



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



# Micron M25P16 Serial Flash Embedded Memory

16Mb, 3V

## Features

- SPI bus compatible serial interface
- 16Mb Flash memory
- 75 MHz clock frequency (maximum)
- 2.7V to 3.6V single supply voltage
- Page program (up to 256 bytes) in 0.64ms (TYP)
- Erase capability
  - Sector erase: 512Kb in 0.6 s (TYP)
  - Bulk erase: 16Mb in 13 s (TYP)
- Write protection
  - Hardware write protection: protected area size defined by non-volatile bits BP0, BP1, BP2
- Deep power down: 1 $\mu$ A (TYP)
- Electronic signature
  - JEDEC standard 2-byte signature (2015h)
  - Unique ID code (UID) and 16 bytes of read-only data, available upon customer request
  - RES command, one-byte signature (14h) for backward compatibility
- More than 100,000 write cycles per sector
- More than 20 years data retention
- Automotive grade parts available
- Packages (RoHS compliant)
  - SO8N (MN) 150 mils
  - SO8W (MW) 208 mils
  - SO16 (MF) 300 mils
  - VFDFPN8 (MP) MLP8 6mm x 5mm
  - VFDFPN8 (ME) MLP8 8mm x 6mm
  - UFDFPN8 (MC) MLP8 4mm x 3mm



## Contents

Functional Description .....	6
Signal Descriptions .....	8
SPI Modes .....	9
Operating Features .....	11
Page Programming .....	11
Sector Erase, Bulk Erase .....	11
Polling during a Write, Program, or Erase Cycle .....	11
Active Power, Standby Power, and Deep Power-Down .....	11
Status Register .....	12
Data Protection by Protocol .....	12
Software Data Protection .....	12
Hardware Data Protection .....	12
Hold Condition .....	13
Configuration and Memory Map .....	14
Memory Configuration and Block Diagram .....	14
Memory Map – 16Mb Density .....	15
Command Set Overview .....	16
WRITE ENABLE .....	18
WRITE DISABLE .....	19
READ IDENTIFICATION .....	20
READ STATUS REGISTER .....	22
WIP Bit .....	23
WEL Bit .....	23
Block Protect Bits .....	23
SRWD Bit .....	23
SRWD Bit .....	23
WRITE STATUS REGISTER .....	24
READ DATA BYTES .....	26
READ DATA BYTES at HIGHER SPEED .....	27
PAGE PROGRAM .....	28
SECTOR ERASE .....	29
BULK ERASE .....	30
DEEP POWER-DOWN .....	31
RELEASE from DEEP POWER-DOWN .....	32
READ ELECTRONIC SIGNATURE .....	33
Power-Up/Down and Supply Line Decoupling .....	34
Power-Up Timing and Write Inhibit Voltage Threshold Specifications .....	36
Maximum Ratings and Operating Conditions .....	37
Electrical Characteristics .....	38
AC Characteristics .....	39
Package Information .....	45
Device Ordering Information .....	51
Standard Parts .....	51
Automotive Parts .....	52
Revision History .....	54
Rev. I – 4/15 .....	54
Rev. H – 1/14 .....	54
Rev. G – 1/13 .....	54
Rev. F – 10/12 .....	54
Rev. E – 8/12 .....	54



# Micron M25P16 Serial Flash Embedded Memory Features

---

Rev. D – 5/12 .....	54
Rev. C – 3/12 .....	54
Rev. B – 3/12 .....	54
Rev. A – 9/2011 .....	55

## List of Figures

Figure 1: Logic Diagram .....	6
Figure 2: Pin Connections: SO8, VFQFPN , VDFPN .....	7
Figure 3: Pin Connections: SO16 .....	7
Figure 4: SPI Modes Supported .....	9
Figure 5: Bus Master and Memory Devices on the SPI Bus .....	10
Figure 6: Hold Condition Activation .....	13
Figure 7: Block Diagram .....	14
Figure 8: WRITE ENABLE Command Sequence .....	18
Figure 9: WRITE DISABLE Command Sequence .....	19
Figure 10: READ IDENTIFICATION Command Sequence .....	21
Figure 11: READ STATUS REGISTER Command Sequence .....	22
Figure 12: Status Register Format .....	22
Figure 13: WRITE STATUS REGISTER Command Sequence .....	24
Figure 14: READ DATA BYTES Command Sequence .....	26
Figure 15: READ DATA BYTES at HIGHER SPEED Command Sequence .....	27
Figure 16: PAGE PROGRAM Command Sequence .....	28
Figure 17: SECTOR ERASE Command Sequence .....	29
Figure 18: BULK ERASE Command Sequence .....	30
Figure 19: DEEP POWER-DOWN Command Sequence .....	31
Figure 20: RELEASE from DEEP POWER-DOWN Command Sequence .....	32
Figure 21: READ ELECTRONIC SIGNATURE Command Sequence .....	33
Figure 22: Power-Up Timing .....	35
Figure 23: AC Measurement I/O Waveform .....	39
Figure 24: Serial Input Timing .....	43
Figure 25: Write Protect Setup and Hold during WRSR when SRWD=1 Timing .....	43
Figure 26: Hold Timing .....	44
Figure 27: Output Timing .....	44
Figure 28: SO8N 150 mils Body Width .....	45
Figure 29: SO8W 208 mils Body Width .....	46
Figure 30: SO16W 300 mils Body Width .....	47
Figure 31: VFDFPN8 (MLP8) 6mm x 5mm .....	48
Figure 32: VFDFPN8 (MLP8) 8mm x 6mm .....	49
Figure 33: UFDFPN8 (MLP8) 4mm x 3mm .....	50



