Numonyx® Omneo™ P8P PCM

128-Mbit Parallel Phase Change Memory

Datasheet

Product Features

- High Performance Read
  - 115 ns initial read access
  - 135 ns initial read access
  - 25 ns 8-word asynchronous-page read
- Architecture
  - Asymmetrically-blocked architecture
  - Four 32-KByte parameter blocks: top or bottom configuration
  - 128-KByte main blocks
  - Serial Peripheral Interface (SPI) to enable lower pin count on-board programming
- Phase Change Memory (PCM)
  - Chalcogenide phase change storage element
  - Bit alterable write operation
- Voltage and Power
  - V_{CC} (core) voltage: 2.7 V – 3.6 V
  - V_{CCQ} (I/O) voltage: 1.7 V – 3.6 V
  - Standby current: 80 µA (Typ)
- Quality and Reliability
  - More than 1,000,000 write cycles
  - 90 nm PCM technology
- Temperature
  - Operating temperature -30 °C to +85 °C (135ns initial read access)
  - Operating temperature 0 °C to +70 °C (115ns initial read access)
- Security
  - One-Time Programmable Registers:
    - 64 unique factory device identifier bits
    - 2112 user-programmable OTP bits
  - Selectable OTP Space in Main Array:
    - Four pre-defined 32-KByte blocks (top or bottom configuration)
    - Three adjacent Main Blocks available for boot code or other secure information
  - Absolute write protection: $V_{PP} = V_{SS}$
  - Power-transition erase/program lockout
  - Individual zero-latency block locking
  - Individual block lock-down
- Simplified Software Management
  - No block erase or cleanup required
  - Bit “twiddle” in either direction (1:0, 0:1)
  - 35 µs (Typ) program suspend
  - 35 µs (Typ) erase suspend
  - Numonyx® Flash Data Integrator optimized
  - Scalable Command Set and Extended Command Set compatible
  - Common Flash Interface capable
- Density and Packaging
  - 128 Mbit density
  - 56-Lead TSOP package
  - 64-Ball Numonyx® Easy BGA package

316144-07
August 2010

Revision History

Date	Revision	Description
December 14th, 2006	0	Initial Advance Information Datasheet
March, 2007	1	Advance Information Datasheet
July, 2007	2	Fixed the spelling error and deleted a repeated sentence on page 10 Added Section 2.3 “64-Ball EBGA Package” on page 12 Added Figure 1 “EBGA Mechanical Specifications” on page 12 Added Table 1 “EBGA Package Dimensions” on page 12 Added note 6 on page 67 Updated note 5 on page 68 Fixed an error on the A33 Device Code on “from 881E8 Hex, to 881E Hex”
April 2008	03	Applied Numonyx branding.
February 2009	04	Changed the Operating Temperature on the Title page as well as Table 19 Changed the Writing Endurance to 100,000 No Read while at Streaming Mode in Section 4.4 Changed the stand by current to 160usec in Section 7.2 Added note 5 in Table 19 footnotes Changed the read latency to 115nsec. Also, changed the values of R1 and R2 to 115nsec in Section 7.4
July 2009	05	Removed Numonyx Confidential Removed Streaming Mode references Changed all A33 references to P8P Revised Easy BGA Package Dimensions (Table 4) Revised SPI Section (Ch-12) Changes Erase & Program Suspend Specification Changed P2 Specification Changed W250 non-streaming mode legacy programming Applied Numonyx DS formatting
April 2010	06	Added Numonyx® Omneo™ Branding Added Program on all 1’s command (D1h) to Table-20 Added Edurance table to operating conditions section Updated AC/DC Specifications: P3 (max), ICCS (typ), ICCS/ICCD/ICCES/ICCWS (typ), ICCR (typ/max), Capacitance (max), tCLQV (max), tHHQX (max), Buffer Program (typ/max), Block Erase (typ/max), Suspend Latency (max)
August 2010	07	Added -30 to +85C (Cover Page, Section 14) Removed Storage Temp Range (Section 14) Revised AC Read Spec for -30 to +85C (Section 16.3) Revised SPI AC Spec for -30 to +85C (Section 16.5) Revised Ordering Information (Section 18)

1.0 Product Description

1.1 Introduction

Numonyx® Omneo™ Phase Change Memory for embedded applications offers all of the best attributes from other memory types in a new, highly scalable and flexible technology.

Phase Change Memory (PCM) is a new type of nonvolatile semiconductor memory that stores information through a reversible structural phase change in a chalcogenide material. The material exhibits a change in material properties, both electrical and optical, when changed from the amorphous (disordered) to the polycrystalline (regularly ordered) state. In the case of Phase Change Memory, information is stored via the change in resistance the chalcogenide material experiences upon undergoing a phase change. The material also changes optical properties after experiencing a phase change, a characteristic that has been successfully mastered for use in current rewritable optical storage devices such as rewritable CDs and DVDs.

The PCM storage element consists of a thin film of chalcogenide contacted by a resistive heating element. In PCM, the phase change is induced in the memory cell by highly localized Joule heating caused by an induced current at the material junction. During a write operation, a small volume of the chalcogenide material is made to change phase. The phase change is a reversible process, and is modulated by the magnitude of injected current, the applied voltage, and the duration of the heating pulse.

PCM combines the benefits of traditional floating gate flash, both NOR-type and NAND-type, with some of the key attributes of RAM and EEpROM. Like NOR flash and RAM technology, PCM offers fast random access times. Like NAND flash, PCM has the ability to write moderately fast. And like RAM and EEpROM, PCM supports bit alterable writes (overwrite). Unlike flash, no separate erase step is required to change information from 0 to 1 and 1 to 0. Unlike RAM, however, the technology is nonvolatile with data retention comparable NOR flash. However, at the current time, PCM technology appears to have a write cycling endurance better than that of NAND or NOR flash, but less than that of RAM.

Unlike other proposed alternative memories, PCM technology uses a conventional CMOS process with the addition of a few additional layers to form the memory storage element. Overall, the basic memory manufacturing process used to make PCM is less complex than that of NAND, NOR or DRAM.

Historically, systems have adopted many different types of memory to meet different needs within a design. Some systems might include boot memory, configuration memory, data storage memory, high speed execution memory, and dynamic working memory. The demands of many of today's designs require better performance from the memory subsystem and a reduction in the overall component count. PCM provides many of the attributes of different kinds of memory found in a typical design, enabling the opportunity to consolidate or eliminate of different types of memory.

The combination of fast random access with high speed, bit alterable writes in a nonvolatile memory is a capability only offered in complex, low density technologies such as parallel EEpROM or battery-backed RAM. The PCM feature set is intended to facilitate easy evaluation and adoption in systems and to enable the consolidation of memory functions into a single device. In some cases, PCM may enable new usages or new solutions to existing problems, in a manner that is more efficient, higher performance and/or more cost effective.

1.2 Product Overview

The Numonyx® Omneo™ P8P PCM provides the convenience and ease of NOR flash emulation while providing a set of Super Set features that exploit the inherent capabilities of the PCM technology. The device emulates most of the features of the Numonyx® Axcell™ Embedded Memory (P33). This is intended to ease the evaluation and design of Numonyx® Omneo™ P8P PCM into existing hardware and software development platforms. This basic features set is supplemented by the Super Set Features. The Super Set Features are intended to allow the designer to exploit the inherent capabilities of the phase change memory technology, and to enable the eventual simplification of hardware and software in the design. This section describes an overview of the features and capabilities of Numonyx® Omneo™ P8P PCM.

- **Density:** Numonyx® Omneo™ P8P PCM product family begins with a 128-Mbit density.
- **Packages:** Numonyx® Omneo™ P8P PCM devices are available in 64 Ball Easy BGA and 56 Lead TSOP packages. These are the same pinouts and packages as the existing P33 NOR flash devices.
- **Low Power:** Designed for low voltage systems, Numonyx® Omneo™ P8P PCM supports read, write and erase operations at a core supply of 2.7V V_{CC}. P8P offers additional power savings through standby mode. Standby mode is initiated when the system deselects the device by driving CE inactive, which significantly reduces power consumption.
- **NOR-Compatible Program and Emulated Erase Operation:** Numonyx® Omneo™ P8P PCM provides a complete set of commands that are compatible with industry-standard command sequences used by NOR-type flash. An internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for block erase and write. Each emulated block erase operation results in the contents of the addressed block being written to all “1s” (ones). Data can be programmed in word or buffer increments. Erase-suspend allows system software to pause an erase command so it can read or program data in another block. Program suspend allows system software to pause programming so it can read from other locations within the device. The Status Register indicates when the WSM’s block erase, or program operation is finished.
- **Write Buffer:** A 64 byte/32 word Write Buffer is also included to allow optimum write performance. By using the write buffer, data is overwritten or programmed in buffer increments. This feature improves system program performance more than 20 times over independent byte writes.
- **Command User Interface**: As with floating gate flash, a Command User Interface (CUI) serves as the interface between the system processor and internal operation of the device. A valid command sequence written to the CUI initiates device automation.
- **Data Protection:** Numonyx® Omneo™ P8P PCM block locking enables zero-latency block locking/unlocking and permanent locking. Permanent block locking provides enhanced security for boot code. The combination of these two locking features provides complete locking solution for code and data.
- **CFI Compliant**: A flash-compatible Common Flash Interface (CFI) permits software algorithms to be used for entire families of devices. This allows device-independent, JEDEC ID-independent, and forward- and backward-compatible software support for the specified flash device families.

