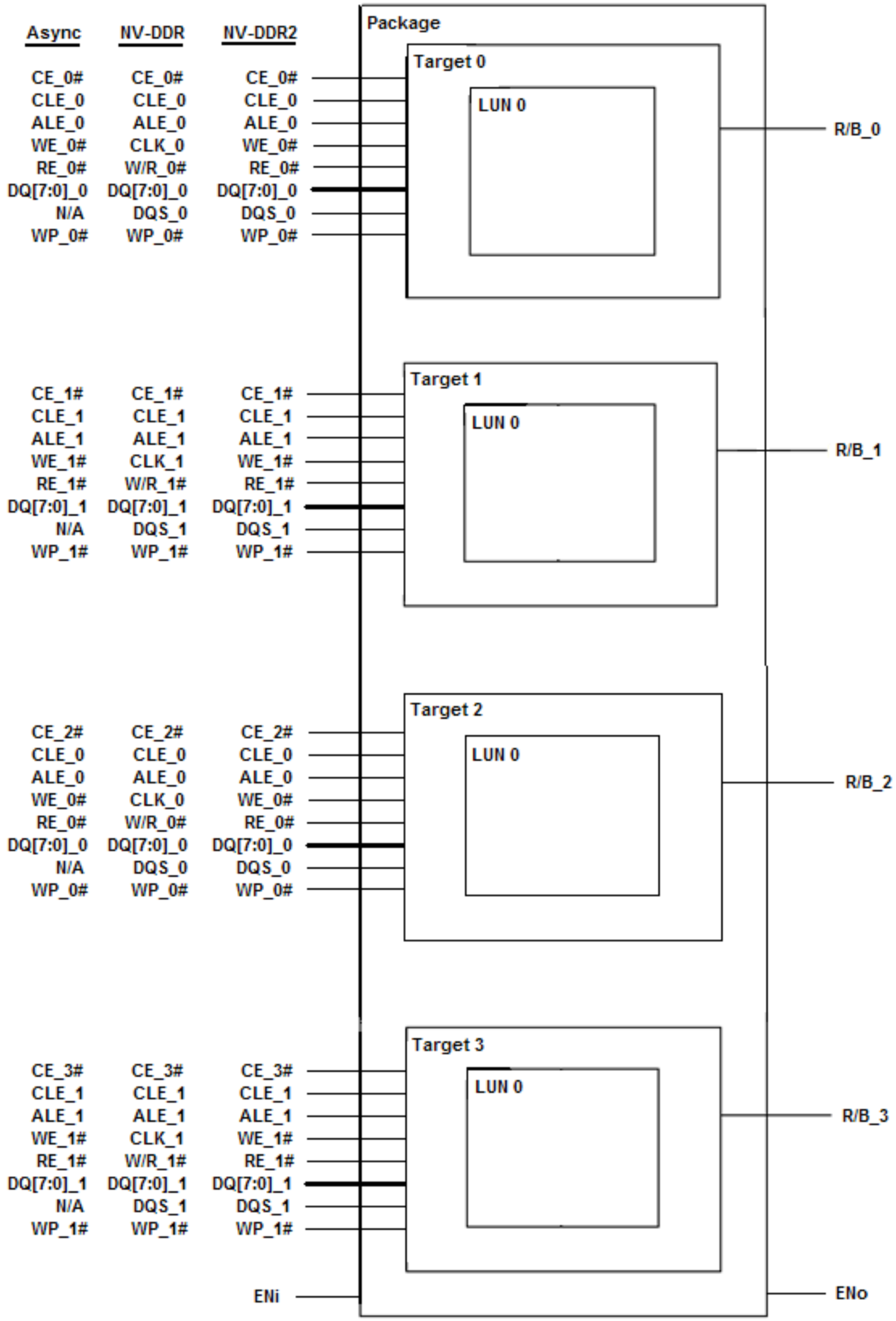
5. Model Block Diagram

These models are implemented modularly based on the LUN core, which is instantiated as many times as needed within each model. The user does not need to instantiate the core directly. The user instantiates the model based on the model's number of dies.