## List of Tables

Table 1: Signal Names .....	6
Table 2: Signal Descriptions .....	8
Table 3: Protected Area Sizes .....	12
Table 4: Sectors 31:0 .....	15
Table 5: Command Set Codes .....	17
Table 6: READ IDENTIFICATION Data Out Sequence .....	20
Table 7: Status Register Protection Modes .....	25
Table 8: Power-Up Timing and $V_{WI}$ Threshold .....	36
Table 9: Absolute Maximum Ratings .....	37
Table 10: Operating Conditions .....	37
Table 11: Data Retention and Endurance .....	37
Table 12: DC Current Specifications .....	38
Table 13: DC Voltage Specifications .....	38
Table 14: AC Measurement Conditions .....	39
Table 15: Capacitance .....	39
Table 16: AC Specifications (25 MHz, Device Grade 3, $V_{CC}[\min]=2.7V$ ) .....	39
Table 17: Instruction Times (25 MHz, Device Grade 3, $V_{CC}[\min]=2.7V$ ) .....	40
Table 18: AC Specifications (75 MHz, Device Grade 3 and 6, $V_{CC}[\min]=2.7V$ ) .....	41
Table 19: Instruction Times (75 MHz, Device Grade 3 and 6, $V_{CC}[\min]=2.7V$ ) .....	42
Table 20: Part Number Example .....	51
Table 21: Part Number Information Scheme .....	51
Table 22: Part Number Example .....	52
Table 23: Part Number Information Scheme .....	52

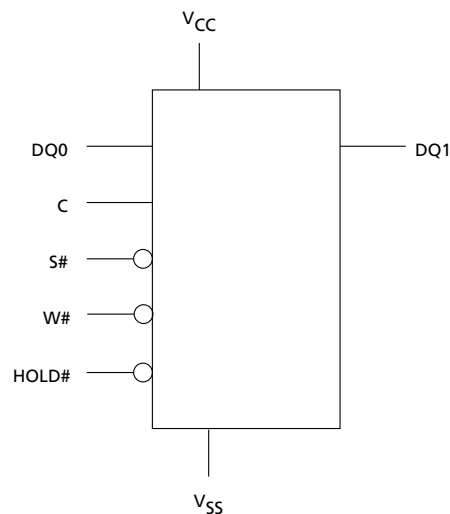
## Functional Description

The M25P16 is an 16Mb (2Mb x 8) serial Flash memory device with advanced write protection mechanisms accessed by a high speed SPI-compatible bus. The device supports high-performance commands for clock frequency up to 75MHz.

The memory can be programmed 1 to 256 bytes at a time using the PAGE PROGRAM command. It is organized as 32 sectors, each containing 256 pages. Each page is 256 bytes wide. Memory can be viewed either as 8,192 pages or as 2,097,152 bytes. The entire memory can be erased using the BULK ERASE command, or it can be erased one sector at a time using the SECTOR ERASE command.

This datasheet details the functionality of the M25P16 device based on 110nm process.

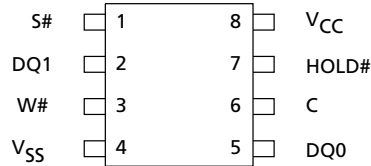
**Figure 1: Logic Diagram**



**Table 1: Signal Names**

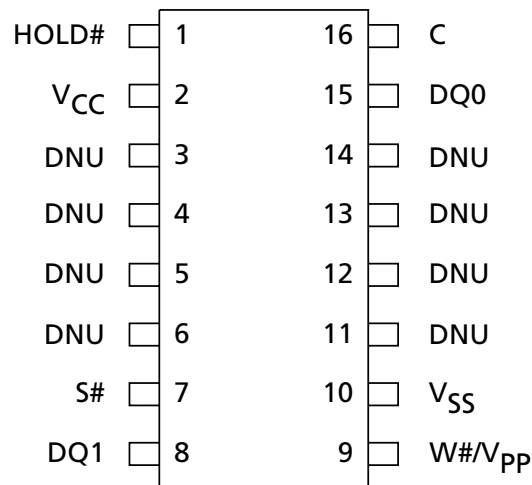
Signal Name	Function	Direction
C	Serial clock	Input
DQ0	Serial data input	Input
DQ1	Serial data output	Output
S#	Chip select	Input
W#	Write protect	Input
HOLD#	Hold	Input
V <sub>CC</sub>	Supply voltage	-
V <sub>SS</sub>	Ground	-

**Figure 2: Pin Connections: SO8, VFQFPN , VDFPN**



- Note: 1. There is an exposed central pad on the underside of the MLP8 package that is pulled internally to V<sub>SS</sub>, and must not be connected to any other voltage or signal line on the PCB. The Package Mechanical section provides information on package dimensions and how to identify pin 1.

**Figure 3: Pin Connections: SO16**



- Notes: 1. DU = Don't Use  
2. The Package Mechanical section provides information on package dimensions and how to identify pin 1.



## Signal Descriptions

**Table 2: Signal Descriptions**

Signal	Type	Description
DQ1	Output	<b>Serial data:</b> The DQ1 output signal is used to transfer data serially out of the device. Data is shifted out on the falling edge of the serial clock (C).
DQ0	Input	<b>Serial data:</b> The DQ0 input signal is used to transfer data serially into the device. It receives commands, addresses, and the data to be programmed. Values are latched on the rising edge of the serial clock (C).
C	Input	<b>Clock:</b> The C input signal provides the timing of the serial interface. Commands, addresses, or data present at serial data input (DQ0) is latched on the rising edge of the serial clock (C). Data on DQ1 changes after the falling edge of C.
S#	Input	<b>Chip select:</b> When the S# input signal is HIGH, the device is deselected and DQ1 is at HIGH impedance. Unless an internal PROGRAM, ERASE, or WRITE STATUS REGISTER cycle is in progress, the device will be in the standby power mode (not the DEEP POWER-DOWN mode). Driving S# LOW enables the device, placing it in the active power mode. After power-up, a falling edge on S# is required prior to the start of any command.
HOLD#	Input	<b>Hold:</b> The HOLD# signal is used to pause any serial communications with the device without deselecting the device. During the hold condition, DQ1 is High-Z. DQ0 and C are "Don't Care." To start the hold condition, the device must be selected, with S# driven LOW.
W#	Input	<b>Write protect:</b> The W# input signal is used to freeze the size of the area of memory that is protected against program or erase commands as specified by the values in BP2, BP1, and BP0 bits of the Status Register.
V <sub>CC</sub>	Power	<b>Device core power supply:</b> Source voltage.
V <sub>SS</sub>	Ground	<b>Ground:</b> Reference for the V <sub>CC</sub> supply voltage.