- **Bit Alterability or Overwrite:** PCM technology supports the ability to change each memory bit independently from 0 to 1 or 1 to 0 without an intervening block erase operation. Bit Alterability enables software to write to the non-volatile memory in a similar manner as writing to RAM or EEpROM without the overhead of erasing blocks prior to write. Bit Alterable writes use similar command sequences as word programming and Buffer Programming.
- **Serial Peripheral Interface (SPI):** SPI allows for in-system programming through a minimal pin count interface. This interface is provided in addition to a traditional parallel system interface. This feature has been added to facilitate the on-board, in-system programming of code into the Numonyx® Omneo™ P8P PCM device, after it has been soldered to a circuit board. Pre-programming of code prior to high temperature board attach is not recommended with the P8P device. Although the device reliability across the operating temperature range is typically superior to that of floating gate flash, the P8P device may be subject to thermally-activated disturbs at higher temperatures. However, no permanent device damage occurs during either leaded and lead-free board attach.

1.3 Memory Map

This section covers the memory map for the Top and Bottom boot devices

Table 1: Top Parameter Memory Map

Programming Region Number	Size (KW)	Blk	128-Mbit
7	16	130	7FC000-7FFFFF
	16	129	7F8000-7FBFFF
	16	128	7F4000-7F7FFF
	16	127	7F0000-7F3FFF
	64	126	7E0000-7EFFFF
	⋮	⋮	⋮
	64	112	700000-70FFFF
6	64	111	6F0000-6FFFFF
	⋮	⋮	⋮
	64	96	600000-60FFFF
5	64	95	5F0000-5FFFFF
	⋮	⋮	⋮
	64	80	500000-50FFFF
4	64	79	4F0000-4FFFFF
	⋮	⋮	⋮
	64	64	400000-40FFFF
3	64	63	3F0000-3FFFFF
	⋮	⋮	⋮
	64	48	300000-30FFFF
2	64	47	2F0000-2FFFFF
	⋮	⋮	⋮
	64	32	200000-20FFFF
1	64	31	1F0000-1FFFFF
	⋮	⋮	⋮
	64	16	100000-10FFFF
0	64	15	0F0000-0FFFFF
	⋮	⋮	⋮
	64	0	000000-00FFFF

Table 2: Bottom Parameter Memory Map

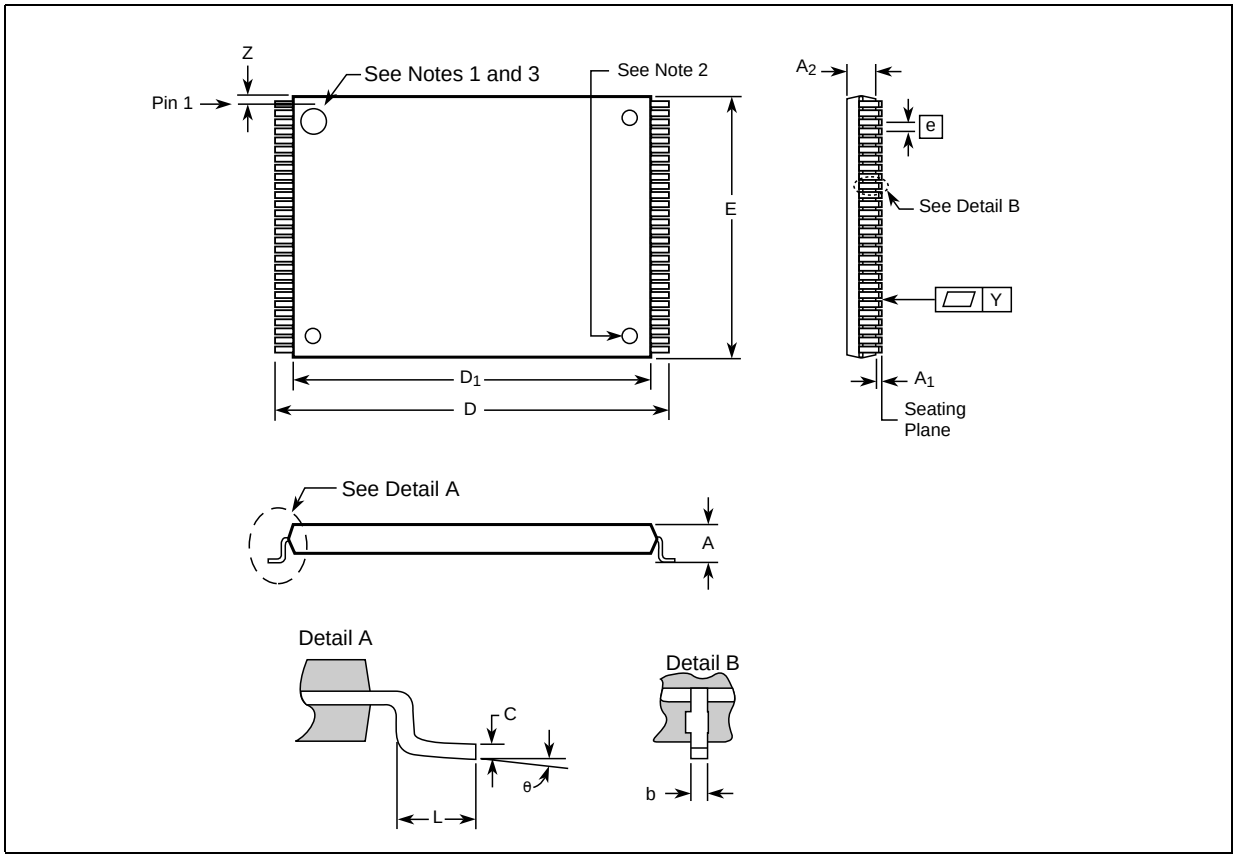
Programming Region Number	Size (KW)	Blk	128-Mbit
7	64	130	7F0000-7FFFFF
	⋮	⋮	⋮
	64	115	700000-70FFFF
6	64	114	6F0000-6FFFFF
	⋮	⋮	⋮
	64	99	600000-60FFFF
5	64	98	5F0000-5FFFFF
	⋮	⋮	⋮
	64	83	500000-50FFFF
4	64	82	4F0000-4FFFFF
	⋮	⋮	⋮
	64	67	400000-40FFFF
3	64	66	3F0000-3FFFFF
	⋮	⋮	⋮
	64	51	300000-30FFFF
2	64	50	2F0000-2FFFFF
	⋮	⋮	⋮
	64	35	200000-20FFFF
1	64	34	1F0000-1FFFFF
	⋮	⋮	⋮
	64	19	100000-10FFFF
0	64	18	0F0000-0FFFFF
	⋮	⋮	⋮
	64	4	010000-01FFFF
	16	3	00C000-00FFFF
	16	2	008000-00BFFF
	16	1	004000-007FFF
	16	0	000000-003FFF

2.0 Package Information

This section covers the mechanical specifications for the available packages.

2.1 56-Lead TSOP

Figure 1: TSOP Mechanical Specifications



Notes:
1. One dimple on package denotes Pin 1.
2. If two dimples, then the larger dimple denotes Pin 1.
3. Pin 1 will always be in the upper left corner of the package, in reference to the product mark.

