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

ONFI 2.0 NAND Flash
Palladium Memory Model
User Guide

Document Version: 2.6

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

ONFI 2.0 NAND Flash Palladium Memory Models

1. Introduction

The Cadence Palladium ONFI 2.0-compliant NAND Flash Models are based on data sheet specifications of the following Micron devices:

MT29E NAND Flash memory with MLC technology (32Gb per die, ONFI 2.1, L63B_enterprise)
MT29F NAND Flash memory with MLC technology (32Gb per die, ONFI 2.1, L63B)
MT29F NAND Flash memory with SLC technology (16Gb per die, ONFI 2.1, M62B)
MT29F NAND Flash memory with MLC technology (64Gb per die, ONFI 2.2, L74A)
MT29F NAND Flash memory with SLC technology (32Gb per die, ONFI 2.2, M73A)
MT29F NAND Flash memory with MLC technology (64Gb per die, ONFI 2.3, L84A)
MT29F NAND Flash memory with MLC technology (32Gb per die, ONFI 2.3, L83A)

Many of the models support both asynchronous and synchronous or DDR interface however some models support asynchronous interface only. Please check the part number section 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 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, 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 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 2.1)	Density	Level	# of Die	# of CE#	I/O	Interface
MT29E64G08CECBB	64Gb	MLC	2	2	Separate	Sync/Async
MT29E128G08CKCBB	128Gb	MLC	4	2	Separate	Sync/Async
MT29E128G08CMCBB	128Gb	MLC	4	4	Separate	Sync/Async
MT29E256G08CUCBB	256Gb	MLC	8	4	Separate	Sync/Async
MT29F32G08CBABA	32Gb	MLC	1	1	Common	Async Only
MT29F64G08CFABA	64Gb	MLC	2	2	Common	Async Only
MT29F128G08CJABA	128Gb	MLC	4	2	Common	Async Only
MT29F32G08CBABB	32Gb	MLC	1	1	Common	Sync/Async
MT29F64G08CFABB	64Gb	MLC	2	2	Common	Sync/Async
MT29F128G08CJABB	128Gb	MLC	4	2	Common	Sync/Async
MT29F64G08CEABA	64Gb	MLC	2	2	Separate	Async Only
MT29F128G08CKABA	128Gb	MLC	4	2	Separate	Async Only
MT29F128G08CMABA	128Gb	MLC	4	4	Separate	Async Only
MT29F256G08CUABA	256Gb	MLC	8	4	Separate	Async Only
MT29F32G08CBCBB	32Gb	MLC	1	1	Common	Sync/Async
MT29F64G08CECBB	64Gb	MLC	2	2	Separate	Sync/Async
MT29F128G08CKCBB	128Gb	MLC	4	2	Separate	Sync/Async
MT29F128G08CMCBB	128Gb	MLC	4	4	Separate	Sync/Async
MT29F256G08CUCBB	256Gb	MLC	8	4	Separate	Sync/Async
MT29F16G08ABABA	16Gb	SLC	1	1	Common	Async Only
MT29F32G08AFABA	32Gb	SLC	2	2	Common	Async Only

MT29F64G08AJABA	64Gb	SLC	4	2	Common	Async Only
MT29F64G08AKABA	64Gb	SLC	4	2	Separate	Async Only
MT29F64G08AMABA	64Gb	SLC	4	4	Separate	Async Only
MT29F128G08AUABA	128Gb	SLC	8	4	Separate	Async Only
MT29F16G08ABCBB	16Gb	SLC	1	1	Common	Sync/Async
MT29F32G08AECBB	32Gb	SLC	2	2	Separate	Sync/Async
MT29F64G08AKCBB	64Gb	SLC	4	2	Separate	Sync/Async
MT29F64G08AMCBB	64Gb	SLC	4	4	Separate	Sync/Async
MT29F128G08AUCBB	128Gb	SLC	8	4	Separate	Sync/Async