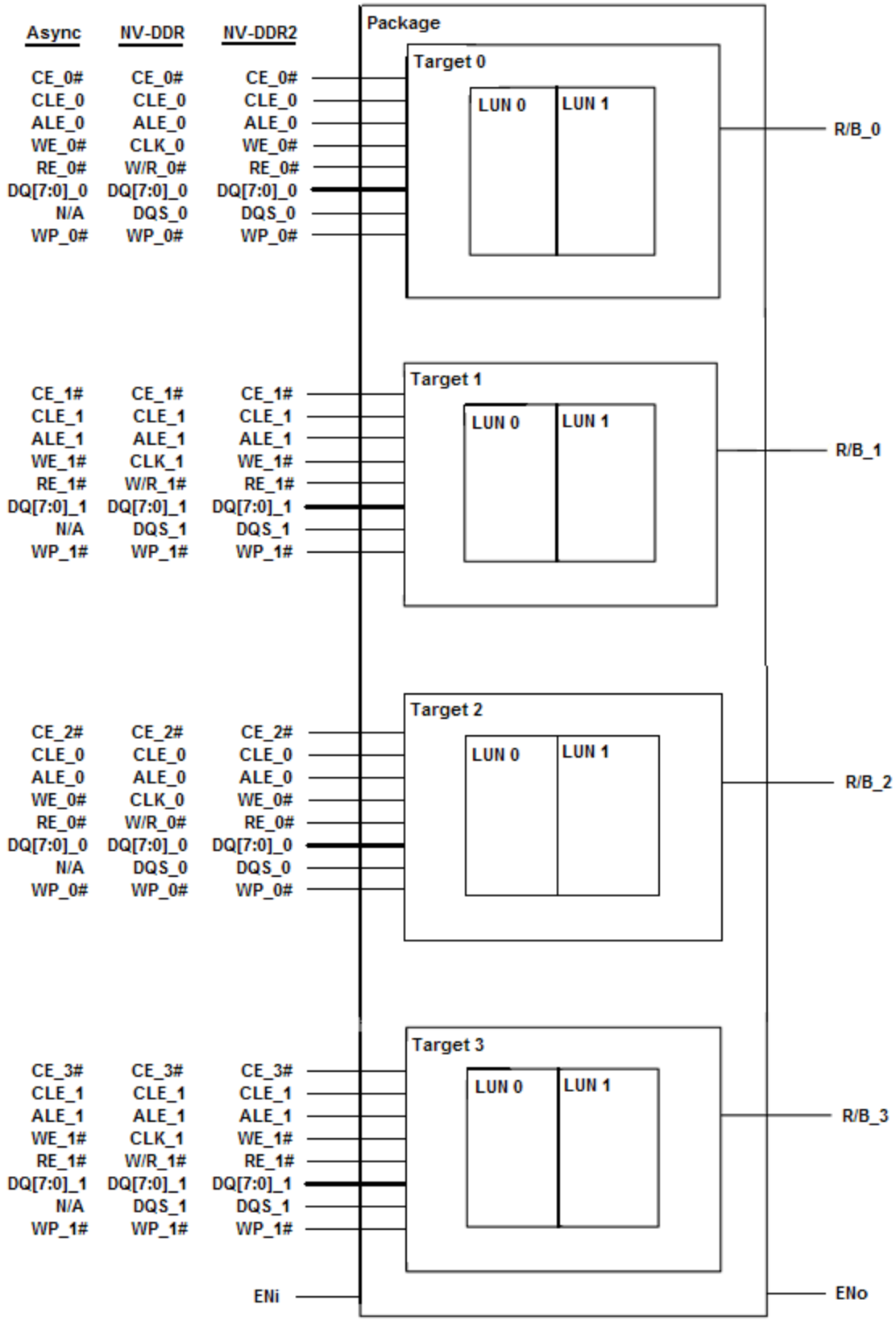
A block diagram of a model that has 4 dies, or LUNs, with 2 CE# is shown below.



The following block diagram shows a model that has 4 LUNs with 4 CE#.



The following block diagram shows a model that has 8 LUNs with 4 CE#.



5.1. I/O Signal Description

Signal Name	Type	Description
CE#	Input	Chip Enable
CLE	Input	Command Latch Enable
ALE	Input	Address Latch Enable
WE#	Input	Write Enable
RE#	Input	Read Enable
DQ	Inout	Data bus
DQS	Inout	Data Strobe
WP#	Input	Write Protect
R/B#	Output	Ready/Busy
ENi	Input	Enumerate Input
ENo	Output	Enumerate Output

#denotes active low

Note that the optional pin/signal DQS_c (DQS complement) is present but it is not required to be connected.