Table 3: TSOP Package Dimensions (Sheet 1 of 2)

Product Information	Symbol	Millimeters			Inches			Notes
		Min	Nom	Max	Min	Nom	Max	
Package Height	A	-	-	1.200	-	-	0.047	
Standoff	A_1	0.050	-	-	0.002	-	-	
Package Body Thickness	A_2	0.965	0.995	1.025	0.038	0.039	0.040	
Lead Width	b	0.100	0.150	0.200	0.004	0.006	0.008	

Table 3: TSOP Package Dimensions (Sheet 2 of 2)

Product Information	Symbol	Millimeters			Inches			Notes
		Min	Nom	Max	Min	Nom	Max	
Lead Thickness	c	0.100	0.150	0.200	0.004	0.006	0.008	
Package Body Length	D_1	18.200	18.400	18.600	0.717	0.724	0.732	
Package Body Width	E	13.800	14.000	14.200	0.543	0.551	0.559	
Lead Pitch	e	-	0.500	-	-	0.0197	-	
Terminal Dimension	D	19.800	20.00	20.200	0.780	0.787	0.795	
Lead Tip Length	L	0.500	0.600	0.700	0.020	0.024	0.028	
Lead Count	N	-	56	-	-	56	-	
Lead Tip Angle	θ	0°	3°	5°	0°	3°	5°	
Seating Plane Coplanarity	Y	-	-	0.100	-	-	0.004	
Lead to Package Offset	Z	0.150	0.250	0.350	0.006	0.010	0.014	

2.2 64-Ball Easy BGA Package

Figure 2: Easy BGA Mechanical Specifications

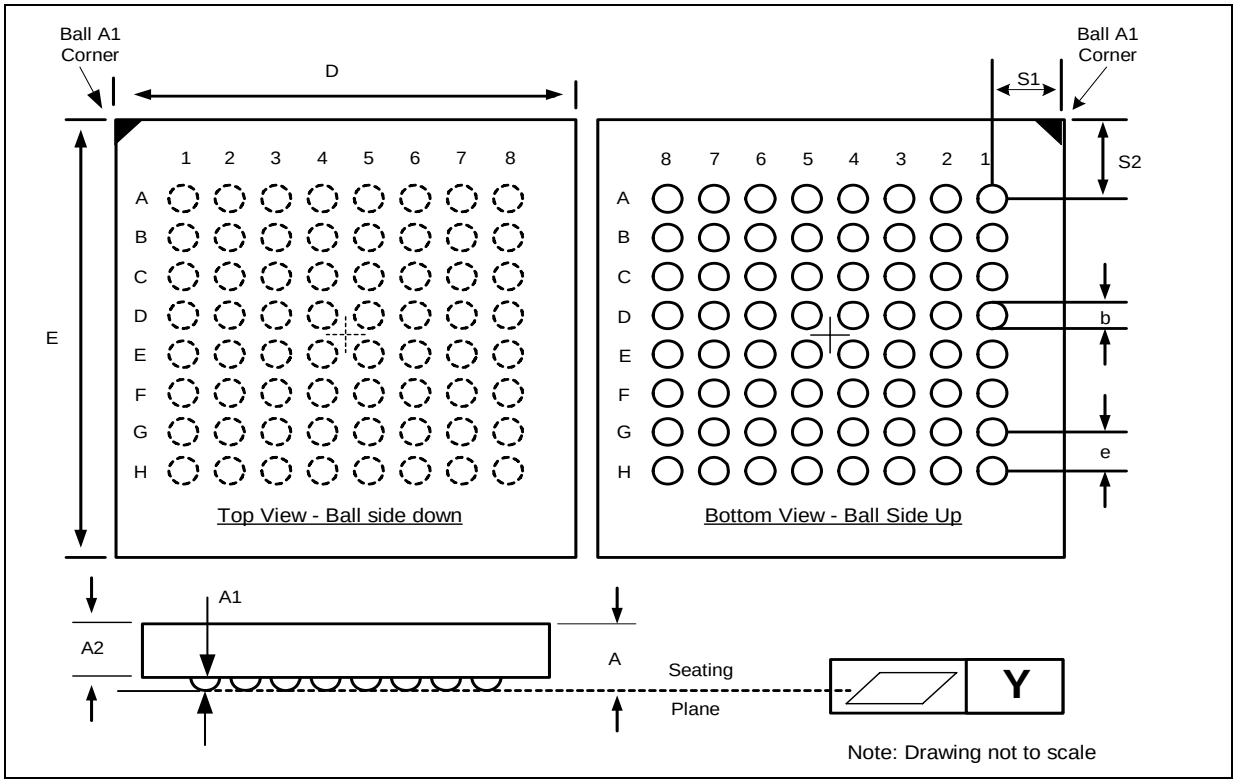
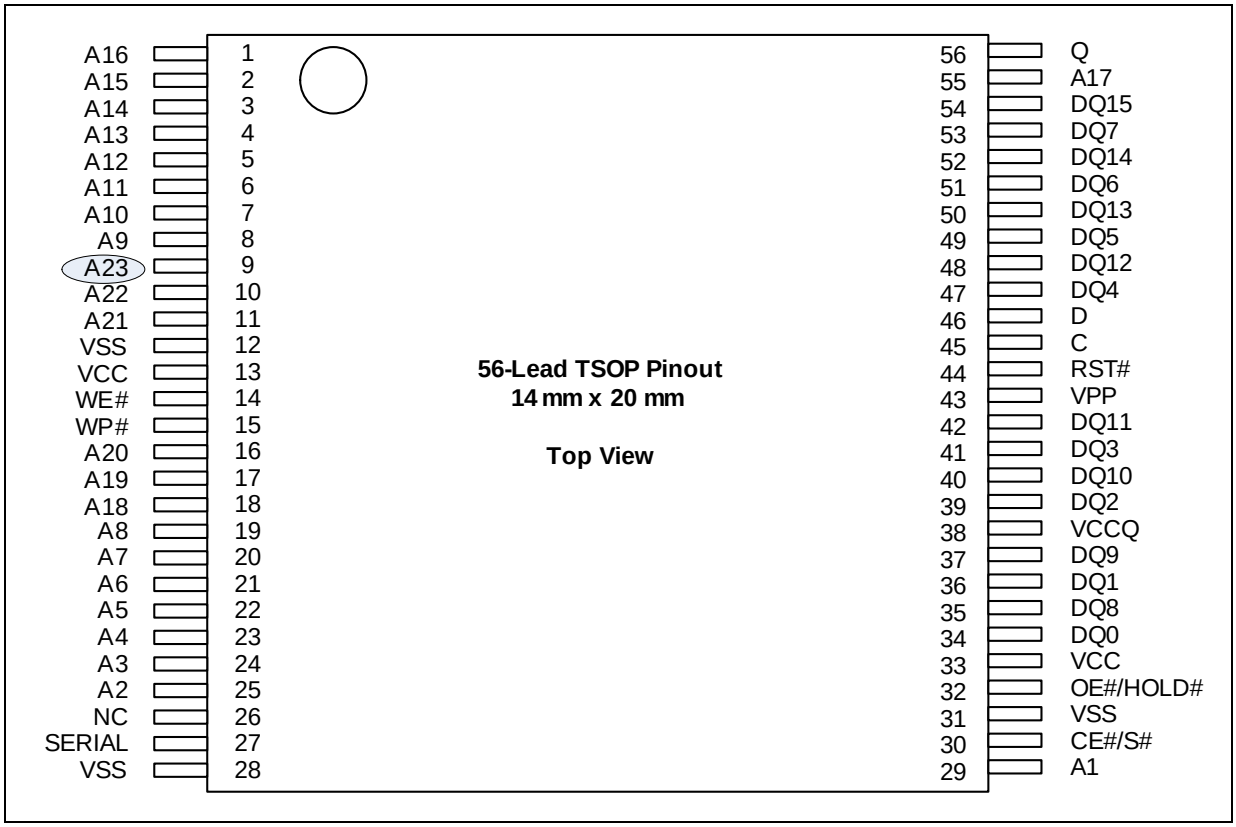


Table 4: Easy BGA Package Dimensions

Product Information	Symbol	Millimeters			Notes
		Min	Nom	Max	
Package Height (128-Mbit)	A	-	-	1.20	
Ball Height	A1	0.25	-	-	
Package Body Thickness (128-Mbit)	A2	-	0.78	-	
Ball (Lead) Width	b	0.33	0.43	0.53	
Package Body Width	D	9.90	10.00	10.10	
Package Body Length	E	7.90	8.00	8.10	
Pitch	e	-	1.00	-	
Ball (Lead) Count	N	-	64	-	
Seating Plane Coplanarity	Y	-	-	0.10	
Corner to Ball A1 Distance Along D	S1	1.40	1.50	1.60	
Corner to Ball A1 Distance Along E	S2	0.49	0.50	0.51	