Model (ONFI 2.2)	Density	Level	# of Die	# of CE#	I/O	Interface
MT29F64G08CBAAA	64Gb	MLC	1	1	Common	Async Only
MT29F64G08CBAAB	64Gb	MLC	1	1	Common	Sync/Async
MT29F64G08CBCAB	64Gb	MLC	1	1	Common	Sync/Async
MT29F128G08CEAAA	128Gb	MLC	2	2	Separate	Async Only
MT29F128G08CECAB	128Gb	MLC	2	2	Separate	Sync/Async
MT29F128G08CFAAA	128Gb	MLC	2	2	Common	Async Only
MT29F128G08CFAAB	128Gb	MLC	2	2	Common	Sync/Async
MT29F256G08CJAAA	256Gb	MLC	4	2	Common	Async Only
MT29F256G08CJAAB	256Gb	MLC	4	2	Common	Sync/Async
MT29F256G08CKAAA	256Gb	MLC	4	2	Separate	Async Only
MT29F256G08CKCAB	256Gb	MLC	4	2	Separate	Sync/Async
MT29F256G08CMAAA	256Gb	MLC	4	4	Separate	Async Only

MT29F256G08CMCAB	256Gb	MLC	4	4	Separate	Sync/Async
MT29F512G08CUAAA	512Gb	MLC	8	4	Separate	Async Only
MT29F512G08CUCAB	512Gb	MLC	8	4	Separate	Sync/Async
MT29F32G08ABAAA	32Gb	SLC	1	1	Common	Async Only
MT29F32G08ABCAB	32Gb	SLC	1	1	Common	Sync/Async
MT29F64G08AFAAA	64Gb	SLC	2	2	Common	Async Only
MT29F64G08AECAB	64Gb	SLC	2	2	Separate	Sync/Async
MT29F128G08AJAAA	128Gb	SLC	4	2	Common	Async Only
MT29F128G08AKAAA	128Gb	SLC	4	2	Separate	Async Only
MT29F128G08AMAAA	128Gb	SLC	4	4	Separate	Async Only
MT29F128G08AKCAB	128Gb	SLC	4	2	Separate	Sync/Async
MT29F128G08AMCAB	128Gb	SLC	4	4	Separate	Sync/Async
MT29F256G08AUAAA	256Gb	SLC	8	4	Separate	Async Only
MT29F256G08AUCAB	256Gb	SLC	8	4	Separate	Sync/Async

Model (ONFI 2.3)	Density	Level	# of Die	# of CE#	I/O	Interface
MT29F32G08CBADA	32Gb	MLC	1	1	Common	Async Only
MT29F32G08CBADB	32Gb	MLC	1	1	Common	Sync/Async
MT29F32G08CBCDB	32Gb	MLC	1	1	Common	Sync/Async
MT29F64G08CECDB	64Gb	MLC	2	2	Separate	Sync/Async
MT29F64G08CBABA	64Gb	MLC	1	1	Common	Async Only
MT29F64G08CBABB	64Gb	MLC	1	1	Common	Sync/Async
MT29F64G08CBCBB	64Gb	MLC	1	1	Common	Sync/Async
MT29F128G08CFABA	128Gb	MLC	2	2	Common	Async Only