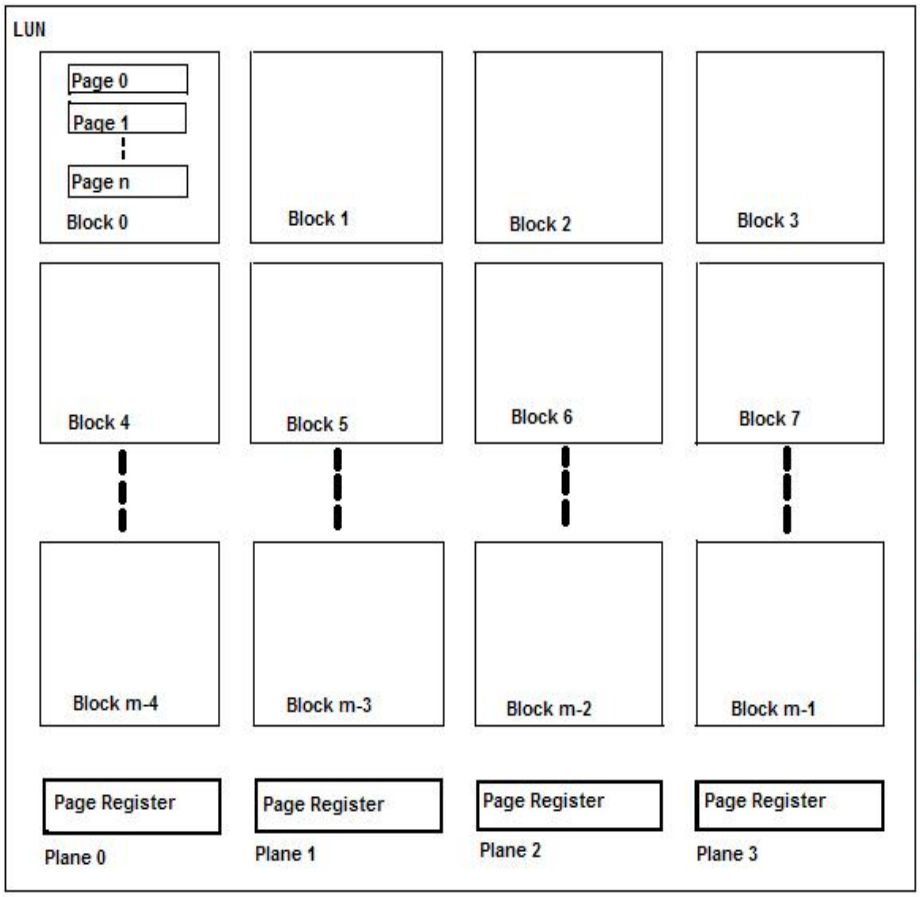
6. Address mapping

The array of the ONFI 4.x 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. Here are the array organization and addressing cycle table for TLC models.

Array Organization for TLC Array



1 page = (16K + 2208 bytes)

1 block = (16K + 2208) bytes x 1536 pages
= (24576K + 3312K) bytes

1 plane = (24576K + 3312K) bytes x 548 blocks
= 119,395Mb

1 LUN = 119,395Mb x 4 planes
= 477,582Mb

Address Cycle Table for MLC Array

Cycle	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Second	LOW	CA14	CA13	CA12	CA11	CA10	CA9	CA8
Third	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Fourth	BA15	BA14	BA13	BA12	BA11	PA10	PA9	PA8
Fifth	LA0	BA22	BA21	BA20	BA19	BA18	BA17	BA16

Notes from Micron data sheet:
CAx = column address, PAx = page address, BAx = block address, LAx = LUN address; the page address, block address, and LUN address are collectively called the row address.

When using the synchronous interface, CA0 is forced to 0 internally; one data cycle always returns one even byte and one odd byte.

For ONFI 4.0 models column addresses 18,592 (48A0h) through 32,767 (7FFFh) are invalid, out of bounds, do not exist in the device, and cannot be addressed.

BA[12] and BA[11] are the plane-select bits:
Plane 0: BA[12] = 0, BA[11] = 0
Plane 1: BA[12] = 0, BA[11] = 1
Plane 2: BA[12] = 1, BA[11] = 0
Plane 3: BA[12] = 1, BA[11] = 1

LA0 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: LA0 = 0
LUN 1: LA0 = 1

TLC models have 144 extended blocks per LUN which requires one additional block address bit BA22, and shifts LA0 up one bit to DQ7 in the fifth address cycle.

7. Feature Address Definitions

In the data sheets there are up to 256 feature addresses defined – a 256 x 32 array. The Palladium models implement the entire array. However, the only features that can be activated are the synchronous data interfaces and volume configuration. Other addresses can be written to and read back using the Set and Get Feature commands but the set features will not be active. For example, timing mode change and Partial Page Read are not supported. The following table shows the feature address definitions as described in the Micron data sheet.

Feature Address	Definition
00h	Reserved
01h	Timing mode and Data interface
02h	NV-DDR2 configuration
03h-0Fh	Reserved
10h	Programmable output drive strength
11h-2Fh	Reserved
30h	Vpp configuration
31h-57h	Reserved
58h	Volume configuration
59h-7Fh	Reserved
80h	Programmable output drive strength
81h	Programmable R/B# pull-down strength
82h-88h	Reserved
89h	Read Retry
8Ah-8Fh	Reserved
90h	Array Operation Mode
91h	SLC mode
92h-F4h	Reserved
F5h	Partial page Read / Express Read
F6h	SleepLite
F7h-FFh	Reserved

8. ID Operations

8.1. READ ID

The READ ID parameters for addresses 00h, 20h and 40h have been hardcoded into each model. Therefore user data file is not required.

