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

NOR Flash
Palladium Memory Model
User Guide

Document Version: **2.1**

Document Date: **July 2018**

Contents

GENERAL INFORMATION **4**

1.1 RELATED PUBLICATIONS 4

NOR FLASH PALLADIUM MEMORY MODELS **5**

1. INTRODUCTION 5
2. MODEL RELEASE LEVELS 6
3. CONFIGURATIONS 7
4. MODEL LOGIC DIAGRAM 9
5. ADDRESS MAPPING 9
6. BLOCK NUMBER DISCREPANCY 10
7. ID OPERATIONS 11
   7.1. Auto Select *11*
   7.2. Read CFI Query *11*
8. COMMANDS 12
9. INITIALIZATION SEQUENCE 13
10. MODEL SIZE 13
11. LIMITATIONS 13
12. **COMPILE AND EMULATION** 13
    12.1. Model file list *14*

REVISION HISTORY **15**

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

NOR Flash Palladium Memory Models

1. Introduction

The Cadence Palladium NOR Flash Models are based on data sheet specifications of the following devices:

Numonyx M29W family of 3V Flash Memories (August 2010)
Spansion s29gl family of 3V Flash Memories (November 2009)
Spansion s70gl family of 3V Flash Memories (May 2007)
Numonyx Axcell M29EW family of 3V Flash Memories (May 2010)

Please note that some devices do not have uniform blocks where the top or bottom block is split into 4 or 8 smaller blocks, but the models implement all blocks at the same size as the main blocks. The smaller blocks are lump together into a main block. This will affect the block erase command. Please keep in mind that smaller size blocks will be erased together. In other words, smaller blocks cannot be erased individually. This discrepancy has a greater effect on the block protection feature; however, block protection is not supported. Please see section 5 for additional information.

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

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.

Numonyx Parts:

Model number	Density	Model Suffix	# of Banks	Block size	# of blocks	Modes
M29W128G	**128Mb**	**HL**	**1**	**128KB**	**128**	**x8, x16**
M29W160E	**16Mb**	**TB**	**1**	**64KB**	**32**	**x8, x16**
M29W320D	**32Mb**	**TB**	**1**	**64KB**	**64**	**x8, x16**
M29W320E	**32Mb**	**TB**	**1**	**64KB**	**64**	**x8, x16**
M29W400D	**4Mb**	**TB**	**1**	**64KB**	**8**	**x8, x16**
M29W640F	**64Mb**	**TB**	**1**	**64KB**	**128**	**x8, x16**
M29W640G	**64Mb**	**TBHL**	**1**	**64KB**	**128**	**x8, x16**
M29W800D	**8Mb**	**TB**	**1**	**64KB**	**16**	**x8, x16**
XX28F00BM29EW	**2Gb**	**H**	**1**	**128KB**	**2048**	**x8, x16**
XX28F00AM29EW	**1Gb**	**HL**	**1**	**128KB**	**1024**	**x8, x16**
XX28F512M29EW	**512Mb**	**HL**	**1**	**128KB**	**512**	**x8, x16**
XX28F256M29EW	**256MB**	**HL**	**1**	**128KB**	**256**	**x8, x16**