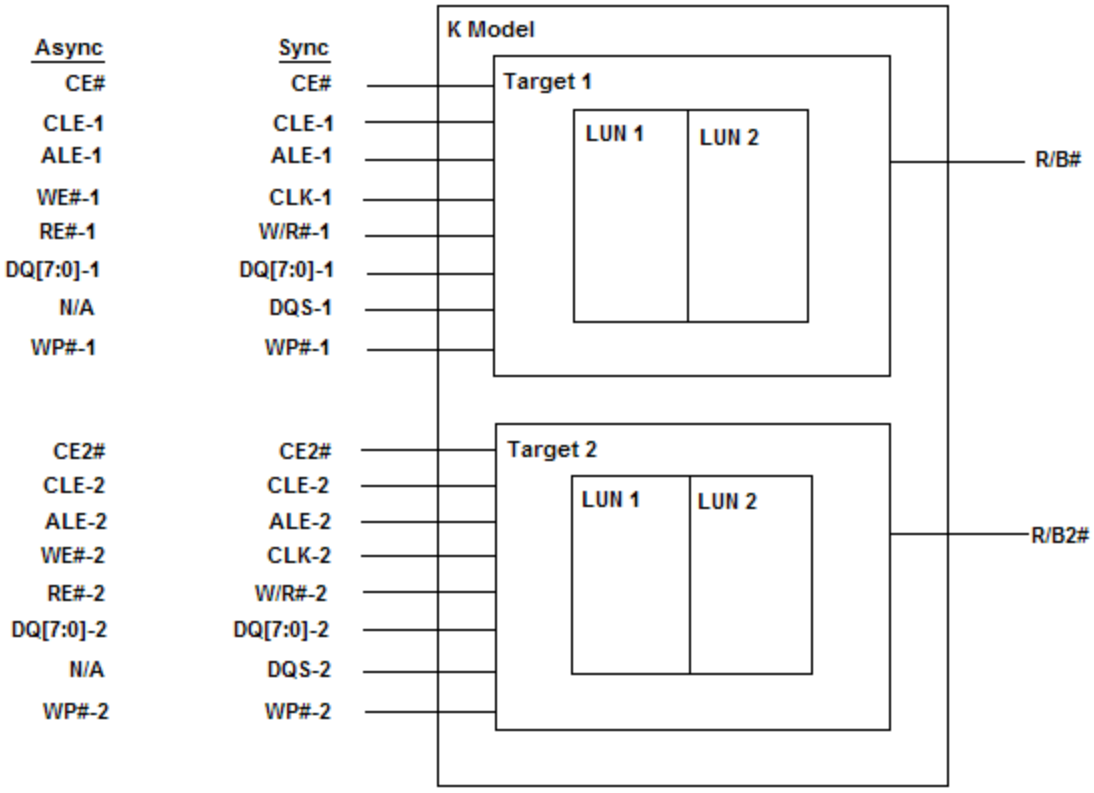
MT29F128G08CFABB	**128Gb**	**MLC**	**2**	**2**	**Common**	**Sync/Async**
MT29F128G08CECBB	**128Gb**	**MLC**	**2**	**2**	**Separate**	**Sync/Async**
MT29F256G08CJABA	**256Gb**	**MLC**	**4**	**2**	**Common**	**Async Only**
MT29F256G08CJABB	**256Gb**	**MLC**	**4**	**2**	**Common**	**Sync/Async**
MT29F256G08CKCBB	**256Gb**	**MLC**	**4**	**2**	**Separate**	**Sync/Async**
MT29F256G08CMCBB	**256Gb**	**MLC**	**4**	**4**	**Separate**	**Sync/Async**
MT29F512G08CUCBB	**512Gb**	**MLC**	**8**	**4**	**Separate**	**Sync/Async**

Notes: MLC = 2bits/cell
SLC = 1bit/cell
Separate I/O = 2 sets of pins
Common I/O = 1 set of pins
(pin set: ALE,CLE,DQ,DQS,RE#,WE#,WP#)
number of R/B# = number of CE#
Die and LUN are used interchangeably
Target and CE# are used interchangeably
Synchronous, DDR, and perhaps Toggle may refer to the same interface