3.0 Pinouts and Ballouts

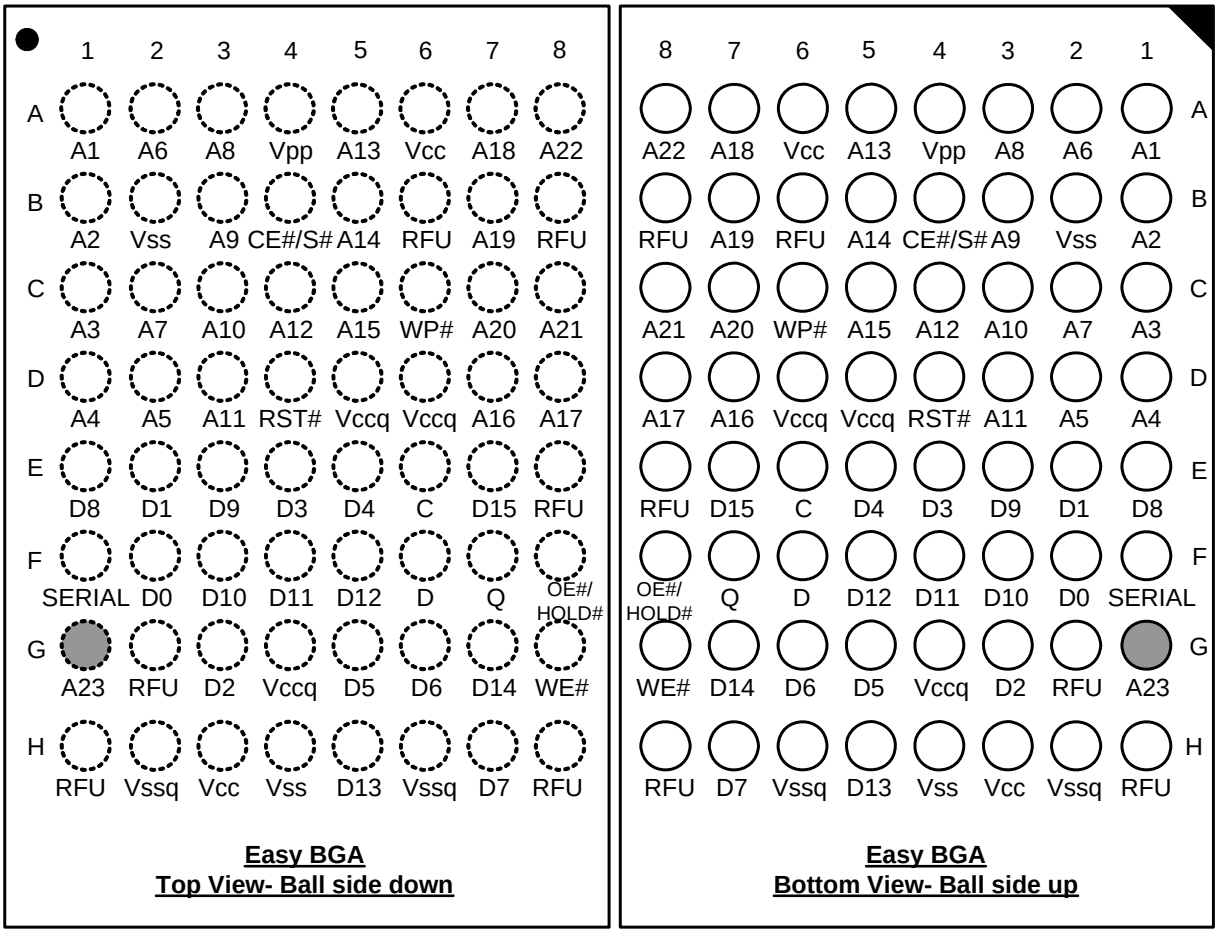
Figure 3: 56-Lead TSOP Pinout (128-Mbit)



Notes:

1. A1 is the least significant address bit to be compatible with x8 addressing systems. Even though Numonyx® Omneo™ P8P PCM is a 16 bit data bus.

Figure 4: 64-Ball Easy BGA Ballout (128-Mbit)



Notes:

1. A1 is the least significant address bit to be compatible with x8 addressing systems, even though Numonyx® Omneo™ P8P PCM is a 16 bit data bus.

4.0 Signals

Table 5: Ball/Pin Descriptions

Symbol	Type	Name and Function
A[MAX:1]	Input	**ADDRESS INPUTS**: Device address inputs. 128-Mbit: A[23:1] ***Note:*** The address bus for TSOP and Easy BGA starts at A1. Numonyx® Omneo™ P8P PCM uses x16 addressing. The Numonyx® Omneo™ P8P PCM package is x8 addressing and is compatible with J3 or P30 products.
DQ[15:0]	Input/ Output	**DATA INPUT/OUTPUTS:** Inputs data and commands during writes (internally latched). Outputs data during read operations. Data signals float when CE# or OE# are V_{IH}. or RST# is V_{IL}.
CE# or S#	Input	**CHIP ENABLE:** CE#-low activates internal control logic, I/O buffers, decoders, and sense amps. CE#-high deselects the device, places it in standby state, and places data outputs at high-Z.
	SPI	**SPI Select:** S# low activates command writes to the SPI interface. Raising S# to VIH completes (or terminates) the SPI command cycle; it also sets Q to high-Z.
OE# or HOLD#	Input	**OUTPUT ENABLE:** Active low OE# enables the device's output data buffers during a read cycle. With OE# at V_{IH}, device data outputs are placed in high-Z state.
	SPI	**SPI HOLD#:** When asserted, suspends the current cycle and sets Q to high-Z until de-asserted.
RST#	Input	**RESET CHIP:** When low, RST# resets internal automation and inhibits write operations. This provides data protection during power transitions. RST#-high enables normal operation. The device is in 8-Word page mode array read after reset exits.
WE#	Input	**WRITE ENABLE:** controls Command User Interface (CUI) and array writes. Its rising edge latches addresses and data.
WP#	Input	**WRITE PROTECT:** Disables/enables the lock-down function. **When WP# is V_{IL}, the lock-down mechanism is enabled** and software cannot unlock blocks marked lock-down. **When WP# is V_{IH}, the lock-down mechanism is disabled** and blocks previously locked-down are now locked; software can unlock and lock them. After WP# goes low, blocks previously marked lock-down revert to that state.
C	SPI	**SPI Clock:** Synchronization clock for input and output data
D	SPI	**SPI Data Input:** Serial data input for Op Codes, address and program data bytes. Input data is clocked in on the rising edge of C, starting with the MSB.
Q	SPI	**SPI Data Output:** Serial data output for read data. Output data is clocked out, triggered by the falling edge of C, starting with the MSB.
SERIAL	SPI	**SPI Enable:** SERIAL is a port select switching between the normal parallel or serial interface. When Vss, the normal (non-SPI) Numonyx® Omneo™ P8P PCM interface is enabled; all other SPI inputs are Don't Care, and Q is at High-Z. When Vcc, SPI mode is enabled, all non-SPI inputs are Don't Care, and all outputs are at High-Z. This pin has an internal weak pull down resistor to select the normal parallel interface when users leave the pin floating. A CAM can be used to permanently disable this feature.
V_{PP}	Pwr	**ERASE AND WRITE POWER:** A valid V_{PP} voltage allows erase or programming. Memory contents can't be altered when $V_{PP} \le V_{PPLK}$. Set $V_{PP} = V_{CC}$ for in-system program and erase operations. To accommodate resistor or diode drops from the system supply, V_{PP}'s V_{IH} level can be as low as V_{PPLMIN}. Program/erase voltage is normally 1.7 V–3.6 V.
V_{CC}	Pwr	**DEVICE POWER SUPPLY:** Writes are inhibited at $V_{CC} \le V_{LKO}$. Device operations at invalid V_{CC} voltages should not be attempted.
V_{CCQ}	Pwr	**OUTPUT POWER SUPPLY:** Enables all outputs to be driven at V_{CCQ}. This input may be tied directly to V_{CC} if V_{CCQ} is to function within the V_{CC} range.
V_{SS}	Pwr	**GROUND:** connects device circuitry to system ground.
V_{SSQ}	Pwr	**I/O GROUND:** Tie to GND
NC		**NO CONNECT:** No internal connection; can be driven or floated.
DU		**DON'T USE:** Don't connect to power supply or other signals.
RFU		RESERVED FOR FUTURE USE: Don't connect to other signals.

5.0 Bus Operations

CE# at V_{IL} and RST# at V_{IH} enables device read operations. Addresses are always assumed to be valid. OE#-low activates the outputs and gates selected data onto the I/O bus. WE#-low enables device write operations. When the VPP voltage ≤ VPPLK (lockout voltage), only read operations are enabled.

Table 6: Bus Operations

State	RST#	CE#	OE#	WE#	DQ[15:0]	Note
Read (Main Array)	V_{IH}	V_{IL}	V_{IL}	V_{IH}	D_{OUT}	
Read (Status, Query, Identifier)	V_{IH}	V_{IL}	V_{IL}	V_{IH}	D_{OUT}	
Output Disable	V_{IH}	V_{IL}	V_{IH}	V_{IH}	High-Z	
Standby	V_{IH}	V_{IH}	X	X	High-Z	2
Reset	V_{IL}	X	X	X	High-Z	2
Write	V_{IH}	V_{IL}	V_{IH}	V_{IL}	D_{IN}	1

Notes:
1. See Table 8, “Command Sequences in x16 Bus Mode” on page 20 for valid D_{IN} during a write operation.
2. X = Don’t care (L or H)
3. OE# and WE# should never be asserted simultaneously. If done so, OE# overrides WE#.