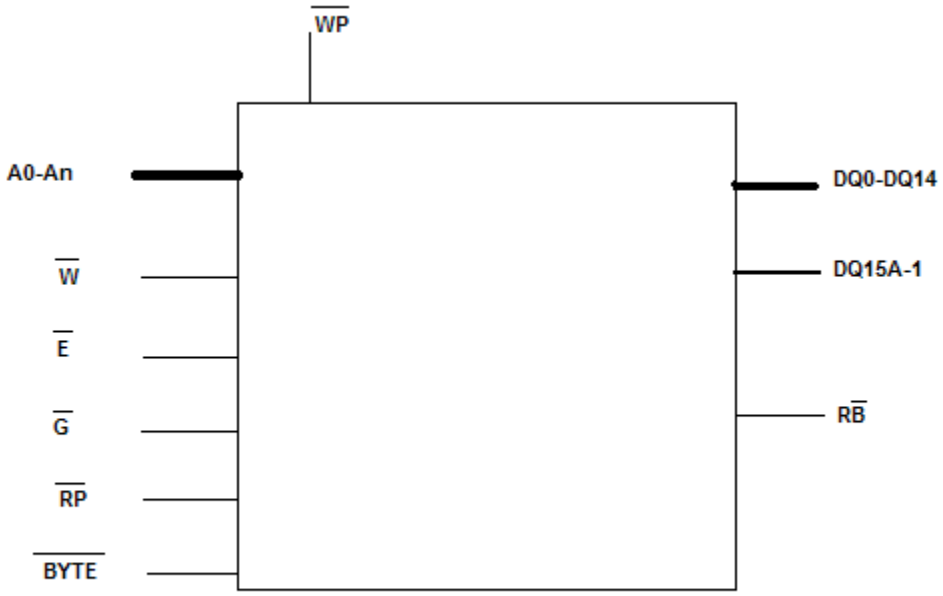
Suffix: T = Top block split into smaller boot and/or parameter blocks
B = Bottom block split into smaller boot and/or parameter blocks
H = High block can be protected by WP pin
L = Low block can be protected by WP pin

Spansion Parts:

Model number	Density	# of Banks	Block size	# of blocks	Modes
S29GL032N	**32Mb**	**1**	**64KB**	**64**	**x8, x16**
S29GL064N	**64Mb**	**1**	**64KB**	**128**	**x8, x16**
S29GL128N	**128Mb**	**1**	**128KB**	**128**	**x8, x16**
S29GL256N	**256Mb**	**1**	**128KB**	**256**	**x8, x16**
S29GL512N	**512Mb**	**1**	**128KB**	**512**	**x8, x16**
S70GL01GN	**1Gb**	**1**	**128KB**	**1024**	**x8, x16**
S29GL128P	**128Mb**	**1**	**128KB**	**128**	**x8, x16**
S29GL256P	**256Mb**	**1**	**128KB**	**256**	**x8, x16**
S29GL512P	**512Mb**	**1**	**128KB**	**512**	**x8, x16**
S29GL01GP	**1Gb**	**1**	**128KB**	**1024**	**x8, x16**
S70GL02GP	**2Gb**	**1**	**128KB**	**2048**	**x8, x16**

4. Model Logic Diagram

The lower density models 4Mb, 8Mb, and 16Mb do not have the WP pin.



5. Address Mapping

The array of the NOR Flash model is mapped into the internal memory of the Palladium system. This array is a single two dimensional array. In x16 mode the mapping of block and word addresses to the internal model array is as follows:

ARRAY_ADDR = {BA, WA}

In x8 mode, DQ15 is used to address the bytes within each word.

6. Block Number Discrepancy

As mentioned earlier there is a difference between device and model in block number referencing in devices that have smaller block sizes. Here are two tables showing the difference for the M29W400DB model.

Device address range per block number

Block #	Size (Kbytes)	Address range (x8)	Address range (x16)
10	64	70000h-7FFFFh	38000h-3FFFFh
9	64	60000h-6FFFFh	30000h-37FFFh
8	64	50000h-5FFFFh	28000h-2FFFFh
7	64	40000h-4FFFFh	20000h-27FFFh
6	64	30000h-3FFFFh	18000h-1FFFFh
5	64	20000h-2FFFFh	10000h-17FFFh
4	64	10000h-1FFFFh	08000h-0FFFFh
3	32	08000h-0FFFFh	04000h-07FFFh
2	8	06000h-07FFFh	03000h-03FFFh
1	8	04000h-05FFFh	02000h-02FFFh
0	16	00000h-03FFFh	00000h-01FFFh