## SPI Modes

These devices can be driven by a microcontroller with its serial peripheral interface (SPI) running in either of the following two SPI modes:

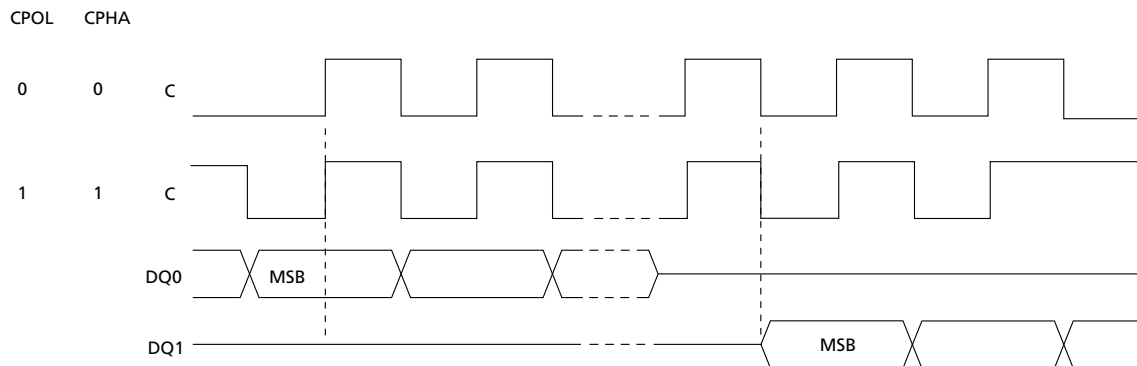
- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

For these two modes, input data is latched in on the rising edge of serial clock (C), and output data is available from the falling edge of C.

The difference between the two modes is the clock polarity when the bus master is in STANDBY mode and not transferring data:

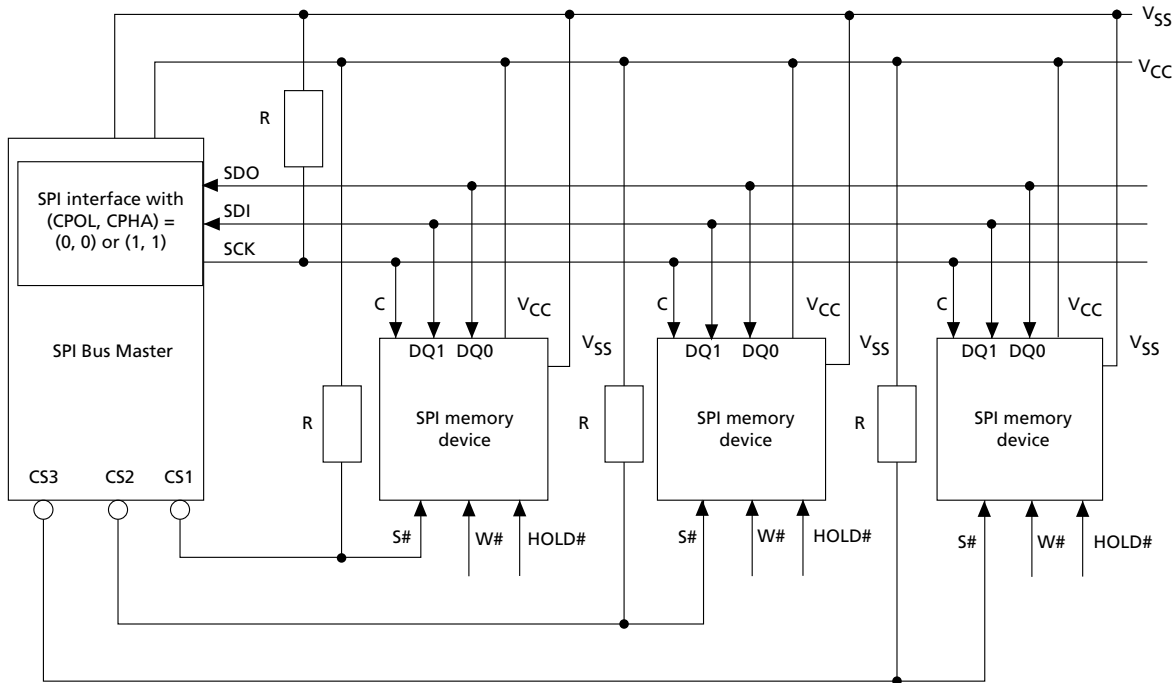
- C remains at 0 for (CPOL=0, CPHA=0)
- C remains at 1 for (CPOL=1, CPHA=1)

**Figure 4: SPI Modes Supported**



Because only one device is selected at a time, only one device drives the serial data output (DQ1) line at a time, while the other devices are HIGH-Z. An example of three devices connected to an MCU on an SPI bus is shown here.

**Figure 5: Bus Master and Memory Devices on the SPI Bus**



- Notes:
1. WRITE PROTECT (W#) and HOLD# should be driven HIGH or LOW as appropriate.
  2. Resistors (R) ensure that the memory device is not selected if the bus master leaves the S# line HIGH-Z.
  3. The bus master may enter a state where all I/O are HIGH-Z at the same time; for example, when the bus master is reset. Therefore, C must be connected to an external pull-down resistor so that when all I/O are HIGH-Z, S# is pulled HIGH while C is pulled LOW. This ensures that S# and C do not go HIGH at the same time and that the <sup>t</sup>SHCH requirement is met.
  4. The typical value of R is 100 kΩ, assuming that the time constant  $R \times C_p$  ( $C_p$  = parasitic capacitance of the bus line) is shorter than the time during which the bus master leaves the SPI bus HIGH-Z.
  5. Example: Given that  $C_p = 50$  pF ( $R \times C_p = 5\mu$ s), the application must ensure that the bus master never leaves the SPI bus HIGH-Z for a time period shorter than 5μs.

## Operating Features

### Page Programming

To program one data byte, two commands are required: WRITE ENABLE, which is one byte, and a PAGE PROGRAM sequence, which is four bytes plus data. This is followed by the internal PROGRAM cycle of duration  $t_{PP}$ . To spread this overhead, the PAGE PROGRAM command allows up to 256 bytes to be programmed at a time (changing bits from 1 to 0), provided they lie in consecutive addresses on the same page of memory. To optimize timings, it is recommended to use the PAGE PROGRAM command to program all consecutive targeted bytes in a single sequence than to use several PAGE PROGRAM sequences with each containing only a few bytes.

### Sector Erase, Bulk Erase

The PAGE PROGRAM command allows bits to be reset from 1 to 0. Before this can be applied, the bytes of memory need to have been erased to all 1s (FFh). This can be achieved a sector at a time using the SECTOR ERASE command, or throughout the entire memory using the BULK ERASE command. This starts an internal ERASE cycle of duration  $t_{SE}$  or  $t_{BE}$ . The ERASE command must be preceded by a WRITE ENABLE command.