5.1 Reads

To perform a read operation, RST# and WE# must be deasserted while CE# and OE# are asserted. CE# is the device-select control. When asserted, it enables the flash memory device. OE# is the data-output control. When asserted, the addressed flash memory data is driven onto the I/O bus.

5.2 Writes

To perform a write operation, both CE# and WE# are asserted while RST# and OE# are deasserted. During a write operation, address and data are latched on the rising edge of WE# or CE#, whichever occurs first. Table 7, “Command Codes and Descriptions” on page 19 shows the bus cycle sequence for each of the supported device commands, while Table 8, “Command Sequences in x16 Bus Mode” on page 20 describes each command. See Section 16.0, “AC Characteristics” on page 62 for signal-timing details.

Note: Write operations with invalid V_{CC} and/or V_{PP} voltages can produce spurious results and should not be attempted.

5.3 Output Disable

When OE# is deasserted, device outputs DQ[15:0] are disabled and placed in a high-impedance (High-Z) state, WAIT is also placed in High-Z.

5.4 Standby

When CE# is deasserted the device is deselected and placed in standby, substantially reducing power consumption. In standby, the data outputs are placed in High-Z, independent of the level placed on OE#. Standby current, I_{CCS}, is the average current measured over any 5 ms time interval, 5 µs after CE# is deasserted. During standby, average current is measured over the same time interval 5 µs after CE# is deasserted.

When the device is deselected (while CE# is deasserted) during a program or erase operation, it continues to consume active power until the program or erase operation is completed.

5.5 Reset

As with any automated device, it is important to assert RST# when the system is reset. When the system comes out of reset, the system processor attempts to read from the flash memory if it is the system boot device. If a CPU reset occurs with no flash memory reset, improper CPU initialization may occur because the flash memory may be providing status information rather than array data. Flash memory devices from Numonyx allow proper CPU initialization following a system reset through the use of the RST# input. RST# should be controlled by the same low-true reset signal that resets the system CPU.

After initial power-up or reset, the device defaults to asynchronous Read Array mode, and the Status Register is set to 0x80. Asserting RST# de-energizes all internal circuits, and places the output drivers in High-Z. When RST# is asserted, the device shuts down the operation in progress, a process which takes a minimum amount of time to complete. When RST# has been deasserted, the device is reset to asynchronous Read Array state.

Note: If RST# is asserted during a program or erase operation, the operation is terminated and the memory contents at the aborted location (for a program) or block (for an erase) are no longer valid, because the data may have been only partially written or erased.

When returning from a reset (RST# deasserted), a minimum wait is required before the initial read access outputs valid data. Also, a minimum delay is required after a reset before a write cycle can be initiated. After this wake-up interval passes, normal operation is restored. See Section 16.0, “AC Characteristics” on page 62 for details about signal-timing.

6.0 Command Set

6.1 Device Command Codes

The system CPU provides control of all in-system read, write, and erase operations of the device via the system bus. The on-chip Write State Machine (WSM) manages all block-erase and word-program algorithms.

Device commands are written to the Command User Interface (CUI) to control all flash memory device operations. The CUI does not occupy an addressable memory location; it is the mechanism through which the flash device is controlled.

Table 7: Command Codes and Descriptions

Mode	Code	Device Mode	Description
Read	FFh	Read Array	Places device in read array mode so that data signals output array data on DQ[15:0].
	70h	Read Status Register	Places the device in Status Register read mode. Status data is output on DQ[7:0]. The device automatically enters this mode after a program or erase command is issued to it.
	90h	Read ID Code	Puts the device in read identifier mode. Device reads from the addresses output manufacturer/ device codes, block lock status, or protection register data on DQ[15:0].
	98h	Read Query	Puts the device in read query mode. Device reads from the address given outputting the Common Flash Interface information on DQ[7:0]
	50h	Clear Status Register	The WSM can set the Status Register's block lock (SR.1), V_{PP} (SR.3), program (SR.4), and erase (SR.5) status bits to "1" but cannot clear them. Device reset or the Clear Status Register command at any device address clears those bits to "0."
Program	40h	Program Set-Up	This preferred program command's first cycle prepares the CUI for a program operation. The second cycle latches address and data and executes the WSM Program algorithm at this location. Status Register updates occur when CE# or OE# is toggled. A Read Array command is required to read array data after programming.
	10h	Alt Set-up	Equivalent to a Program Set-Up command (40h).
	42h	Bit Alterable Write	The command sequence is the same as Word Program (40h). The difference is the state of the PCM memory cell can change from a 0 to 1 or 1 to 0, unlike a flash memory cell, which can only change from 1 to 0 during programming.
	E8h	Buffered Program	This command loads a variable number of bytes up to the buffer size 32 words onto the program buffer.
	EAh	Bit Alterable Buffered Write	This command sequence is the similar to Buffered Program, but the buffer write command is bit alterable or overwrite operation. The command sequence is the same as E8h.
	DEh	Buffer Program on all 1s	This command is the same as Buffered Program, but user indicates that the pagee is already set to all 1s. The command sequence is the same as E8h
	D0h	Buffered Write Confirm	The confirm command is issued after the data streaming for writing into the buffer is done. This initiates the WSM to carry out the buffered programing algorithm.
Erase	20h	Block Erase Set-Up	Prepares the CUI for Block Erase. The device emulates erasure of the block addressed by the Erase Confirm command by writing all ones. If the next command is not Erase Confirm, the CUI (a) sets Status Register bits SR.4 and SR.5 to "1," (b) places the device in the read Status Register mode, and (c) waits for another command.
	D0h	Erase Confirm	If the first command was Erase Set-Up (20h), the CUI latches address and data then emulates erasure of the block indicated by the Erase confirm cycle address.
Suspend	B0h	Write or Erase Suspend	This command issued at any device address initiates suspension of the currently executing program/erase operation. The Status Register, invoked by a Read Status Register command, indicates successful suspend operation by setting (1) status bits SR.2 (write suspend) or SR.6 (erase suspend) and SR.7. The WSM remains in the Suspend mode regardless of the control signal states, except RST# = V_{IL}.
	D0h	Suspend Resume	This command issued at any device address resumes suspended program or erase operation.

Table 7: Command Codes and Descriptions

Mode	Code	Device Mode	Description
Block Locking	60h	Lock Set-Up	Prepares the CUI for lock configuration. If the next command is not Block-Lock, Unlock, or Lock-Down the CUI sets SR.4 and SR.5 to indicate command sequence error.
	01h	Lock Block	If the previous command was Lock Set-Up (60h), the CUI locks the addressed block.
	D0h	Unlock Block	After a Lock Set-Up (60h) command the CUI latches the address and unlocks the addressed block.
	2Fh	Lock-Down	After a Lock Set-Up (60h) command, the CUI latches the address and locks-down the addressed block.
Protection	C0h	Protection Program Set-Up	Prepares the CUI for a protection register program operation. The second cycle latches address, data, and starts the WSM's protection register program or lock algorithm. Toggling CE# or OE# updates the PCM Status Register data. To read array data after programming issue a Read Array command.

Note: Don't use unassigned (reserved) commands

6.2 Device Command Bus Cycles

Device operations are initiated by writing specific device commands to the Command User Interface (CUI). Several commands are used to modify array data including Word Program and Block Erase commands. Writing either command to the CUI initiates a sequence of internally-timed functions that culminate in the completion of the requested task. However, the operation can be aborted by either asserting RST# or by issuing an appropriate suspend command

Table 8: Command Sequences in x16 Bus Mode

Mode	Command	Bus Cycles	First Bus Cycle			Second Bus Cycle		
			Oper	Addr(1)	Data(2)	Oper	Addr(1)	Data(2)
Read	Read Array/Reset	1	Write	DnA	FFh	-	-	-
	Read Device Identifiers	≥ 2	Write	DnA	90h	Read	DBA+IA	ID
	Read Query	≥ 2	Write	DnA	98h	Read	DBA+QA	QD
	Read Status Register	2	Write	BA	70h	Read	BA	SRD
	Clear Status Register	1	Write	X	50h	-	-	-
Program	Program	2	Write	WA	40h or 10h	Write	WA	WD
	Bit Alterable Program	2	Write	WA	42h	Write	PA	PD
	Buffered Program(3)	> 2	Write	WA	E8h	Write	WA	N-1
	Bit Alterable Buffered Program(3)	>2	Write	WA	EAh	Write	WA	N-1
	Buffered Program on all 1s	>2	Write	WA	DEh	Write	WA	N-1
Erase	Block Erase	2	Write	BA	20h	Write	BA	D0h
Suspend	Program/Erase Suspend	1	Write	X	B0h	-	-	-
	Program/Erase Resume	1	Write	X	D0h	-	-	-

