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

# SPI Flash
# Palladium Memory Model
# User Guide

**Document Version:** 2.2

**Document Date:** January 2018

# Contents



## List of Figures



## List of Tables

# 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

# SPI Flash Memory Model

## 1. Introduction

The Cadence Palladium SPI Flash Memory models are available with model sizes matching real SPI flashes manufactured by Numonyx or Macronix. Different sizes are available; please consult the memory model catalog for the current available list.

Available SPI Flash Memory models:

Numonyx M25Pxxx series:

- m25p05a : Numonyx M25P05-A SPI Flash model
- m25p10a : Numonyx M25P10-A SPI Flash model
- m25p16 : Numonyx M25P16 SPI Flash model
- m25p20 : Numonyx M25P20 SPI Flash model
- m25p32 : Numonyx M25P32 SPI Flash model
- m25p40 : Numonyx M25P40 SPI Flash model
- m25p64 : Numonyx M25P64 SPI Flash model
- m25p80 : Numonyx M25P80 SPI Flash model
- m25p128 : Numonyx M25P128 SPI Flash model

Numonyx M25PExxx series:

- m25pe10 : Numonyx M25PE10 SPI Flash model
- m25pe16 : Numonyx M25PE16 SPI Flash model
- m25pe20 : Numonyx M25PE20 SPI Flash model
- m25pe40 : Numonyx M25PE40 SPI Flash model
- m25pe80 : Numonyx M25PE80 SPI Flash model

Numonyx M25PXxxx series:

- m25px16 : Numonyx M25PX16 SPI Flash model
- m25px32 : Numonyx M25PX32 SPI Flash model
- m25px64 : Numonyx M25PX64 SPI Flash model
- m25px80 : Numonyx M25PX80 SPI Flash model

Numonyx M45PExxx series:

- m45pe10 : Numonyx M45PE10 SPI Flash model
- m45pe16 : Numonyx M45PE16 SPI Flash model
- m45pe20 : Numonyx M45PE20 SPI Flash model
- m45pe40 : Numonyx M45PE40 SPI Flash model
- m45pe80 : Numonyx M45PE80 SPI Flash model

Macronix MX25Lxxx6E series:

- mx25l5126e : Macronix MX25L5126E SPI Flash model
- mx25l1006e : Macronix MX25L1006E SPI Flash model
- mx25l2006e : Macronix MX25L2006E SPI Flash model
- mx25l4006e : Macronix MX25L4006E SPI Flash model

- mx25l8006e : Macronix MX25L8006E SPI Flash model
- mx25l1606e : Macronix MX25L1606E SPI Flash model
- mx25l3206e : Macronix MX25L3206E SPI Flash model
- mx25l6406e : Macronix MX25L6406E SPI Flash model

Note: The user guide only covers the SPI Flash Memory models of the Numonyx and Macronix's Dual IO and Single IO SPI flashes. For information about the Quad SPI Flash Memory models, please refer to the *Quad SPI Flash Model User Guide*.

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

# 3. Configurations

The SPI Flash models are in compliance with the data sheet of the corresponding SPI parts. The following table lists some configurations for the models. Only a few features are included in the table. For more details, please see the SPI flash data sheet from the vendor.