### Polling during a Write, Program, or Erase Cycle

An improvement in the time to complete the following commands can be achieved by not waiting for the worst case delay ( $t_W$ ,  $t_{PP}$ ,  $t_{SE}$ , or  $t_{BE}$ ).

- WRITE STATUS REGISTER
- PROGRAM
- ERASE (SECTOR ERASE, BULK ERASE)

The write in progress (WIP) bit is provided in the status register so that the application program can monitor this bit in the status register, polling it to establish when the previous WRITE cycle, PROGRAM cycle, or ERASE cycle is complete.

### Active Power, Standby Power, and Deep Power-Down

When chip select ( $S\#$ ) is LOW, the device is selected, and in the ACTIVE POWER mode. When  $S\#$  is HIGH, the device is deselected, but could remain in the ACTIVE POWER mode until all internal cycles have completed (PROGRAM, ERASE, WRITE STATUS REGISTER). The device then goes in to the STANDBY POWER mode. The device consumption drops to  $I_{CC1}$ .

The DEEP POWER-DOWN mode is entered when the DEEP POWER-DOWN command is executed. The device consumption drops further to  $I_{CC2}$ . The device remains in this mode until the RELEASE FROM DEEP POWER-DOWN command is executed. While in the DEEP POWER-DOWN mode, the device ignores all WRITE, PROGRAM, and ERASE commands. This provides an extra software protection mechanism when the device is not in active use, by protecting the device from inadvertent WRITE, PROGRAM, or ERASE operations. For further information, see the DEEP POWER DOWN command.

## Status Register

The status register contains a number of status and control bits that can be read or set (as appropriate) by specific commands. For a detailed description of the status register bits, see READ STATUS REGISTER (page 22).

## Data Protection by Protocol

Non-volatile memory is used in environments that can include excessive noise. The following capabilities help protect data in these noisy environments.

Power on reset and an internal timer ( $t_{PUW}$ ) can provide protection against inadvertent changes while the power supply is outside the operating specification.

PROGRAM, ERASE, and WRITE STATUS REGISTER commands are checked before they are accepted for execution to ensure they consist of a number of clock pulses that is a multiple of eight.

All commands that modify data must be preceded by a WRITE ENABLE command to set the write enable latch (WEL) bit.

In addition to the low power consumption feature, the DEEP POWER-DOWN mode offers extra software protection since all PROGRAM, and ERASE commands are ignored when the device is in this mode.

## Software Data Protection

Memory can be configured as read-only using the block protect bits (BP2, BP1, BP0) as shown in the Protected Area Sizes table.

## Hardware Data Protection

Hardware data protection is implemented using the write protect signal applied on the W# pin. This freezes the status register in a read-only mode. In this mode, the block protect (BP) bits and the status register write disable bit (SRWD) are protected.

**Table 3: Protected Area Sizes**

Status Register Content			Memory Content	
BP Bit 2	BP Bit 1	BP Bit 0	Protected Area	Unprotected Area
0	0	0	none	All sectors (sectors 0 to 31)
0	0	1	Upper 32nd (sector 31)	Lower 31/32nds (sectors 0 to 30)
0	1	0	Upper 16th (sectors 30 and 31)	Lower 15/16ths (sectors 0 to 29)
0	1	1	Upper 8th (sectors 28 to 31)	Lower 7/8ths (sectors 0 to 27)
1	0	0	Upper 4th (sectors 24 to 31)	Lower 3/4ths (sectors 0 to 23)
1	0	1	Upper half (sectors 16 to 31)	Lower half (sectors 0 to 15)
1	1	0	All sectors (sectors 0 to 31)	none
1	1	1	All sectors (sectors 0 to 31)	none

Note: 1. 0 0 0 = unprotected area (sectors): The device is ready to accept a BULK ERASE command only if all block protect bits (BP2, BP1, BP0) are 0.

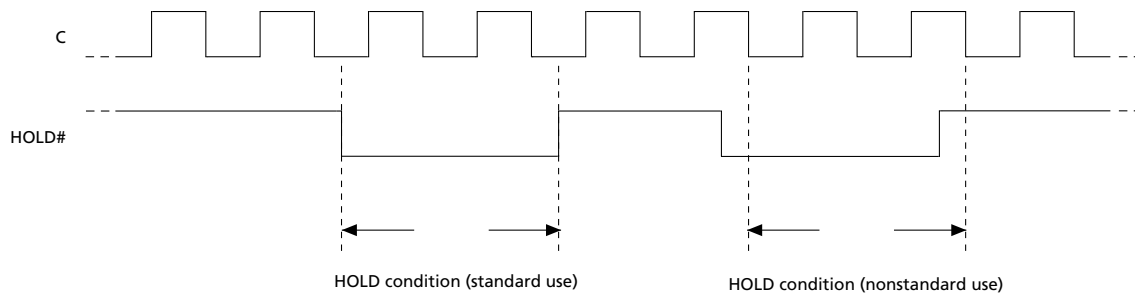
## Hold Condition

The HOLD# signal is used to pause any serial communications with the device without resetting the clocking sequence. However, taking this signal LOW does not terminate any WRITE STATUS REGISTER, PROGRAM, or ERASE cycle that is currently in progress.

To enter the hold condition, the device must be selected, with S# LOW. The hold condition starts on the falling edge of the HOLD# signal, if this coincides with serial clock (C) being LOW. The hold condition ends on the rising edge of the HOLD# signal, if this coincides with C being LOW. If the falling edge does not coincide with C being LOW, the hold condition starts after C next goes LOW. Similarly, if the rising edge does not coincide with C being LOW, the hold condition ends after C next goes LOW.

During the hold condition, DQ1 is HIGH impedance while DQ0 and C are Don't Care. Typically, the device remains selected with S# driven LOW for the duration of the hold condition. This ensures that the state of the internal logic remains unchanged from the moment of entering the hold condition. If S# goes HIGH while the device is in the hold condition, the internal logic of the device is reset. To restart communication with the device, it is necessary to drive HOLD# HIGH, and then to drive S# LOW. This prevents the device from going back to the hold condition.