Model address range per block number

Block #	Size (Kbytes)	Address range (x8)	Address range (x16)
7	64	70000h-7FFFFh	38000h-3FFFFh
6	64	60000h-6FFFFh	30000h-37FFFh
5	64	50000h-5FFFFh	28000h-2FFFFh
4	64	40000h-4FFFFh	20000h-27FFFh
3	64	30000h-3FFFFh	18000h-1FFFFh
2	64	20000h-2FFFFh	10000h-17FFFh
1	64	10000h-1FFFFh	08000h-0FFFFh
0	64	00000h-0FFFFh	00000h-07FFFh

7. ID Operations

7.1. Auto Select

The Manufacturer and Device codes have been hardcoded into each model. Therefore user data file is not required.

7.2. Read CFI Query

The CFI data for each device is provided in the <model_name>_cfi.dat file. This data file should be preloaded into the model if the user wants to read CFI information from the model. The model core's instance name is i1. Here is an example command to load the CFI data:

memory –load %readmemh <path.to.model.inst>.i1.CFI_ROM –file </path/to/m29w800db_cfi.dat

8. Commands

The NOR flash model accepts the following commands. However not all commands are available for all devices listed in this user guide. Please see each device's data sheet for available commands.

- Read/Reset
- Auto Select
- Program
- Chip Erase
- Block Erase
- Erase/Program Suspend
- Erase/Program Resume
- Read CFI Query
- Write to Buffer Program
- Write to Buffer Program Confirm
- Buffer Program Abort Reset
- Unlock Bypass
- Unlock Bypass Program
- Unlock Bypass Block Erase
- Unlock Bypass Chip Erase
- Unlock Bypass Write to Buffer Program
- Unlock Bypass Reset
- Enhanced Buffered Program
- Enhanced Buffered Program Confirm
- Unlock Bypass Enhanced Buffered Program
- Enter Extended Block
- Exit Extended Block
- Double Word Program
- Quadruple Word Program
- Double Byte Program
- Quadruple Byte Program
- Octuple Byte Program

Status register read during program and erase operations is supported. The RB pin can be monitored for Ready/Busy status.
Protection related commands are not supported; therefore all blocks are not protected.

9. Initialization Sequence

The Nor Flash model requires that the memory controller follows the initialization sequence as documented in the specification. The sequence basically entails the following steps:

1. Pulse the Reset RP pin to low to initialize the model. 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.

10. Model Size

Since the Nor flash devices listed in this user guide are in the Mbits range in size the models implement the full memory size without any size reduction at this time.

11. Limitations

1. Protection related commands are not supported.
2. Model does not check illegal sequence of command cycles.
3. Model does not check for attempts to program a bit to 1, user should make sure the block is erased before program.

12. 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+s29gl256n \
../tb.v ../src/m29w_core.vp \
../src/ s29gl256n.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/m29w_core.vp ../src/s29gl256n.vp

vaelab –keepRtlSymbol –keepAllFlipFlop –outputVlog s29gl256n.vgp s29gl256n
```

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.

12.1. Model file list

m29w_core.vp - LUN or die module that is instantiated by NOR models. It is located under flash/nor/numonyx 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
October 2010	1.0	Initial version
February 2011	1.1	Updated user guide
March 2011	1.2	Updated User Guide with Numonyx M29EW family.
July 2014	1.3	Repaired doc title property. Added revision history. Updated legal.
September 2014	1.4	Remove version from UG file name. Update UXE / IXE documentation reference titles.
November 2014	1.5	Remove emulation capacity info. Update related publications list.
July 2015	1.6	Update Cadence naming on front page
September 2015	1.7	Added Emulation section with compile examples.
January 2016	1.8	Update for Palladium-Z1 and VXE
July 2016	1.9	Remove hyphen in Palladium naming
January 2018	2.0	Add data sheet dates. Modify header and footer
July 2018	2.1	Update for new utility library