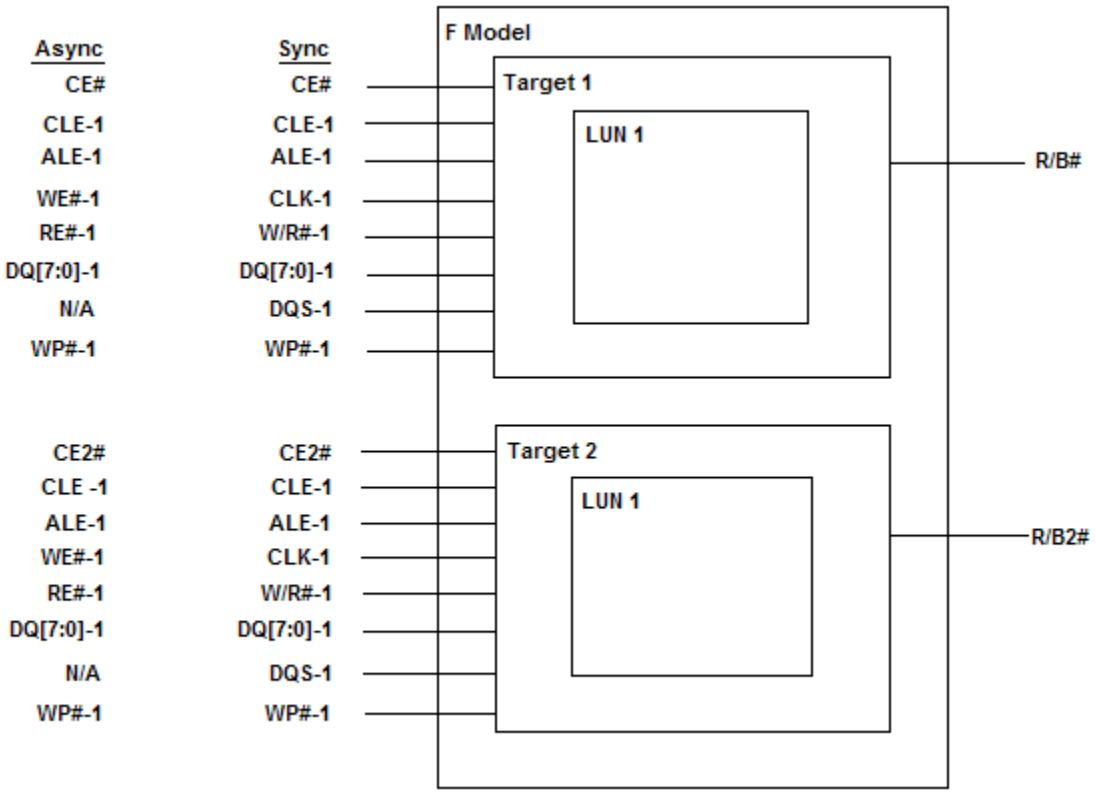
4. Model Block Diagram

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 instantiate the model based on the model's I/O type – Common or Separate.

A block diagram of a K model that has 4 die or LUNs, 2 CE#s or targets, and separate I/O pins is shown below. For models that have 4 targets and separate I/O, targets 1 and 3 share pin set 1, targets 2 and 4 share pin set 2. For models with common I/O, all targets share one set of pins. Each target gets its own CE# and R/B# pins.



The following block diagram shows an F model that has 2 LUNs, 2 CE#s and common I/O pins.



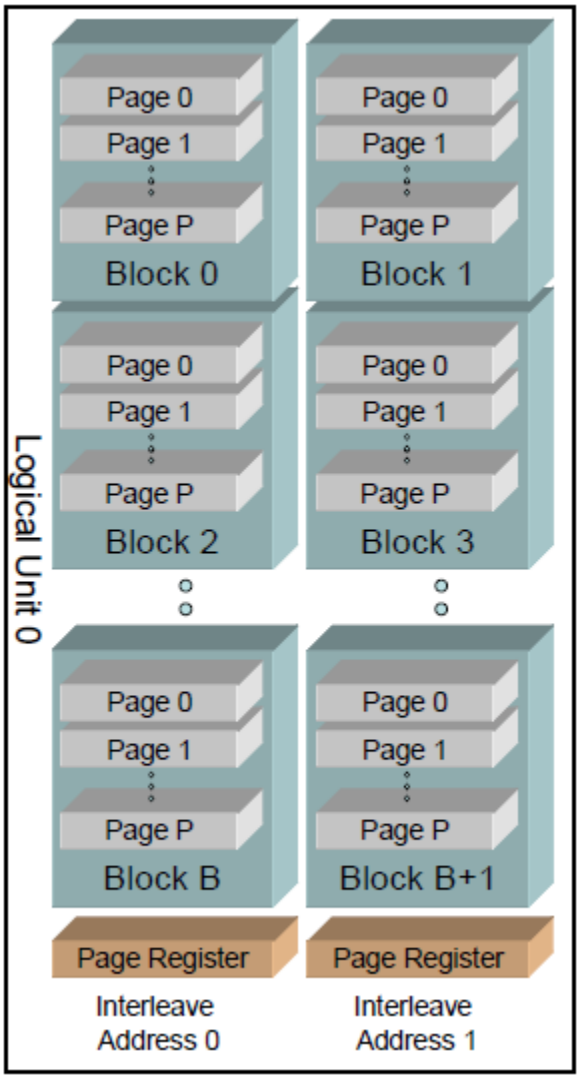
5. Address mapping

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 named *mem_array* within each LUN. 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. Here are the array organization and addressing cycle table for MLC models, followed by SLC models. Note that SLC models have one less page address bit.