**Figure 6: Hold Condition Activation**



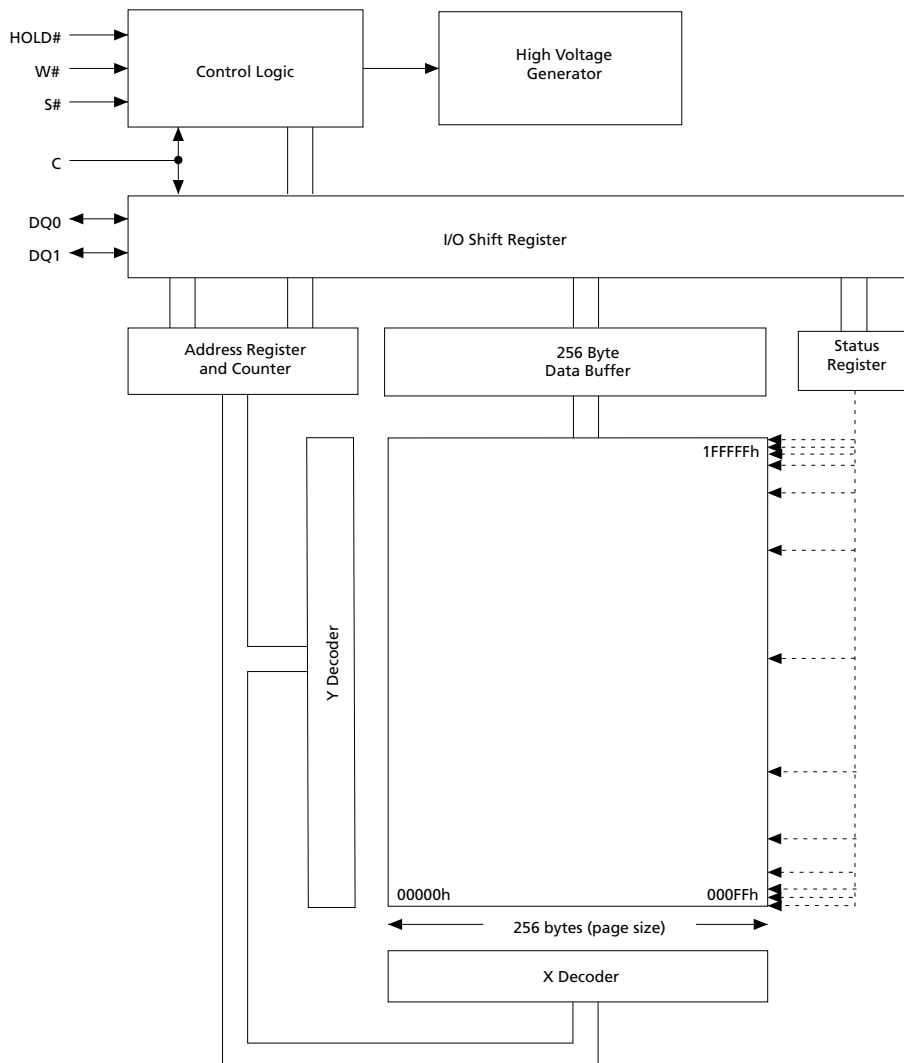
## Configuration and Memory Map

### Memory Configuration and Block Diagram

Each page of memory can be individually programmed; bits are programmed from 1 to 0. The device is sector or bulk-erasable, but not page-erasable; bits are erased from 0 to 1. The memory is configured as follows:

- 2,097,152 bytes (8 bits each)
- 32 sectors (512Kb, 65KB each)
- 8,192 pages (256 bytes each)

**Figure 7: Block Diagram**





## Memory Map – 16Mb Density

Table 4: Sectors 31:0

Sector	Address Range	
	Start	End
31	001F 0000	001F FFFF
30	001E 0000	001E FFFF
29	001D 0000	001D FFFF
28	001C 0000	001C FFFF
27	001B 0000	001B FFFF
26	001A 0000	001A FFFF
25	0019 0000	0019 FFFF
24	0018 0000	0018 FFFF
23	0017 0000	0017 FFFF
22	0016 0000	0016 FFFF
21	0015 0000	0015 FFFF
20	0014 0000	0014 FFFF
19	0013 0000	0013 FFFF
18	0012 0000	0012 FFFF
17	0011 0000	0011 FFFF
16	0010 0000	0010 FFFF
15	000F 0000	000F FFFF
14	000E 0000	000E FFFF
13	000D 0000	000D FFFF
12	000C 0000	000C FFFF
11	000B 0000	000B FFFF
10	000A 0000	000A FFFF
9	0009 0000	0009 FFFF
8	0008 0000	0008 FFFF
7	0007 0000	0007 FFFF
6	0006 0000	0006 FFFF
5	0005 0000	0005 FFFF
4	0004 0000	0004 FFFF
3	0003 0000	0003 FFFF
2	0002 0000	0002 FFFF
1	0001 0000	0001 FFFF
0	0000 0000	0000 FFFF



## Command Set Overview

All commands, addresses, and data are shifted in and out of the device, most significant bit first.

Serial data inputs DQ0 and DQ1 are sampled on the first rising edge of serial clock (C) after chip select (S#) is driven LOW. Then, the one-byte command code must be shifted in to the device, most significant bit first, on DQ0 and DQ1, each bit being latched on the rising edges of C.

Every command sequence starts with a one-byte command code. Depending on the command, this command code might be followed by address or data bytes, by address and data bytes, or by neither address or data bytes. For the following commands, the shifted-in command sequence is followed by a data-out sequence. S# can be driven HIGH after any bit of the data-out sequence is being shifted out.

- READ DATA BYTES (READ)
- READ DATA BYTES at HIGHER SPEED
- READ STATUS REGISTER
- READ IDENTIFICATION
- RELEASE from DEEP POWER-DOWN

For the following commands, S# must be driven HIGH exactly at a byte boundary. That is, after an exact multiple of eight clock pulses following S# being driven LOW, S# must be driven HIGH. Otherwise, the command is rejected and not executed.

- PAGE PROGRAM
- SECTOR ERASE
- BULK ERASE
- WRITE STATUS REGISTER
- WRITE ENABLE
- WRITE DISABLE

All attempts to access the memory array are ignored during a WRITE STATUS REGISTER command cycle, a PROGRAM command cycle, or an ERASE command cycle. In addition, the internal cycle for each of these commands continues unaffected.