8.2. READ PARAMETER PAGE

The data for the ONFI and JEDEC parameter pages are provided in the <model_name><package_code>_param.dat and <model_name><package_code>_jedec_param.dat files. The package_code is two characters. These data files should be preloaded into the model if the user wants to read ONFI and JEDEC information from the model. The LUN instance names are L<CE><LUN>. Using a model with 2 CEs and each CE has 2 LUNs as an example the preload command and path to the parameter pages are as follows:

memory –load %readmemh <path.to.model.inst>.L11.param_page –file mt29f1ht08ekcbbj4_param.dat
memory –load %readmemh <path.to.model.inst>.L11.jedec_param_page –file mt29f1ht08ekcbbj4_jedec_param.dat
memory –load %readmemh <path.to.model.inst>.L12.param_page –file mt29f1ht08ekcbbj4_param.dat
memory –load %readmemh <path.to.model.inst>.L12.jedec_param_page –file mt29f1ht08ekcbbj4_jedec_param.dat
memory –load %readmemh <path.to.model.inst>.L21.param_page –file mt29f1ht08ekcbbj4_param.dat
memory –load %readmemh <path.to.model.inst>.L21.jedec_param_page –file mt29f1ht08ekcbbj4_jedec_param.dat
memory –load %readmemh <path.to.model.inst>.L22.param_page –file mt29f1ht08ekcbbj4_param.dat
memory –load %readmemh <path.to.model.inst>.L22.jedec_param_page –file mt29f1ht08ekcbbj4_jedec_param.dat

8.3. READ UNIQUE ID

The READ UNIQUE ID command is used to read a unique identifier programmed into the target. Preloading the uid_page is similar to preloading the param_page mentioned in the previous section. The path to each LUN's unique id page is as follows:

<path.to.model.inst>.L11.uid_page
<path.to.model.inst>.L12.uid_page
<path.to.model.inst>.L21.uid_page
<path.to.model.inst>.L22.uid_page

9. Features

The following table shows a list of features and feature support for the NAND flash model:

FEATURE	SUPPORT	NOTE
COMMANDS		
Reset	Yes	
Synchronous Reset	Yes	
Read ID	Yes	
Read Parameter Page (ONFI)	Yes	
Read Parameter Page (JEDEC)	Yes	
Read Unique ID	Yes	
Volume Select	Yes	
Get Features	Yes	
Set Features	Partial	Activate Synchronous Interface, Appoint volume address, SLC mode.
Get Features by LUN	Yes	
Set Features by LUN	Partial	See Set Features above.
Read Status	Yes	
Read Status Enhanced	Yes	
Change Read Column	Yes	
Change Read Column Enhanced (ONFI)	Yes	
Change Read Column Enhanced (JEDEC)	Yes	
Change Write Column	Yes	
Change Row Address	Yes	
Read Mode	Yes	
Read Page	Yes	
Read Page Multi-Plane	Yes	
Read Page Cache Sequential	Yes	
Read Page Cache Random	Yes	
Read Page Cache Last	Yes	
Program Page	Yes	
Program Page Multi-Plane	Yes	
Program Page Cache	Yes	
Program Suspend	Yes	
Program Resume	Yes	
Erase Block	Yes	
Erase Block Multi-Plane (ONFI)	Yes	
Erase Block Multi-Plane (JEDEC)	Yes	
Erase Suspend	Yes	
Erase Resume	Yes	
Copyback Read	Yes	
Copyback Program	Yes	
Copyback Program Multi-Plane	Yes	
Reset LUN	Yes	
Snap Read	Yes	B17A and B16A models
SLC Mode Enable	Yes	B17A and B16A models
SLC Mode Disable	Yes	B17A and B16A models
Fixed address read status enhanced	Yes	B17A and B16A models
SPECIAL OPERATIONS		

FEATURE	SUPPORT	NOTE
One-Time Programmable (OTP) Operations	No	
ODT Configure Operation	Partial	Command accepted only, no function executed
ZQ Calibration	No	Command accepted only, no function executed
Multi-Plane Operations	Yes	
Read Retry	No	
Interleaved Die (Multi-LUN) Operations	Yes	
TLC Two-Pass Programming	Yes	B17A and B16A models
SLC Mode via Set Feature	Yes	
Error Management	No	Spare area is available
Read Offset Operations	No	
Soft Data Read Operations	No	
Auto Read Calibration Operations	No	
Partial Page Read	No	
SleepLite	No	

The following table shows the command set as described in the Micron data sheet.