**Table 1: SPI Flash Model Configurations**

| Model Number | Density | Configure | Feature |
|---|---|---|---|
| M25P05A | 512K bit | 64K * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P10A | 1M bit | 128K * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P20 | 2M bit | 256K * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P40 | 4M bit | 512K * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P80 | 8M bit | 1M * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P16 | 16M bit | 2M * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P32 | 32M bit | 4M * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P64 | 64M bit | 8M * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25P128 | 128M bit | 16M * 8 | Hardware Write Protection, Bulk Erase, Hold pin |
| M25PE10 | 1M bit | 128K * 8 | Page Erase, Sub Sector Erase, Bulk Erase, Reset Pin |
| M25PE20 | 2M bit | 256K * 8 | Page Erase, Sub Sector Erase, Bulk Erase, Reset Pin |
| M25PE40 | 4M bit | 512K * 8 | Page Erase, Sub Sector Erase, Bulk Erase, Reset Pin |
| M25PE80 | 8M bit | 1M * 8 | Page Erase, Sub Sector Erase, Bulk Erase, Reset Pin |
| M25PE16 | 16M bit | 2M * 8 | Page Erase, Sub Sector Erase, Bulk Erase, Reset Pin |
| M25PX80 | 8M bit | 1M * 8 | Dual IO, Software & Hardware Write Protect, OTP, Hold Pin |
| M25PX16 | 16M bit | 2M * 8 | Dual IO, Software & Hardware Write Protect, OTP, Hold Pin |
| M25PX32 | 32M bit | 4M * 8 | Dual IO, Software & Hardware Write Protect, OTP, Hold Pin |
| M25PX64 | 64M bit | 8M * 8 | Dual IO, Software & Hardware Write Protect, OTP, Hold Pin |
| M45PE10 | 1M bit | 128K * 8 | Hardware Write Protect of Bottom Sector, Reset Pin |
| M45PE20 | 2M bit | 256K * 8 | Hardware Write Protect of Bottom Sector, Reset Pin |
| M45PE40 | 4M bit | 512K * 8 | Hardware Write Protect of Bottom Sector, Reset Pin |
| M45PE80 | 8M bit | 1M * 8 | Hardware Write Protect of Bottom Sector, Reset Pin |
| M45PE16 | 16M bit | 2M * 8 | Hardware Write Protect of Bottom Sector, Reset Pin |
| MX25L5126E | 512K bit | 64K * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L1006E | 1M bit | 128K * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L2006E | 2M bit | 256K * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L4006E | 4M bit | 512K * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L8006E | 8M bit | 1M * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L1606E | 16M bit | 2M * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L3206E | 32M bit | 4M * 8 | Hardware/Software Write Protect, Hold Pin |
| MX25L6406E | 64M bit | 8M * 8 | Hardware/Software Write Protect, Hold Pin |

**Table 2: Feature List for Numonyx Models**

| Features | | Supported | Notes |
|---|---|---|---|
| Identification Operations | Read ID | Yes | |
| Read Operations | Read | Yes | |
| | Fast Read | Yes | |
| Write Operations | Write Enable | Yes | |
| | Write Disable | Yes | |
| Register Operations | Read Status Register | Yes | |
| | Write Status Register | Yes | |
| | Read Lock Register | Yes | |
| | Write Lock Register | Yes | |
| Power Operations | Power Down | Yes | |
| | Release Power Down | Yes | |
| Program Operations | Page Program | Yes | |
| Erase Operations | Page Erase | Yes | |
| | Sub-Sector Erase | Yes | |
| | Sector Erase | Yes | |
| | Bulk Erase | Yes | |
| OTP Operations | Read OTP array | Some Models | 1 |
| | Program OTP array | Some Models | 1 |

Notes:
1. Only the M25PXxxx and MX25Lxxx6E devices have the OTP memory

**Table 3: Feature List for Macronix Models**

| Features | | Supported | Notes |
|---|---|---|---|
| Identification Operations | Read ID | Yes | |
| Read Operations | Read | Yes | |
| | Fast Read | Yes | |
| | Dual Output Read | Yes | |
| Write Operations | Write Enable | Yes | |
| | Write Disable | Yes | |
| Register Operations | Read Status Register | Yes | |
| | Write Status Register | Yes | |
| | Read Security Register | Some Models | 1 |
| | Write Security Register | Some Models | 1 |
| Power Operations | Power Down | Yes | |
| | Release Power Down | Yes | |
| Program Operations | Page Program | Yes | |
| Erase Operations | Sector Erase | Yes | |
| | Block Erase | Yes | |
| | Chip Erase | Yes | |
| OTP Operations | Enter OTP | Yes | |
| | Exit OTP | Yes | |
| | Read OTP array | Yes | |
| | Program OTP array | Yes | |

Notes:
1. Security register is initialized to all zeroes

# 4. Model Block Diagram

## 4.1 Interface Diagram

There are 2 different types of interface for the Numonyx SPI flashes, one with HOLD_n pin and the other with Reset_n pin. For the use of these pins, please refer to the flash data sheet.