**Table 5: Command Set Codes**

Command Name	One-Byte Command Code		Bytes		
			Address	Dummy	Data
WRITE ENABLE	0000 0110	06h	0	0	0
WRITE DISABLE	0000 0100	04h	0	0	0
READ IDENTIFICATION	1001 1111	9Fh	0	0	1 to 20
	1001 1110	9Eh			
READ STATUS REGISTER	0000 0101	05h	0	0	1 to ∞
WRITE STATUS REGISTER	0000 0001	01h	0	0	1
READ DATA BYTES	0000 0011	03h	3	0	1 to ∞
READ DATA BYTES at HIGHER SPEED	0000 1011	0Bh	3	1	1 to ∞
PAGE PROGRAM	0000 0010	02h	3	0	1 to 256
SECTOR ERASE	1101 1000	D8h	3	0	0
BULK ERASE	1100 0111	C7h	0	0	0
DEEP POWER-DOWN	1011 1001	B9h	0	0	0
RELEASE from DEEP POWER-DOWN	1010 1011	ABh	0	0	1 to ∞

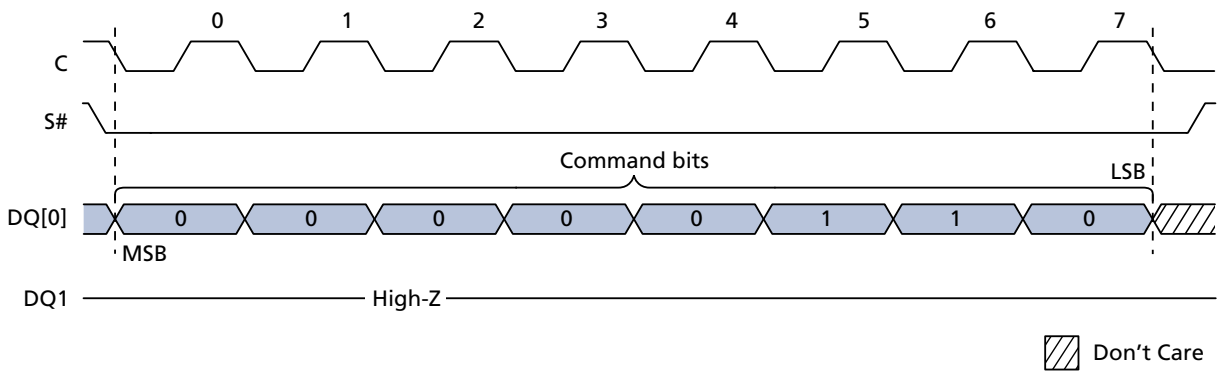
## WRITE ENABLE

The WRITE ENABLE command sets the write enable latch (WEL) bit.

The WEL bit must be set before execution of every PROGRAM, ERASE, and WRITE command.

The WRITE ENABLE command is entered by driving chip select (S#) LOW, sending the command code, and then driving S# HIGH.

**Figure 8: WRITE ENABLE Command Sequence**



## WRITE DISABLE

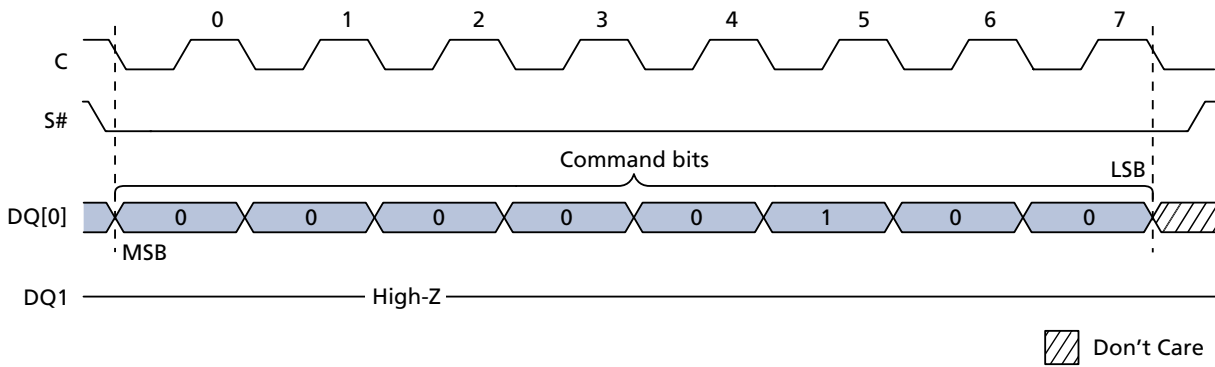
The WRITE DISABLE command resets the write enable latch (WEL) bit.

The WRITE DISABLE command is entered by driving chip select (S#) LOW, sending the command code, and then driving S# HIGH.

The WEL bit is reset under the following conditions:

- Power-up
- Completion of any ERASE operation
- Completion of any PROGRAM operation
- Completion of any WRITE REGISTER operation
- Completion of WRITE DISABLE operation

**Figure 9: WRITE DISABLE Command Sequence**



## READ IDENTIFICATION

The READ IDENTIFICATION command reads the following device identification data:

- Manufacturer identification (1 byte): This is assigned by JEDEC.
- Device identification (2 bytes): This is assigned by device manufacturer; the first byte indicates memory type and the second byte indicates device memory capacity.
- A Unique ID code (UID) (17 bytes, 16 available upon customer request): The first byte contains length of data to follow; the remaining 16 bytes contain optional Customized Factory Data (CFD) content.