Command	Com-mand Cycle #1	# of Valid Address Cycles[9]	Data Input Cycles	Com-mand Cycle #2	# of Valid Address Cycles #2	Com-mand Cycle #3	Valid while se-lected LUN is busy (1)	Valid while other LUNs are busy (2)	Notes
Reset Operations									
RESET	FFh	0	-	-	-	-	Yes	Yes	
SYNCHRONOUS RESET	FCh	0	-	-	-	-	Yes	Yes	
RESET LUN	FAh	3/4	-	-	-	-	Yes	Yes	
Identification Operations									
READ ID	90h	1	-	-	-	-			3
READ PARAMETER PAGE	ECh	1	-	-	-	-			
READ UNIQUE ID	EDh	1	-	-	-	-			
Configuration Operations									
VOLUME SELECT	E1h	1	-	-	-	-			
GET FEATURES	EEh	1	-	-	-	-			3
SET FEATURES	EFh	1	4	-	-	-			4
GET FEATURES by LUN	D4h	2	-	-	-	-		Yes	3
SET FEATURES by LUN	D5h	2	4	-	-	-		Yes	4
Status Operations									
READ STATUS	70h	0	-	-	-	-	Yes		
READ STATUS ENHANCED	78h	3/4	-	-	-	-	Yes	Yes	
Column Address Operations									
CHANGE READ COLUMN	05h	2	-	E0h	-	-		Yes	
CHANGE READ COLUMN ENHANCED (ONFI)	06h	5/6	-	E0h	-	-		Yes	
CHANGE READ COLUMN ENHANCED (JEDEC)	00h	5/6	-	05h	2	E0h		Yes	
CHANGE WRITE COLUMN	85h	2	opt	-	-	-		Yes	
CHANGE ROW ADDRESS	85h	5/6	opt	11h(opt)	-	-		Yes	5
Read Operations									
READ MODE	00h	0	-	-	-	-		Yes	
READ PAGE	00h	5/6	-	30h	-	-		Yes	6
SNAP READ	00h	5/6	-	20h				Yes	

Command	Com-mand Cycle #1	# of Valid Address Cycles[9]	Data Input Cycles	Com-mand Cycle #2	# of Valid Address Cycles #2	Com-mand Cycle #3	Valid while se-lected LUN is busy (1)	Valid while other LUNs are busy (2)	Notes
READ PAGE MULTIPLANE	00h	5/6	-	32h	-	-		Yes	
READ PAGE CACHE SEQUENTIAL	31h	0	-	-	-	-		Yes	7
READ PAGE CACHE RANDOM	00h	5/6	-	31h	-	-		Yes	6,7
READ PAGE CACHE LAST	3Fh	0	-	-	-	-		Yes	7
Program Operations									
PROGRAM PAGE	80h	5/6	Yes	10h				Yes	
PROGRAM PAGE MULTI-PLANE	80h or 81h	5/6	Yes	11h				Yes	
PROGRAM PAGE CACHE	80h	5/6	Yes	15h				Yes	8
PROGRAM SUSPEND	84h	5/6	-	-	-	-	Yes	Yes	
PROGRAM RESUME	13h	5/6	-	-	-	-		Yes	
Erase Operations									
ERASE BLOCK	60h	3/4	-	D0h				Yes	
ERASE BLOCK MULTI-PLANE (ONFI)	60h	3/4	-	D1h				Yes	
ERASE BLOCK MULTI-PLANE (JEDEC)	60h	3/4	-	60h	3	D0h		Yes	
ERASE SUSPEND	61h	3/4	-	-	-	-	Yes	Yes	
ERASE RESUME	D2h	-	-	-	-	-		Yes	
Copyback Operations									
COPYBACK READ	00h	5/6	-	35h				Yes	6
COPYBACK PROGRAM	85h	5/6	opt	10h				Yes	
COPYBACK PROGRAM MULTI-PLANE	85h	5/6	opt	11h				Yes	

Notes from Micron data sheet:

1. Busy means RDY = 0.
2. These commands can be used for interleaved die (multi-LUN) operations.
3. The READ ID (90h), GET FEATURES (EEh), and GET FEATURES by LUN (D4h) output identical data on rising and falling DQS edges.
4. The SET FEATURES (EFh) and SET FEATURES by LUN (D5h) commands require data transition prior to the rising edge of CLK, with identical data for the rising and falling edges.
5. Command cycle #2 of 11h is conditional. See CHANGE ROW ADDRESS (85h) (page 114 of data sheet) for more details.
6. This command can be preceded by READ PAGE MULTI-PLANE (00h-32h) command to accommodate a maximum simultaneous multi-plane array operation.
7. Issuing a READ PAGE CACHE-series (31h, 00h-31h, 00-32h, 3Fh) command when the array is busy (RDY = 1, ARDY = 0) is supported if the previous command was a READ PAGE (00h-30h) or READ PAGE CACHE-series command; otherwise, it is prohibited.
8. Issuing a PROGRAM PAGE CACHE (80h-15h) command when the array is busy (RDY = 1, ARDY = 0) is supported if the previous command was a PROGRAM PAGE CACHE (80h-15h) command; otherwise, it is prohibited.
9. Refer to Device and Array Organization section for details of when the additional address cycle is required.

Set Feature Command for SLC mode:

Mode	FA	P1	P2	P3	P4
SLC	91h	00h	01h	00h	00h

TLC	91h	04h	01h	00h	00h

P2 = 01h means Non-squashed SLC address mode – address bits do not shift from TLC address (default).
P2 = 00h means Squashed address mode - address bits shift from TLC address.

10. 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.

10.1. 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

11. Initialization Sequence

The ONFI 4.x Flash model requires that the memory controller or host follows the initialization sequence as documented in the specification. The sequence basically entails the following steps.