Array Organization for MLC Array



1 page = (4K + 224 bytes)

1 block = (4K + 244) bytes x 256 pages
= (1024K + 56K) bytes

1 plane = (1024K + 56K) bytes x 2048 blocks
= 17,280Mb

1 LUN = 17,280Mb x 2 planes
= 34,560Mb or 32Gb

Plane 0 contains blocks 0, 2, 4, ..., 4094
Plane 1 contains blocks 1, 3, 5, ..., 4095

MLC Array Organization per Logical Unit (LUN) or die.
credit: diagram taken from ONFI 2.0 Specification dated 27-Feb-2008 page 32 figure 16

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	LOW	LOW	CA12	CA11	CA10	CA9	CA8
Third	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Fourth	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
Fifth	LOW	LOW	LOW	LA0	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.

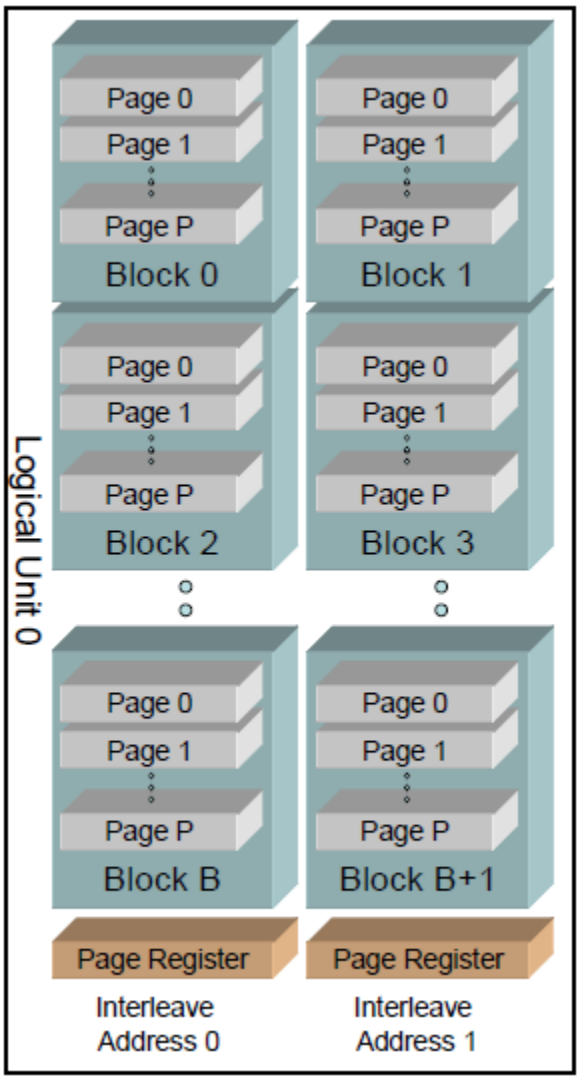
When using the synchronous interface, CA0 is forced to 0 internally; one data cycle always returns one even byte and one odd byte.

Column addresses 4320 (10E0h) through 8191 (1FFFh) are invalid, out of bounds, do not exist in the device, and cannot be addressed.

BA[8] are the plane-select bits:
Plane 0: BA[8] = 0
Plane 1: BA[8] = 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

Array Organization for SLC Array



1 page = (4K + 224 bytes)

1 block = (4K + 224) bytes x 128 pages
= (512K + 28K) bytes

1 plane = (512K + 28K) bytes x 2048 blocks
= 8640Mb

1 LUN = 8640Mb x 2 planes
= 17,280Mb or 16Gb

Plane 0 contains blocks 0, 2, 4, ..., 4094
Plane 1 contains blocks 1, 3, 5, ..., 4095

SLC Array Organization per Logical Unit (LUN) or die.
credit: diagram take from ONFI 2.0 Specification
dated 27-Feb-2008 page 32 figure 16

Address Cycle Table for SLC Array

Cycle	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Second	LOW	LOW	LOW	CA12	CA11	CA10	CA9	CA8
Third	BA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Fourth	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
Fifth	LOW	LOW	LOW	LOW	LA0	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.