**Table 6: READ IDENTIFICATION Data Out Sequence**

Manufacturer Identification	Device Identification		UID	
	Memory Type	Memory Capacity	CFD Length	CFD Content
20h	20h	15h	10h	16 bytes

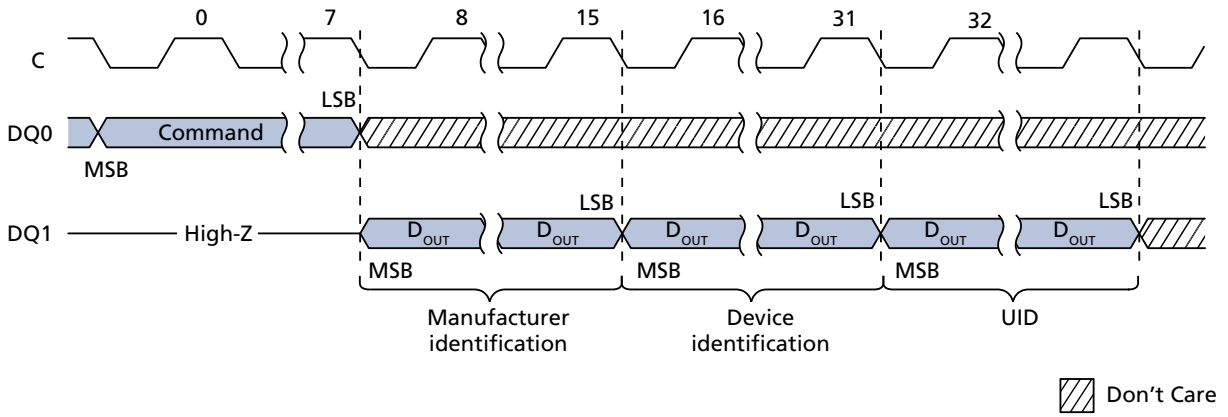
Note: 1. The CFD bytes are read-only and can be programmed with customer data on demand. If customers do not make requests, the devices are shipped with all the CFD bytes programmed to zero. Parts with a top marking year and week code greater than "06/2015" and assembled in the UDFPN8 4mm x 3mm CFD package area are programmed with an UID. To decode top marking year and week codes (week of assembly), see TN-12-24 "Serial Flash Memory Device Marking for the M25P/PE/PX, M45PE, and N25Q Product Families" on [micron.com](http://micron.com).

A READ IDENTIFICATION command is not decoded while an ERASE or PROGRAM cycle is in progress and has no effect on a cycle in progress. The READ IDENTIFICATION command must not be issued while the device is in DEEP POWER-DOWN mode.

The device is first selected by driving S# LOW. Then the 8-bit command code is shifted in and content is shifted out on DQ1 as follows: the 24-bit device identification that is stored in the memory, the 8-bit CFD length, followed by 16 bytes of CFD content. Each bit is shifted out during the falling edge of serial clock (C).

The READ IDENTIFICATION command is terminated by driving S# HIGH at any time during data output. When S# is driven HIGH, the device is put in the STANDBY POWER mode and waits to be selected so that it can receive, decode, and execute commands.

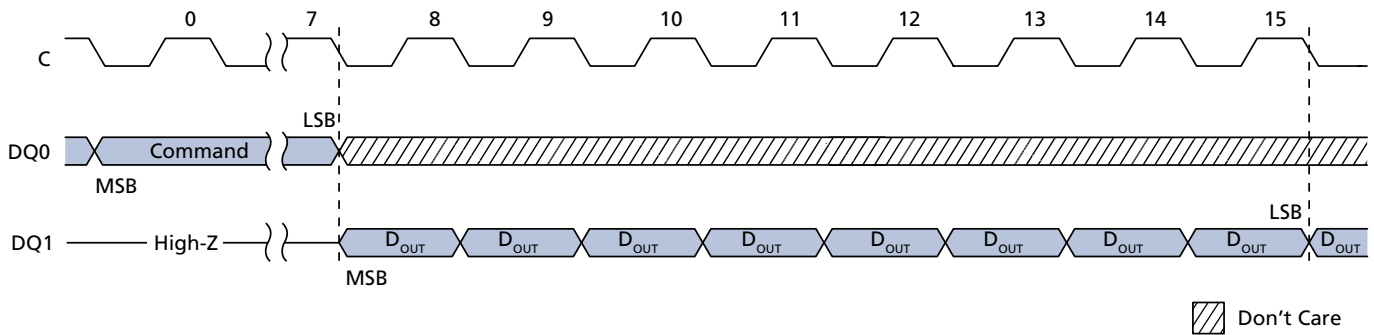
**Figure 10: READ IDENTIFICATION Command Sequence**



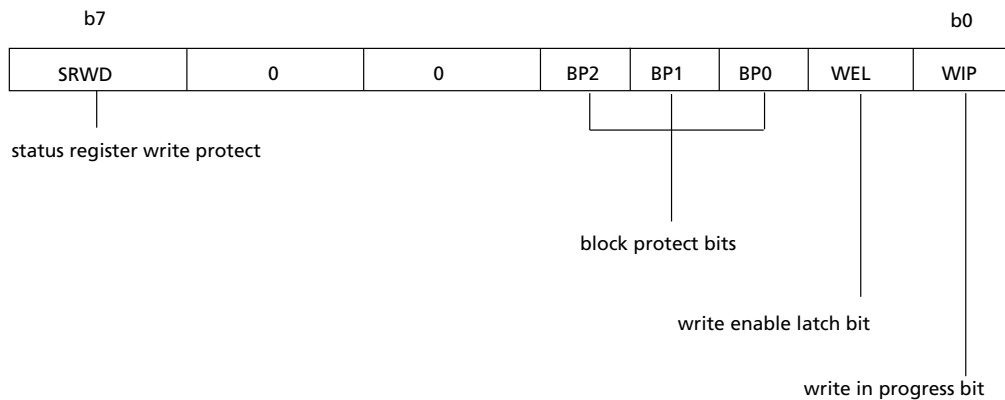
## READ STATUS REGISTER

The READ STATUS REGISTER command allows the status register to be read. The status register may be read at any time, even while a PROGRAM, ERASE, or WRITE STATUS REGISTER cycle is in progress. When one of these cycles is in progress, it is recommended to check the write in progress (WIP) bit before sending a new command to the device. It is also possible to read the status register continuously.

**Figure 11: READ STATUS REGISTER Command Sequence**



**Figure 12: Status Register Format**



### WIP Bit

The write in progress (WIP) bit indicates whether the memory is busy with a WRITE STATUS REGISTER cycle, a PROGRAM cycle, or an ERASE cycle. When the WIP bit is set to 1, a cycle is in progress; when the WIP bit is set to 0, a cycle is not in progress.

### WEL Bit

The write enable latch (WEL) bit indicates the status of the internal write enable latch. When the WEL bit is set to 1, the internal write enable latch is set; when the WEL bit is set to 0, the internal write enable latch is reset and no WRITE STATUS REGISTER, PROGRAM, or ERASE command is accepted.

### Block Protect Bits

The block protect bits are non-volatile. They define the size of the area to be software protected against PROGRAM and ERASE commands. The block protect bits are written with the WRITE STATUS REGISTER command.