For Single Volume (one package per CE# or host target):

1. The asynchronous interface is active by default for each target.
2. The RESET (FFh) command must be the first command issued to the target (CE#). The target will become busy. The RESET busy time can be monitored with the R/B# pin or checked by polling the status register with the 70h status command.
3. Read configuration data from the model using READ ID, READ PARAMETER PAGE, etc. (optional)
4. The model is now initialized and ready for normal operation.

For CE# Pin Reduction and Volume Addressing:

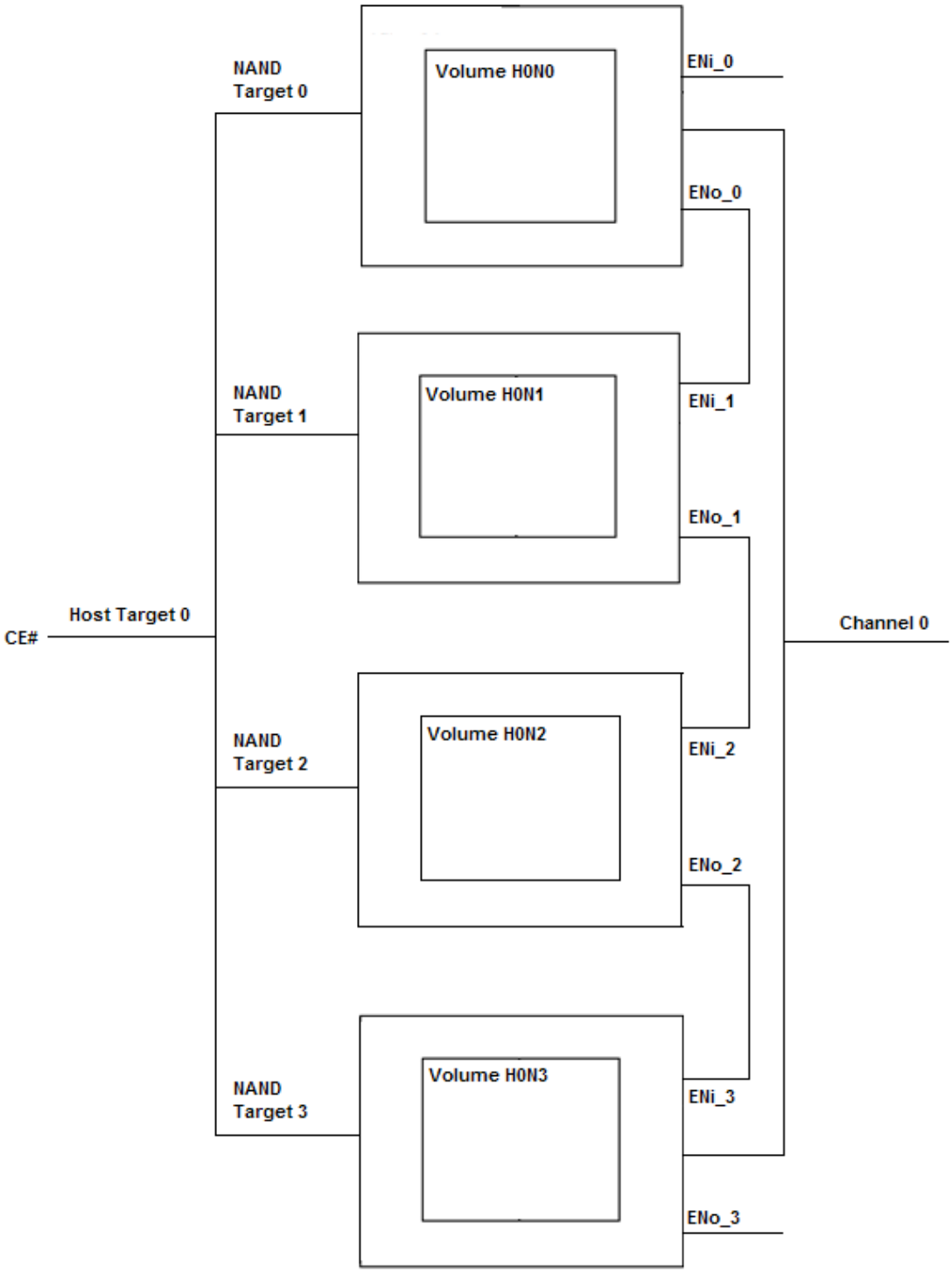
1. The asynchronous interface is active by default for each target.
2. Multiple Host Targets (CE#'s) can be active at the same time. See the note below on how to properly issue multiple SET FEATURES (EFh) commands to appoint volume addresses.
3. The RESET (FFh) command must be the first command issued to the target (CE#) or targets. The target will become busy. The RESET busy time can be monitored with the R/B# pin or checked by polling the status register with the 70h status command.
4. R/B# goes LOW for tRST and can be monitored by the host by issuing READ STATUS (70h) commands and monitoring.
5. Read configuration data from the model using READ ID, READ PARAMETER PAGE, etc. (optional)
6. The host issues SET FEATURES (EFh) command to the Volume Configuration feature address to appoint the Volume address for the first NAND Target. The Volume Address specified shall be unique amongst all NAND Targets. After the SET FEATURE (EFh) command completes, ENo is set to one to enable the next volume in the chain to be appointed an address.
7. For each NAND Target connected to a CE# pin or Host Target, repeat step 6.