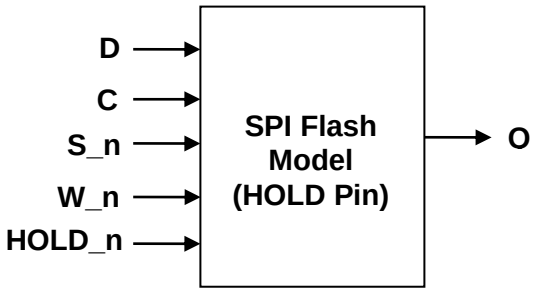


**Figure 1: Logic Diagram with HOLD Pin for M25Pxxx & M25PXxxx**

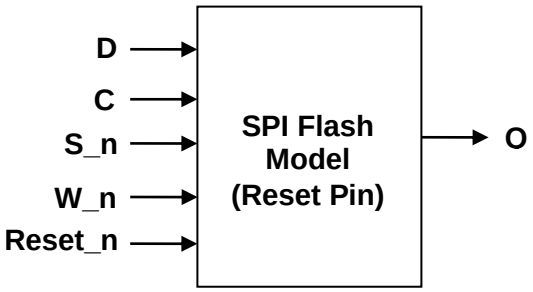


**Figure 2: Logic Diagram with Reset Pin for M25PExxx & M45PExxx**

**Table 4: Signal Names**

| Signal | Description | I/O |
|---|---|---|
| C | Serial clock | Input |
| D | Serial data input | Input |
| Q | Serial data output | Output |
| S_n | Chip select | Input |
| W_n | Write protect | Input |
| Reset | Reset | Input |
| HOLD_n | Hold | Input |

### 4.2 Connection Diagram

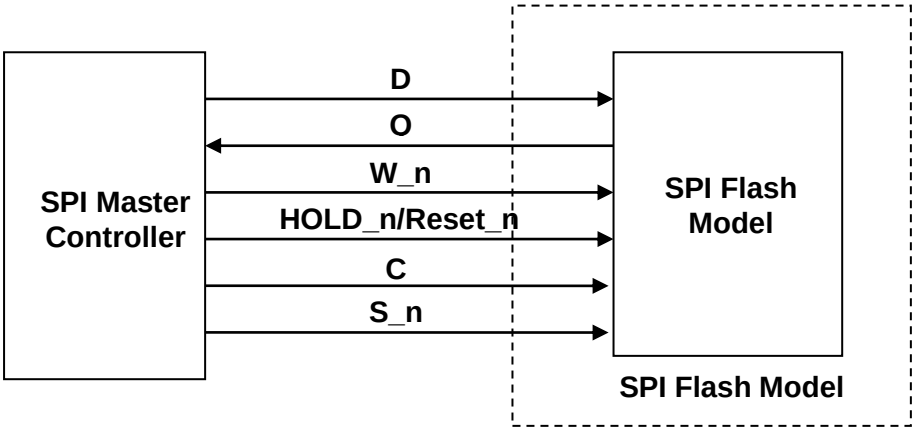


**Figure 3: Connection Diagram**

## 5. Limitations

There are some limitations for this version of the models, which are listed below:

- Initial content of flash memory
  After download into the Palladium the initial value of the flash memory array is all 0's. Like real Flash parts the model only allows programming of 1's to 0's. The memory region of the Flash array being programmed must first either be erased with an erase command or initialized to all 1's through the user issuing a memory set command in Palladium runtime tools.

- Initial content of OTP memory
  The initial value of the OTP memory is all 0's. The user may need to set the content of the OTP memory to all 1's in Palladium tools. Note that only the M25PXxxx and MX25Lxxx6E devices have the OTP memory.

# 6. Model Emulation

## 6.1 Emulation Notes

Some things needed to be kept in mind when running emulation with the SPI Flash model in Palladium.

- The SPI Flash models are delivered as protected vhdl source with the top-level declaration unprotected and the flash memory and OTP memory unprotected.
  - The main data array is called mem_array
  - The OTP array is called mem_otp (only available for M25PXxxx and MX25Lxxx6E device).

- The top-level module in the SPI Flash modules is the part number, for example M25P20. To use the model in your design, just instantiate the top-level module and map the ports to your actual wires. For more details of the top-level declaration, please open the protected vhdl source with a text viewer and search the top-level module name.