When one or more of the block protect bits is set to 1, the relevant memory area, as defined in the Protected Area Sizes table, becomes protected against PAGE PROGRAM and SECTOR ERASE commands. The block protect bits can be written provided that the HARDWARE PROTECTED mode has not been set. The BULK ERASE command is executed only if all block protect bits are 0.

### SRWD Bit

The status register write disable (SRWD) bit is operated in conjunction with the write protect ( $W\#$ ) signal. When the SRWD bit is set to 1 and  $W\#$  is driven LOW, the device is put in the hardware protected mode. In the hardware protected mode, the non-volatile bits of the status register (SRWD, and the block protect bits) become read-only bits and the WRITE STATUS REGISTER command is no longer accepted for execution.

### SRWD Bit

The status register write disable (SRWD) bit is operated in conjunction with the write protect ( $W\#/V_{pp}$ ) signal. When the SRWD bit is set to 1 and  $W\#/V_{pp}$  is driven LOW, the device is put in the hardware protected mode. In the hardware protected mode, the non-volatile bits of the status register (SRWD, and the block protect bits) become read-only bits and the WRITE STATUS REGISTER command is no longer accepted for execution.

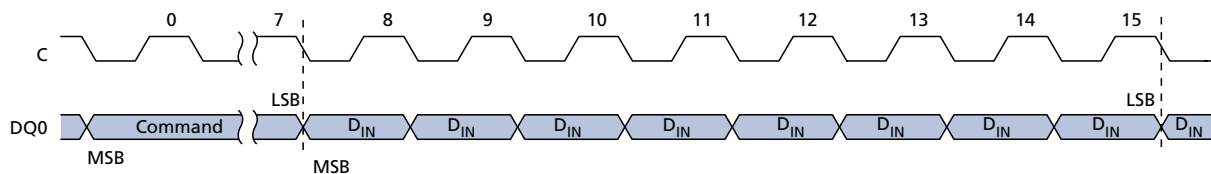


## WRITE STATUS REGISTER

The WRITE STATUS REGISTER command allows new values to be written to the status register. Before the WRITE STATUS REGISTER command can be accepted, a WRITE ENABLE command must have been executed previously. After the WRITE ENABLE command has been decoded and executed, the device sets the write enable latch (WEL) bit.

The WRITE STATUS REGISTER command is entered by driving chip select (S#) LOW, followed by the command code and the data byte on serial data input (DQ0). The WRITE STATUS REGISTER command has no effect on b6, b5, b4, b1, and b0 of the status register. The status register b6 b5, and b4 are always read as '0'. S# must be driven HIGH after the eighth bit of the data byte has been latched in. If not, the WRITE STATUS REGISTER command is not executed.

**Figure 13: WRITE STATUS REGISTER Command Sequence**



As soon as S# is driven HIGH, the self-timed WRITE STATUS REGISTER cycle is initiated; its duration is  $t_W$ . While the WRITE STATUS REGISTER cycle is in progress, the status register may still be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed WRITE STATUS REGISTER cycle, and is 0 when the cycle is completed. Also, when the cycle is completed, the WEL bit is reset.

The WRITE STATUS REGISTER command allows the user to change the values of the block protect bits (BP2, BP1, BP0). Setting these bit values defines the size of the area that is to be treated as read-only, as defined in the Protected Area Sizes table.

The WRITE STATUS REGISTER command also allows the user to set and reset the status register write disable (SRWD) bit in accordance with the write protect (W#/V<sub>PP</sub>) signal. The SRWD bit and the W#/V<sub>PP</sub> signal allow the device to be put in the HARDWARE PROTECTED (HPM) mode. The WRITE STATUS REGISTER command is not executed once the HPM is entered. The options for enabling the status register protection modes are summarized here.

**Table 7: Status Register Protection Modes**

W#/V <sub>PP</sub> Signal	SRWD Bit	Protection Mode (PM)	Status Register Write Protection	Memory Content		Notes
				Protected Area	Unprotected Area	
1	0	SOFTWARE PROTECTED mode (SPM)	Software protection	Commands not accepted	Commands accepted	1, 2, 3
0	0					
1	1					
0	1	HARDWARE PROTECTED mode (HPM)	Hardware protection	Commands not accepted	Commands accepted	3, 4, 5,

- Notes:
1. Software protection: status register is writable (SRWD, BP2, BP1, and BP0 bit values can be changed) if the WRITE ENABLE command has set the WEL bit.
  2. PAGE PROGRAM, SECTOR ERASE, AND BULK ERASE commands are not accepted.
  3. PAGE PROGRAM and SECTOR ERASE commands can be accepted.
  4. Hardware protection: status register is not writable (SRWD, BP2, BP1, and BP0 bit values cannot be changed).
  5. PAGE PROGRAM, SECTOR ERASE, AND BULK ERASE commands are not accepted.

When the SRWD bit of the status register is 0 (its initial delivery state), it is possible to write to the status register provided that the WEL bit has been set previously by a WRITE ENABLE command, regardless of whether the W#/V<sub>PP</sub> signal is driven HIGH or LOW. When the status register SRWD bit is set to 1, two cases need to be considered depending on the state of the W#/V<sub>PP</sub> signal:

- If the W#/V<sub>PP</sub> signal is driven HIGH, it is possible to write to the status register provided that the WEL bit has been set previously by a WRITE ENABLE command.
- If the W#/V<sub>PP</sub> signal is driven LOW, it is not possible to write to the status register even if the WEL bit has been set previously by a WRITE ENABLE command. Therefore, attempts to write to the status register are rejected, and are not accepted for execution. The result is that all the data bytes in the memory area that have been put in SPM by the status register block protect bits (BP2, BP1, BP0) are also hardware protected against data modification.

Regardless of the order of the two events, the HPM can be entered in either of the following ways:

- Setting the status register SRWD bit after driving the W#/V<sub>PP</sub> signal LOW
- Driving the W#/V<sub>PP</sub> signal LOW after setting the status register SRWD bit.

The only way to exit the HPM is to pull the W#/V<sub>PP</sub> signal HIGH. If the W#/V<sub>PP</sub> signal is permanently tied HIGH, the HPM can never be activated. In this case, only the SPM is available, using the status register block protect bits (BP2, BP1, BP0).