Table 8: Command Sequences in x16 Bus Mode

Mode	Command	Bus Cycles	First Bus Cycle			Second Bus Cycle		
			Oper	Addr(1)	Data(2)	Oper	Addr(1)	Data(2)
Block Lock	Lock Block	2	Write	BA	60h	Write	BA	01h
	Unlock Block	2	Write	BA	60h	Write	BA	D0h
	Lock-down Block	2	Write	BA	60h	Write	BA	2Fh
Protection	Protection Program	2	Write	PA	C0h	Write	PA	PD
	Lock Protection Program	2	Write	LPA	C0h	Write	LPA	FFFDh

Notes:

1. First command cycle address should be the same as the operation's target address.
   X = Any valid address within the device.
   IA = Identification code address.
   BA = Address within the block.
   LPA = Lock Protection Address (from the CFI). P8P LPA is at 0080h.
   PA = 4-word protection address in the user programmable area of device identification plane.
   DnA = Address within the device.
   DBA = Device Base Address. (A[MAX:1]=0h)
   PRA = Program Region
   QA = Query code address.
   WA = Word address of memory location to be written.
2. SRD = Data read from the status register.
   WD = Data to be written at location WA.
   ID = Identifier code data.
   PD =User programmable protection data.
   QD = Query code data on DQ[7:0].
   N = Data count to be loaded into the device to indicate how many words would be written into the buffer. Because the internal registers count from 0, the user writes N-1 to load N words.
3. The second cycle of the Buffered Program command, which is the count being loaded into the buffer is followed by data streaming up to 32 words and then a confirm command is issued which triggers the programming operation. Refer to the Appendix A, "Buffered Program Flowchart".

7.0 Read Operation

Numonyx® Omneo™ P8P PCM has several read modes:

- **Read array mode:** read returns PCM array data from the addressed locations.
- **Read identifier mode:** reads returns manufacturer device identifier data, block lock status, and protection register data.
- **Read query mode:** read returns device CFI (or query) data.
- **Read Status Register mode:** read returns the device Status Register data. A system processor can check the Status Register to determine the device's state or monitor program or erase progress.

7.1 Read Array Command

The Read Array command places (or resets) the device to read array mode. Upon initial device power-up or after reset (RST# transitions from V_{IL} to V_{IH}), the device defaults to read array mode. If an Erase- or Program-Suspend command suspends the WSM, a subsequent Read Array command will place the device in read array mode. The Read Array command functions independently of V_{PP} voltage.

7.2 Read Identifier Command

The read identifier mode is used to access the manufacturer/device identifier, block lock status, and protection register codes. The identifier space occupies the address range supplied by the Read Identifier command (90h) address.

Table 9: Read Identifier Table

Item	Address[1,2]		Data
Manufacturer Code	DBA + 000000h		
			0089h
Device Code	DBA + 000001h		ID (see Table 10)
Block Lock Configuration	BBA + 000002h		Lock
• Block Is Unlocked			DQ_0 = 0
• Block Is Locked			DQ_0 = 1
• Block Is not Locked-Down			DQ_1 = 0
• Block Is Locked-Down			DQ_1 = 1
• Reserved for Future Use[3]			DQ[7:2]
Lock Protection Register 0	DBA + 000080h		PR-LK0
64-bit Factory-Programmable Protection Register	DBA + 000081h–000084h		Protection Register Data
64-bit User-Programmable Protection Register	DBA + 000085h–000088h		Protection Register Data
Lock Protection Register 1	DBA + 000089h		PR-LK1
16x128 bit User-Programmable Protection Registers	DBA + 00008Ah–0000109h		Protection Register Data

Notes:
1. DBA = Device Base Address. (A[MAX:18] = DBA). Numonyx reserves other configuration address locations.
2. BBA = Block Base Address.

Table 10: Device ID Table

Device	Device Code (Byte/Word)			Mode
	Hex	Binary		
		High Byte	Low Byte	
128 Mb	881E	10001000	00011110	Top Boot
128 Mb	8821	10001000	00100001	Bottom Boot

7.3 Read Query Command

The Query space comes to the foreground and occupies the device address range supplied by the Read Query command address. The mode outputs Common Flash Interface (CFI) data when the device addresses are read. Appendix A, “Common Flash Interface” on page 84 shows the query mode information and addresses. Write the Read Array command to return to read array mode.

The read performance of this CFI data follows the same timings as the main array.

7.4 Other ID Mode Data

Other ID mode data besides the Protection registers (such as block locking information and the device JEDEC ID) may be accessed as long as there are no ongoing write or erase operations.

7.5 Query (CFI) Data

Query data is read by sending the Read Query command to the device. Reading the Query data is subject to the same restrictions as reading the Protection Registers.

8.0 Program Operations

There are five kinds of write operations available in Numonyx® Omneo™ P8P PCM.

- Word Program (40h, or 10h)
- Bit Alterable Word Write (42h)
- Buffered Program (E8h)
- Bit Alterable Buffered Write (EAh)
- Buffered Program on all 1's (DEh)

Writing a program command to the device initiates internally timed sequences that write the requested word.

The WSM executes a sequence of internally timed events to write desired bits at the addressed location and verify that the bits are sufficiently written. For Word Programming the memory changes specifically addressed bits to "0". "1" bits do not change the memory cell contents. This allows individual data-bits to be programmed ("0") while "1" bits serve as data masks. For Bit Alterable Word Write, the memory cell can change from "0" to "1" or "1" to a "0".

The Status Register can be examined for write progress and errors by reading any address within the device during a write operation. Issuing a Read Status Register command brings the Status Register to the foreground allowing write progress to be monitored or detected at other device addresses. Status Register bit SR.7 indicates device write status while the write sequence executes. CE# or OE# toggle (during polling) updates the Status Register. Valid commands that can be issued to the writing device during write are Read Status Register, Write Suspend, Read Identifier, Read Query, and Read Array. However Read Array will return unknown data while the device is busy.

When writing completes, Status Register bit SR.4 indicates write success if zero (0) or failure if set (1). If SR.3 is set (1), the WSM couldn't execute the write command because V_{PP} was outside acceptable limits. If SR.1 is set (1), the write operation targeted a locked block and was aborted. Attempting to write in an erase suspended block will result in failure and SR.4 will be set (1).

After examining the Status Register, it should be cleared by the Clear Status Register command before issuing a new command. The device remains in Status Register mode until another command is written to that device. Any command can follow once writing completes.

8.1 Word Program

The system processor writes the Word Program Setup command (40h/10h) to the device followed by a second write that specifies the address and data to be programmed. The device accessed during both of the command cycles automatically outputs Status Register data when the device address is read. The device accessed during the second cycle (the data cycle) of the program command sequence will be where the data is programmed. See Section 32, "Buffer Program or Bit Alterable Buffer Write Flowchart" on page 75.

When V_{PP} is greater than V_{PPLK}, program-and erase-currents are drawn through the V_{CC} input. If V_{PP} is driven by a logic signal, V_{PP} must remain above V_{PPMIN} to perform in-system PCM modifications. Figure 5, "Example V_{PP} Power Supply Configuration" on page 27 shows PCM power supply usage in various configurations.

8.2 Bit Alterable Word Write Command

The Bit Alterable Word Write Command executes just like Word Program Command (40h/10h), using a two-write command sequence. The Bit Alterable Write Setup command (42h) is written to the CUI followed by the specific address and data to be written. The WSM will start executing the programming algorithm, but the data written to CUI will be directly overwritten into the PCM memory unlike flash memory, which can only be written from 1 to 0 without a prior erase of the entire block. See Table 12, “Bit Alterability vs. Flash Bit-Masking” on page 26. This overwrite function eliminates Flash Bit Masking, which means software cannot use a “1” in a data mask to produce no change of the memory cell, as might occur with floating gate flash.

8.3 Buffered Program Command

A Buffered Program command sequence initiates the loading of a variable number of words, up to the buffer size (32 words), into the program buffer and after that into the PCM device. First, the Buffered Program setup command is issued along with the Block Address (Section 32, “Buffer Program or Bit Alterable Buffer Write Flowchart” on page 75). When Status Register bit 7 is set to 1, the buffer is ready for loading. Now a word count is given to the part with the Block Address.

On the next write, a device starting address is given along with the Program Buffer data. Subsequent writes provide additional device addresses and data, depending on the count. All subsequent addresses must lie within the starting address plus the buffer size. Maximum programming performance and lower power are obtained by aligning the starting address at the beginning of a 32 word boundary. A misaligned starting address is not allowed and will result in invalid data. After the final buffer data is given, a Program Buffer Confirm command is issued. This initiates the WSM (Write State Machine) to begin copying the buffer data to the PCM array.