Column addresses 4320 (10E0h) through 8191 (1FFFh) are invalid, out of bounds, do not exist in the device, and cannot be addressed.

BA[7] are the plane-select bits:
Plane 0: BA[7] = 0
Plane 1: BA[7] = 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

6. Feature Address Definitions

In the NAND Flash 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 feature that can be activated is the synchronous or DDR interface, in models that support Sync/Async interface. 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 OTP 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
02h-0Fh	Reserved
10h	Programmable output drive strength
11h-7Fh	Reserved
80h	Programmable output drive strength
81h	Programmable RB# pull-down strength
82h-8Fh	Reserved
90h	Array operation mode
91h-FFh	Reserved

7. ID Operations

7.1. READ ID

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

7.2. READ PARAMETER PAGE

The data for the parameter page is provided in the <model_name><package_code>_<param_page_revision>_param.dat file. For example, mt29f16g08abcbbh1_v03_param.dat. The package_code is two characters. The param_page_revision, if specified, is a newer revision of the parameter page. The revision number corresponds to byte 253 of the Parameter Page Data Structure table in Micron data sheet, as well as byte 253 in the data file. This data file should be preloaded into all LUNs in the model if the user wants to read ONFI information from the model. The instance names are L<CE><LUN>. Using a K model as an example, there are 2 CEs and 2 LUNs per CE. The path to each LUN's parameter page is as follows:

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

7.3. 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 section 5.2. 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

8. Features

The following table shows a list of features 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	ONFI 2.3 models
Read Unique ID	Yes	
Get Feature	Yes	
Set Feature	Partial	Activate Synchronous Interface
Read Status	Yes	
Read Status Enhanced	Yes	
Change Read Column	Yes	
Change Read Column Enhanced (ONFI)	Yes	
Change Read Column Enhanced (JEDEC)	Yes	ONFI 2.3 models
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	
Erase Block	Yes	
Erase Block Multi-Plane (ONFI)	Yes	
Erase Block Multi-Plane (JEDEC)	Yes	ONFI 2.3 models
Copyback Read	Yes	
Copyback Program	Yes	
Copyback Program Multi-Plane	Yes	
Reset LUN	Yes	ONFI 2.2 and 2.3 models
SPECIAL OPERATIONS		
One-Time Programmable (OTP) Operations	No	
Multi-Plane Operations	Yes	
Interleaved Die (Multi-LUN) Operations	Yes	
Error Management	No	Spare area is available

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

Command	Command Cycle #1	Number of Valid Address Cycles	Data Input Cycles	Command Cycle #2	Number of Valid Address Cycles #2	Command Cycle #3	Valid While Selected 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	-	-	-	-	Yes	Yes	
Identification operations									
READ ID	90h	1	-	-	-	-			3
READ PARAMETER PAGE	ECh	1	-	-	-	-			
READ UNIQUE ID	EDh	1	-	-	-	-			
Configuration operations									
GET FEATURES	EEh	1	-	-	-	-			3
SET FEATURES	EFh	1	4	-	-	-			4
Status operations									
READ STATUS	70h	0	-	-	-	-	Yes		
READ STATUS ENHANCED	78h	3	-	-	-	-	Yes	Yes	
Column address operations									
CHANGE READ COLUMN	05h	2	-	E0h	-	-		Yes	
CHANGE READ COLUMN ENHANCED (ONFI)	06h	5	-	E0h	-	-		Yes	
CHANGE READ COLUMN ENHANCED (JEDEC)	00h	5	-	05h	2	E0h		Yes	
CHANGE WRITE COLUMN	85h	2	Optional	-	-	-		Yes	
CHANGE ROW ADDRESS	85h	5	Optional	-	-	-		Yes	5
Read operations									
READ MODE	00h	0	-	-	-	-		Yes	
READ PAGE	00h	5	-	30h	-	-		Yes	6
READ PAGE MULTI-PLANE	00h	5	-	32h	-	-		Yes	
READ PAGE CACHE SEQUENTIAL	31h	0	-	-	-	-		Yes	7
READ PAGE CACHE RANDOM	00h	5	-	31h	-	-		Yes	6,7
READ PAGE CACHE LAST	3Fh	0	-	-	-	-		Yes	7
Program operations									
PROGRAM PAGE	80h	5	Yes	10h	-	-		Yes	
PROGRAM PAGE MULTI-PLANE	80h	5	Yes	11h	-	-		Yes	
PROGRAM PAGE CACHE	80h	5	Yes	15h	-	-		Yes	8
Erase operations									
ERASE BLOCK	60h	3	-	D0h	-	-		Yes	
ERASE BLOCK MULTI-PLANE (ONFI)	60h	3	-	D1h	-	-		Yes	
ERASE BLOCK MULTI-PLANE (JEDEC)	60h	3	-	60h	3	D0h		Yes	
Copyback operations									
COPYBACK READ	00h	5	-	35h	-	-		Yes	6
COPYBACK PROGRAM	85h	5	Optional	10h	-	-		Yes	
COPYBACK PROGRAM MULTI-PLANE	85h	5	Optional	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) and GET FEATURES (EEh) output identical data on rising and falling DQS edges.
4. The SET FEATURES (EFh) command requires data transition prior to the rising edge of CLK, with idential data for the rising and falling edges.
5. Command cycle #2 of 11h is conditional.
6. This command can be preceded by up to one READ PAGE MULTI-PLANE (00h-32h) command to accommodate a maximum simultaneous two-plane array operation.