8. Repeat steps 3-7 for the next Host Target (CE# pin), if any.
9. To complete the initialization process, a VOLUME SELECT (E1h) command is issued to select the next Volume that is going to execute a command.

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.

Note: In cases where there are multiple NAND Targets within a package (i.e., multiple CE#'s), those NAND Targets share the same ENo signal, the host shall not stagger SET FEATURES (EFh) commands that appoint the Volume addresses. If the SET FEATURES (EFh) commands are not issued simultaneously then the host shall wait until Volume appointment for previous NAND Target(s) is complete before issuing the next SET FEATURES (EFh) command to appoint the Volume address for the next NAND Target that shares the ENo within a package.

A diagram of CE# Pin Reduction Topology with single channel is shown below:



12. Model Size

To reduce memory utilization each LUN has only 16 blocks (blocks 0 to 15) but the actual device has 2192 blocks. If larger size is needed please see the section on large memory support.

13. Large Memory Support

MMP model capacity has been limited to 1G words due to the current 30 bit address width limitation in IXCOM. To work around this constraint, a multiple core memory array generator (mmp_gen_mem_4rp_8wp.vp) has been incorporated into these large sized memory models that splits larger core memory arrays into multiple 1G arrays. In these cases, the file mmp_gen_mem_4rp_8wp.vp automatically defines Verilog macro MMP_LG_MEM_BITS with a value of '30' by default. To enable large memory support the user should specify the macro MMP_LG_MEM_BITS with value '30' on the command line or add `define within <model_name>.vp and instantiate the model with nb (number of blocks per LUN) parameter value greater than the default value of 16 or modify nb value within <model_name>.vp.

Two ways to enable large memory support	Command line and instantiation	Modify <model_name>.vp
Macro define	ixcom <model_name>.vp … +define+MMP_LG_MEM_BITS=30	`define MMP_LG_MEM_BITS 30
Parameter change	mt29f384g08ebcbb #(.nb(32)) nand_i0 (…);	parameter nb = 32;

12.1 Preloading Multiple Arrays

With a single array of 30 bits or less the address mapping is simply: array_addr = {LA,BA,PA,CA}. In this case the hierarchical memory array name appears thusly:

tb_top.nand_i0.L11.mem_array

In scenarios with array address space that is greater than 30 bits, there are multiple arrays which are mapped using distinctly different hierarchical naming. The hierarchical path for the array names associated with each core memory array of the large memory are reported as output to the "memory –list" command or can be viewed in the dbFiles/xcva_top_et5mpart file. For example, with a 32 bit address the user will see 4 arrays with naming as follows:

tb_top.nand_i0.L11.mem1.\multiple.array_0_.u1 .memcore addresses 0 .. 1G - 1
tb_top.nand_i0.L11.mem1.\multiple.array_1_.u1 .memcore addresses 1G .. 2G - 1
tb_top.nand_i0.L11.mem1.\multiple.array_2_.u1 .memcore addresses 2G .. 3G - 1
tb_top.nand_i0.L11.mem1.\multiple.array_3_.u1 .memcore addresses 3G .. 4G - 1

Multiple data files for preloading each separate memory array are also required. In the example above, there will be four data files needed to preload the entire large memory.

Likewise, multiple memory –load commands are needed to preload the large memory of this example. An xeDebug preload example for the case above will look as follows:

memory -load %readmemh tb_top.nand_i0.L11.mem1.multiple.array_0_.u1.memcore -file mem0.dat

memory -load %readmemh tb_top.nand_i0.L11.mem1.multiple.array_1_.u1.memcore -file mem1.dat

memory -load %readmemh tb_top.nand_i0.L11.mem1.multiple.array_2_.u1.memcore -file mem2.dat

memory -load %readmemh tb_top.nand_i0.L11.mem1.multiple.array_3_.u1.memcore -file mem3.dat

14. Limitations

1. Model does not check illegal sequence of command cycles.
2. Model does not support timing parameters, except tRHOH.
3. The time to program a page or erase a block is 1 fclk (fast clock) per address location.
4. Unsupported and partially supported features are listed in Features table.