If a command other than Buffered Program Confirm command (D0h) is written to the device, an “Invalid Command/Sequence” error will be generated and Status Register bits SR.5 and SR.4 will be set to a “1.” For additional buffer writes, issue another Program Buffer Setup command and check SR.7. If an error occurs while writing, the device will stop writing, and Status Register bit SR.4 will be set to a “1” to indicate a program failure. The internal WSM verify only detects errors for “1”s that do not successfully program to “0”s.

If a program error is detected, the Status Register should be cleared by the user before issuing the next program command. Additionally, if the user attempts to program past the block boundary with a Program Buffer command, the device will abort the Program Buffer operation. This will generate an “Invalid Command/Sequence” error and Status Register bits SR.5 and SR.4 will be set to a “1. All bus cycles in the buffered programming sequence should be addressed to the same block. If a buffered programming is attempted while the $V_{PP} \leq V_{PPLK}$, Status Register bits SR.4 and SR.3 will be set to “1”.

Buffered write attempts with invalid V_{CC} and V_{PP} voltages produce spurious results and should not be attempted. Buffered program operations with $V_{IH} < RST\# < V_{HH}$ may produce spurious results and should not be attempted.

Successful programming requires that the addressed block’s locking status to be cleared. If the block is locked down, then the WP# pin must be raised high and then the block could be unlocked to execute a program operation. An attempt to program a locked block results in setting of SR.4 and SR.1 to a ‘1’ (i.e. “Error in Programming”).

8.4 Bit Alterable Buffer Write

The Bit Alterable Buffer Write command sequence is the same as for Buffer Program. For command sequence see Section 8.3, "Buffered Program Command" on page 25. The primary difference between the two Buffer commands is when the Write State Machine starts executing, the data written to the buffer will be directly overwritten into the PCM memory, unlike Flash Memory, which can only go from "1" to "0" before an erase of the entire block. See Table 12, "Bit Alterability vs. Flash Bit-Masking" on page 26. This overwrite function eliminates Flash Bit Masking, which means software cannot use a "1" in a data mask for no change of the memory cell, as might occur with floating gate flash.

The advantage of Bit Alterability is no block erase is needed prior to writing a block, which minimizes system overhead for software management of data, and ultimately improves latency, determinism, and reduces power consumption because of reduction of system overhead. Storing of counter variables can easily be handled by using PCM memory because a "0" can change to a "1" or a "1" can change to a "0".

Table 11: Buffered Programming and Bit Alterable Buffer Write Timing Requirements

Alignment	Programming Time	Example
32-word/64-byte Aligned	$t_{PROG/PB}$	Start Address = 1FFF10h; End Address = 1FFF2Fh

Table 12: Bit Alterability vs. Flash Bit-Masking

Programming Function	Command Issued	Memory Cell Current State	Data From User	Memory Cell After Programming
Flash Bit-Masking	40h or E8h	0	0	0
	40h or E8h	0	1	0
	40h or E8h	1	0	0
	40h or E8h	1	1	1
Bit Alterability	42h or EAh	0	0	0
	42h or EAh	0	1	1
	42h or EAh	1	0	0
	42h or EAh	1	1	1

8.5 Bit Alterable Buffer Program

This mode is sometimes referred to as PreSET Buffered Program.

'Program on all 1s' is similar to program mode ("1"s treated as masks; "0"s written to cells) with the assumption that all the locations in the addressed page have previously been SET ("1"s). [Performance of Buffer Program on All 1s expected to be better than buffered program mode because the pre-read step before programming is eliminated.]

The command sequence for Buffered Program on all 1s is the same as Buffered Program Command (E8h).

8.6 Program Suspend

Issuing the Program Suspend command while programming suspends the programming operation. This allows data to be accessed from the device other than the one being programmed. The Program Suspend command can be issued to any device address. A program operation can be suspended to perform reads only. Additionally, a program operation that is running during an erase suspend can be suspended to perform a read operation.

When a programming operation is executing, issuing the Program Suspend command requests the WSM to suspend the programming algorithm at predetermined points. The device continues to output Status Register data after the Program Suspend command is issued. Programming is suspended when Status Register bits SR[7,2] are set.

To read data from the device, the Read Array command must be issued. Read Array, Read Status Register, Read Device Identifier, Read CFI, and Program Resume are valid commands during a program suspend.

During a program suspend, deasserting CE# places the device in standby, reducing active current. V_{PP} must remain at its programming level, and WP# must remain unchanged while in program suspend. If RST# is asserted, the device is reset.

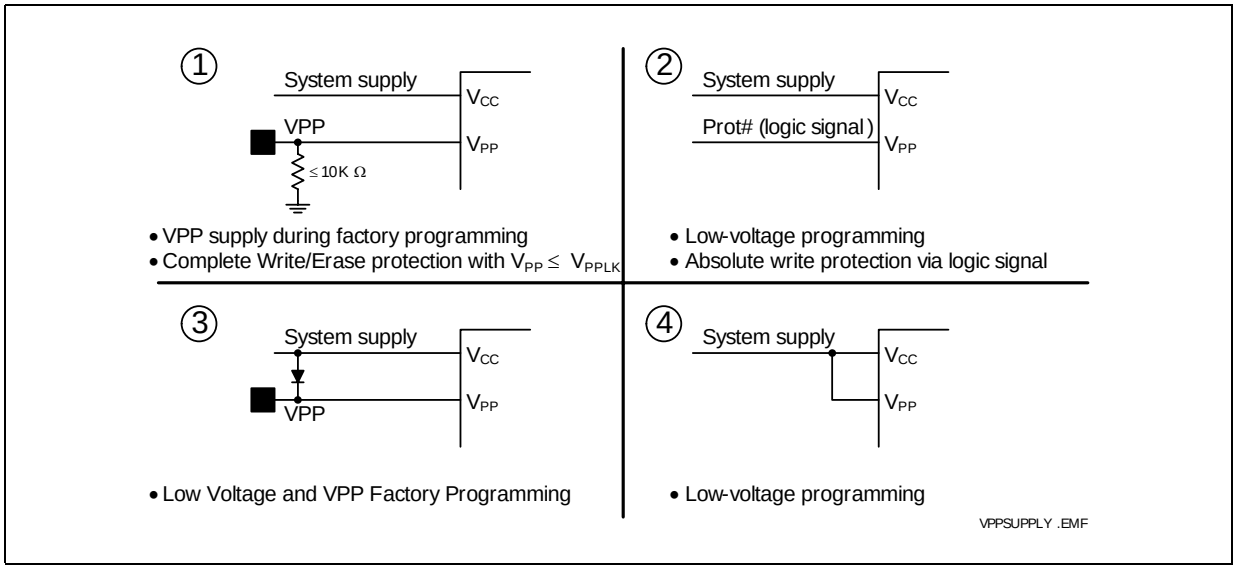
8.7 Program Resume

The Resume command instructs the device to continue programming, and automatically clears Status Register bits SR[7,2]. This command can be written to any address. If error bits are set, the Status Register should be cleared before issuing the next instruction. RST# must remain deasserted.

8.8 Program Protection

Holding the V_{PP} input at V_{IL} provides absolute hardware write protection for all PCM-device blocks. If V_{PP} is below V_{PPLK}, write or erase operations halt and an error is posted in Status Register bit SR.3. The block lock registers are not affected by the V_{PP} level; they may be modified and read even if V_{PP} is below V_{PPLK}.

Figure 5: Example V_{PP} Power Supply Configuration



9.0 Erase

Unlike floating gate flash, PCM does not require a high voltage block erase operation to change all the bits in a block to “1.” As a bit alterable technology, each bit is capable of independently being changed from a “0” to a “1” and from a “1” to a “0”. With floating gate flash, a high voltage potential must be placed in parallel upon a group of bits called an erase block. Each bit within the block may be changed independently from “1” to a “0”, but only may be changed from a “1” to a “0” through a grouped erase operation. To maintain compatibility with legacy flash system software, Numonyx® Omneo™ P8P PCM mimics or emulates a flash erase by writing each bit within a block to “1”, thereby emulating flash-style erase.

9.1 Block Erase

The system processor writes the Erase Setup command (20h) to the device followed by a second Confirm (D0h) command write that specifies the address of the block to be erased. The device during both of the command cycles automatically outputs Status Register data when the device address is read. See Section 33, “Block Erase Flowchart” on page 76.

After writing the command, the device automatically enters read status mode. The device Status Register bit SR.7 will be set (“1”) when the erase completes. If the erase fails, Status Register bit SR.5 will be set (“1”). SR.3 = “1” indicates an invalid V_{PP} voltage. SR.1 = “1” indicates an erase operation was attempted on a locked block. CE# or OE# toggle (during polling) updates the Status Register.

If an error bit is set, the Status Register can be cleared by issuing the Clear Status Register command before attempting the next operation. The device will remain in Status Register mode until another command is written to the device. Any command can follow once erase completes. Only one block can be in erase mode at a time.