7. Issuing a READ PAGE CACHE-series (31h,00h-31h, 00h-32h, 3Fh) comand 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) comand 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. 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.

9.1. 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

10. 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 asynchronous interface is active by default for each target.
2. 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#.
3. 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.

11. Model Size

To reduce memory utilization each LUN has only 32 blocks but the actual device has 4096 blocks. Each MLC block is 1 MB or 8 Mb. If larger size is needed please contact Customer Support, or adjust the nb parameter in the <model>.vp file and resynthesize.

12. Limitations

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

13. 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
```

14. Compile and Emulation

The memory models are currently provided in one format: an encrypted RTL file(s) (*.vp) that targets use in the IXCOM flows 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+mt29f256g08cucbb \
../tb.v ../src/mt29e_32.vp \
../src/mt29f256g08cucbb.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/mt29e_32.vp ../src/mt29f256g08cucbb.vp

vaelab –keepRtlSymbol –keepAllFlipFlop –outputVlog mt29f256g08cucbb.vgp
mt29f256g08cucbb
```

14.1. Model file list

mt29f_16.vp - LUN or die module that is instantiated by ONFI 2.1 models. It is located under flash/nand/onfi_2.1 within MMP release.
mt29e_32.vp - LUN or die module that is instantiated by ONFI 2.1 models. It is located under flash/nand/onfi_2.1 within MMP release.
mt29f_32.vp - LUN or die module that is instantiated by ONFI 2.2 models. It is located under flash/nand/onfi_2.2 within MMP release.
mt29f_64.vp - LUN or die module that is instantiated by ONFI 2.2 and 2.3 models. It is located under flash/nand/onfi_2.2 within MMP release.

<model_name>.vp - model wrapper that instantiates one or more LUNs.

Revision History

The following table shows the revision history for this document

Date	Version	Revision
June 2010	1.0	Initial version
February 2011	1.1	Updated user guide
June 2011	1.2	Updated with “Related Documents” information
July 2014	1.3	Repaired doc title property. Added revision history. Updated legal.
September 2014	1.4	Remove version from UG file name. Add spec level to UG file name. Update UXE / IXE documentation reference titles.
November 2014	1.5	Remove emulation capacity info.
December 2014	1.6	Add emulation script info for IXCOM compile, including +sv switch. Update related publications list.
March 2015	1.7	Add timing parameter limitation and a section on tRHOH.
April 2015	1.8	Added ONFI 2.3 models and JEDEC commands. Replaced list of commands with features table.
July 2015	1.9	Update Cadence naming on front page
September 2015	2.0	Updated compile examples with .vp files. Removed references to <model_name>.vgp as an input file. Updated Limitations list.
January 2016	2.1	Update for Palladium-Z1 and VXE
February 2016	2.2	Added parameter page revision number to parameter page data file name.
July 2016	2.3	Remove hyphen in Palladium naming
January 2018	2.4	Modify header and footer
February 2018	2.5	Ported ECC section from NAND UG to ONFI and TDDR UGs
July 2018	2.6	Update for new utility library