15. Timing Parameter tRHOH

The model does not generally support timing parameters because it is cycle based. However tRHOH (an asynchronous mode parameter that controls RE# HIGH to output hold) is supported in number of fclks (fast clocks). The user may define a macro at compile time to hold the data by a number of fclks after RE# goes high. Use this macro only if data output is not already held long enough. The model holds the data valid for 2 fclks by default. The data can be held 3 additional fclks by using the following example command line option in ixcom flow:

vlan +define+tRHOH=3

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+mt29f384g08ebcbb \
../tb.v \
../src/mt29f_384.vp \
../src/mt29f384g08ebcbb.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/mt29f_384.vp ../src/mt29f384g08ebcbb.vp

vaelab –keepRtlSymbol –keepAllFlipFlop –outputVlog mt29f384g08ebcbb.vgp
mt29f384g08ebcbb
```

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.

An example of the command for compilation (including synthesis) of a model with large memory support in IXCOM flow is shown below:

```
ixcom -64bit +sv -ua +dut+mt29f384g08ebcbb \
../tb.v \
../src/mt29f_384.vp \
../src/mmp_gen_mem_4rp_8wp.vp \
```

```
../src/mmp_submem_4rp_8wp.vp \
../src/mt29f384g08ebcbb.vp \
-incdir ../../../utils/cdn_mmp_utils/sv \
../../../utils/cdn_mmp_utils/sv/cdn_mmp_utils.sv \
+define+MMP_LG_MEM_BITS=30
```

ICE flow synthesis commands:

```
vavlog ../src/mt29f_384.vp ../src/mmp_gen_mem_4rp_8wp.vp
../src_mmp_submem_4rp_8wp.vp ../src/mt29f384g08ebcbb.vp
+define+MMP_LG_MEM_BITS=30

vaelab –keepRtlSymbol –keepAllFlipFlop –outputVlog mt29f384g08ebcbb.vgp
mt29f384g08ebcbb
```

16.1. Model file list

mt29f_384.vp - LUN or die module that is instantiated by a subset of ONFI 4.0 models
mt29f_512.vp - LUN or die module that is instantiated by a subset of ONFI 4.0 models
<model_name>.vp - model wrapper that instantiates one or more LUNs.
mmp_gen_mem_4rp_8wp.vp – memory array generator for large memory support.
mmp_submem_4rp_8wp.vp – memory array being instantiated by memory array generator.

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 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, read, 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

- For issues that may not be ONFI 4.0 specific please review the *Memory Model Portfolio FAQ for All Models User Guide.*

- **Debug signals**:

  The following key signals are useful to examine when debugging possible issues:

  - CE — Chip Enable should be low when the selected LUN's DQ bus is active
  - WP — Write Protect should be high during normal operations
  - R/B — Ready/Busy, the host should wait for model to be ready before issuing new command
  - DQS — Data Strobe should be active in NV-DDR and NV-DDR2 modes during data IO cycles
  - CLK — Clock input should be active in NV-DDR mode during data IO cycles
  - Fclk — Fast clock should be active when LUN is busy

  - AL — Address Latch Enable (ALE)
  - CL — Command Latch Enable (CLE)
  - Ebar — Chip Enable (CE#)
  - Rbar — Read Enable (RE#)
  - Wbar — Write Enable (WE#)
  - Rbbar — Ready/Busy (R/B#)
  - Io — IO bus (DQ)
  - cmd_reg — Current command
  - avol_addr — Appointed Volume address

- **Golden waveform:** A package with a reference waveform is available. The waveform shows basic startup sequence - reset, read id, program page, read page, is available in the release under the onfi_4.0/micron/golden_waveform directory. It may be useful to have a look at this waveform when using the model for the first time.

  Command sequences:
  - Initialization sequence:
    - Issue RESET command in asynchronous mode
    - R/B# shall go low for about 200 fclks then high
    - The model is ready of normal operations in the default asynchronous mode
  - Volume Appoint, Volume Select, Read Parameter Page, Read Unique ID, Program Page, Read Page
    - Volume 3 is appointed to cdn1.L11 and volume 3 is selected
  - Set Features to sync mode, Get Features, Read Page
  - Program Page, Program Suspend, Read Page, Program Resume, Read Page

- **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
August 2015	1.0	Initial Release
September 2015	1.1	Specified that read configuration in initialization sequence is optional, removed 'model does not check attempts to program a bit to 1' from Limitations list, and corrected a typo on 'Set Features by LUN' in Features table.
October 2015	1.2	Added large memory support.
November 2015	1.3	Added I/O Signal Description section. Mention MMP FAQ in Debugging section.
January 2016	1.4	Update for Palladium-Z1 and VXE
May 2016	1.5	Remove BETA watermark
July 2016	1.6	Remove hyphen in Palladium naming
August 2017	1.7	Add 512Gb and 256Gb per die models. Merge 'Reference Waveform' section with Debugging section to align with other UG. Add info.
September 2017	1.8	Add two command sequences to golden waveform
October 2017	1.9	Update Feature and Parameter Tables for B17A and B16A models
November 2017	2.0	Add MLC models
January 2018	2.1	Modify header and footer
February 2018	2.2	Ported ECC section from NAND UG to ONFI and TDDR UGs
March 2018	2.3	Add sections on model clocking and on extendable busy timing
May 2018	2.4	Add note in Features table for B17A and B16A features
June 2018	2.5	Remove notes for Beta level indication for B17A and B16A TLC NAND models.
July 2018	2.6	Update for new utility library