- The initial content of flash memory and OTP memory is all ‘0’ at start-up, which will cause the OTP memory to be locked after Palladium emulation is started, and program to the flash memory will not work. The content of flash memory and OTP can be initialize to all ‘1’ in Memory tab of questDebug GUI or by issuing commands in the questQel console as below:

```
QEL> memory -set <dspath>.mem_array
QEL> memory -set <dspath>.mem_otp
```

  To check the full path of the mem_array or mem_otp, please go to the Memory tab in questDebug, or invoke the memory command in the questQEL console. Below is an example to get the memory path in the design.

```
QEL> memory
flash_tb.m25p05a _i.m25p05a_core_i.mem_array
flash_tb.m25p05a _i.m25p05a_core_i.mem_otp
```

  For those devices supporting Bulk Erase command, the user can also erase the flash memory to all ‘1’ using the Bulk Erase instruction after emulation has been started.

## 6.2 Emulation Example

Below is an example of the command list to synthesize the protected RTL files (*.vhdp) into the netlist (*.vgp):

```
vavlog  mmp_spi_clk_gen.v
vavhdl  m25p05a.vhdp
vaelab  –keepRtlSymbol –keepAllFlipFlop –outputvlog m25p05a.vgp m25p05a
```

**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.

Below is an example of the command list for importing the SPI Flash model to your design:

```
QEL> delimiterRule Verilog
QEL> importOption {mode full} {protection on} {library libf}
QEL> reflib qtref
QEL> netlistFile "verilog m25p05a.vgp"
QEL> designImport
QEL> importOption {protection on} {library libf}
QEL> reflib qtref libf
QEL> netlistFile "verilog <your design netlists>"
QEL> designImport
```

## 6.3 IXCOM Notes

For the IXCOM the SPI model requires an internal clock generator. The module uses an IXCOM clock generator which is SystemVerilog based. As part of the IXCOM compile flow you must include the IXCOMClkGen.sv file from the UXE installation. The mmp_spi_clk_gen.v file can be found in the <MMP install>/common/ directory.

Below is an example.

```
vlan mmp_spi_clk_gen.v

vhan m25p40.vhdp

Other design files…

ixcom -ua -top tb +dut+m25p40
```

Several enhancements to the SPI family compile are introduced.

- For earlier versions of the models, the option +rtlCommentPragma was a required option for the IXCOM flow and as such was shown in the vhan command above. The +rtlCommentPragma option is no longer required.

- For earlier versions of the models, the top model name was “flash” and now the top model name is the same as the part name and file name. Thus, in the ixcom command example above +dut+flash is now +dut+m25p40.

- For earlier versions of the models, the –work command was used to specify a vhdlaux_lib as a separate logical library in which to compile instead of the default logical library called WORK. That separate logical library is no longer required. The model compile uses the default logical library WORK. Hence the option –work is removed from the example above.

- The previous requirement for a separate logical library also necessitated the –libmaprc option to specify a non-default libmaprc script to specify the vhdl_aux to path for vhdl_aux mapping (vhdlaux_lib => ./vhdlaux_lib). The removal of the separate logical library also means that the –libmaprc option is no longer required to specify this mapping. Hence the –libmaprc option has been removed from the example compile.

# Revision History

The following table shows the revision history for this document

| Date | Version | Revision |
|---|---|---|
| June 2011 | 1.0 | Initial version |
| April 2012 | 1.1 | Update the SPI flash model user guide to add MX25Lxxx6E small models |
| March 2013 | 1.2 | SPI Docs update to add IXCOM stuff |
| July 2014 | 1.3 | Repaired doc title property. Added revision history. Updated legal. |
| September 2014 | 1.5 | Remove version from UG file name. Update UXE / IXE documentation reference titles. |
| December 2014 | 1.6 | Rename top module from flash to flash part number<br>+rtlCommentPragma not needed for IXCOM flow<br>Updated IXCOM compile section. Update related publications list. |
| March 2015 | 1.7 | Update supported features and unsupported features list |
| July 2015 | 1.8 | Update Cadence naming on front page |
| August 2015 | 1.9 | Add example to check the full path of internal memory. Add synthesis options note. |
| January 2016 | 2.0 | Update for Palladium-Z1 and VXE |
| July 2016 | 2.1 | Add mmp_spi_clk_gen.v location information<br>Remove hyphen from palladium naming |
| January 2018 | 1.8 | Modify header and footer |