9.2 Erase Suspend Command

The Write/Erase Suspend command halts an in-progress write or erase operation. The command can be issued at any device address. The Suspend command allows data to be accessed from memory locations other than the one block being written or the block being erased.

A Write operation can be suspended to perform reads only at any location except the address being programmed. An Erase operation can be suspended to perform either a write or a read operation within any block except the block that is erase suspended. A Write command nested within a suspended Erase can subsequently be suspended to read yet another location. Once the write/erase process starts, the Suspend command requests that the WSM suspend the write/erase sequence at predetermined points in the algorithm. An operation is suspended when status bits SR.7 and SR.6 and/or SR.2 display “1.” $t_{SUSP/P}$/$t_{SUSP/E}$ specifies suspend latency.

To read data from other blocks within the device (other than an erase-suspended block), a Read Array command can be written. During Erase Suspend, a Write command can be issued to a block other than the erase-suspended block. Block erase cannot resume until write operations initiated during erase suspend complete. Read Array, Read Status Register, Read Identifier (ID), Read Query, and Write Resume are valid commands during Write or Erase Suspend. Additionally, Clear Status Register, Program, Write Suspend, Erase Resume, Lock Block, Unlock Block, and Lock-Down Block are valid commands during Erase Suspend.

During a suspend, CE# = V_{IH} places the device in standby state, which reduces supply current. V_{PP} must remain at its program level and WP# must remain unchanged while in suspend mode.

The Resume (D0h) command instructs the WSM to continue writing/erasing and automatically clears Status Register bits SR.2 (or SR.6) and SR.7. If Status Register error bits are set, the Status Register can be cleared before issuing the next instruction. RST# must remain at V_{IH}. See Section 31, "Write Suspend/Resume Flowchart" on page 74 and Section 34, "Erase Suspend/Resume Flowchart" on page 77.

If software compatibility with the Numonyx™ P33 device is desired, a minimum $t_{ERS/SUSP}$ time (See Section 17.0, "Program and Erase Characteristics" on page 71) should elapse between an Erase command and a subsequent Erase Suspend command to ensure that the device achieves sufficient cumulative erase time. Occasional Erase-to-Suspend interrupts do not cause problems, but out-of-spec Erase-to-Suspend commands issued too frequently to a P33 device may produce uncertain results. However, this specification is not required for this PCM device.

9.3 Erase Resume

The Erase Resume command instructs the device to continue erasing, and automatically clears status register bits SR[7,6]. This command can be written to any address. If status register error bits are set, the Status Register should be cleared before issuing the next instruction. RST# must remain deasserted (see Figure 31, "Write Suspend/Resume Flowchart" on page 74).

10.0 Security Mode

The device features security modes used to protect the information stored in the flash memory array. The following sections describe each security mode in detail.

10.1 Block Locking

There are two types of block locking on Numonyx® Omneo™ P8P PCM:

- Zero Latency Block Locking
- Selectable One Time Programmable (OTP) Block Locking

  This type of locking allows for permanent locking of the parameter blocks and 3 main blocks.

10.1.1 Zero Latency Block Locking

Individual instant block locking protects code and data. It allows software to control block locking or it can require hardware interaction before locking can be changed. Any block can be locked or unlocked with no latency. Locked blocks cannot be written or erased; they can only be read. Write or erase operations to a locked block returns a Status Register bit SR.1 error. The following sections discuss the locking operations. State [WP#, LAT1, LAT0] specifies lock states (WP# = WP# state, LAT1= internal Block Lock Down latch status, LAT0 = internal Block Lock latch status). Figure 6, “Block Locking State Diagram” on page 33 defines possible locking states. The following summarizes the locking functionality.

- All blocks power-up in the locked state. Then Unlock and Lock commands can unlock or lock them
- The Lock-Down command locks and prevents a block from being unlocked when WP# = V_{IL}

  WP# = V_{IH} overrides lock-down so commands can unlock/lock blocks

  If a previously locked-down block is given a Lock/Unlock/Lock-Down command and WP# returns to V_{IL} then those blocks will return to lock-down

  Lock-Down is cleared only when the device is reset or powered-down.

The block lock registers are not affected by the V_{PP} level; they may be modified and read even if V_{PP} is below V_{PPLK}.

The following sections describe how to lock, unlock, and lock-down a block. Table 14 on page 32 shows the state table for the locking functions. See also Section 35, “Locking Operations Flowchart” on page 78.

10.1.2 Lock Block

All blocks default power-up or reset state is locked (states [001] or [101]) to fully protect it from alteration. Write or erase operations to a locked block return a Status Register bit SR.1 error. The Lock Block command sequence can lock an unlocked block.

Table 13: Block Locking Truth Table

V_{PP}	WP#	RST#	Block Write Protection	Block Lock Bits
X	X	V_{IL}	All blocks write/erase protected	Block lock bits may not be changed
$\leq V_{PPLK}$	V_{IL}	V_{IH}	**All** blocks write/erase protected	Lock-Down block states may **not** be changed

Table 13: Block Locking Truth Table

V_{PP}	WP#	RST#	Block Write Protection	Block Lock Bits
$\leq V_{PPLK}$	V_{IH}	V_{IH}	**All** blocks write/erase protected	All Lock-Down block states **may be** changed
$> V_{PPLK}$	V_{IL}	V_{IH}	All Lock-Down and Locked blocks write/erase protected	Lock-Down block states may **not** be changed
$> V_{PPLK}$	V_{IH}	V_{IH}	All Lock-Down and Locked blocks write/erase protected	All Lock-Down block states **may be** changed

10.1.3 Unlock Block

The Unlock Block command unlocks locked blocks (if block isn't locked-down) so they can be programmed or erased. Unlocked blocks return to the locked state at device reset or power-down.

10.1.4 Lock-Down Block

Locked-down blocks (state 3 or [011]) are protected from write and erase operations (just like locked blocks), but software commands alone cannot change their protection state. When WP# is V_{IH}, the lock-down function is disabled (state 7 or [111]), and an Unlock command (60h/D0h) must be issued to unlocked locked-down block (state 6 or [110]), prior to modifying data in these blocks. To return an unlocked block to locked-down state, a Lock command (60h/01h) must be issued prior to changing WP# to V_{IL} (state 7 or [111] and then state 3 or [011]). A locked or unlocked block can be locked-down by writing the Lock-Down Block command sequence. Locked-down blocks revert to the locked state at device reset or power-down.

10.1.5 WP# Lock-Down Control

WP# = V_{IH} overrides the block lock-down. See Table 13, "Block Locking Truth Table" on page 30. The WP# signal controls the lock-down function. WP# = 0 protects lock-down blocks [011] from write, erase, and lock status changes. When WP# = 1, the lock-down function is disabled [111] and a software command can individually unlock locked-down blocks [110] so they can be erased and written. When the lock-down function is disabled, locked-down blocks remain locked, and must first be unlocked by writing the Unlock command prior to modifying data in these blocks. These blocks can then be re-locked [111] and unlocked [110] while WP# remains high.

When WP# goes low, blocks in re-locked state [111] returns to locked-down state [011]. However, WP# going low changes blocks at unlocked state [110] to [010] or "virtual lock-down" state. When the lock status of a "virtual lock-down" blocks is read, it appears to be a "locked-down" state to user when WP# is V_{IL}. Blocks in "virtual lock-down" will be immediately unlocked when WP# is V_{IH}. Therefore, to avoid "virtual lock-down", a Lock command must be issued to an unlocked block prior to WP# going low. Device reset or power-down resets all blocks to the locked state[101] or [001], including locked-down blocks.

10.1.6 Block Lock Status

Every block's lock status can be read in the device's read identifier mode. To enter this mode, write 90h to the device. Subsequent reads at Block base-address + 00002h output that block's lock status. Data bits DQ_0 and DQ_1 represent the lock status. DQ_0 indicates the block lock/unlock state as set by the Lock command and cleared by the Unlock command. It is also automatically set when entering Lock-Down. DQ_1 indicates lock-down state as set by the Lock-Down command. It cannot be cleared by software, only by device reset or power-down. See Table 14, "Block Locking State Transitions" on page 32.

10.1.7 Locking Operations During Erase Suspend

Block lock configurations can be performed during an erase suspend by using the standard locking command sequences to unlock, lock, or lock-down a block. This is useful when another block needs to be updated while an erase operation is suspended.

To change block locking during an erase operation, first write the Erase Suspend command, then check the Status Register until it indicates that the erase operation has suspended. Next write the desired lock command sequence to a block; the lock state will be changed. After completing lock, read, or program operations, resume the erase operation with the Erase Resume command (D0h).

If a block is locked or locked-down during a suspended erase of the same block, the locking status bits will change immediately. But, when resumed, the erase operation will complete. Locking operations cannot occur during write suspend. Appendix A, "Write State Machine" on page 80 shows valid commands during erase suspend.

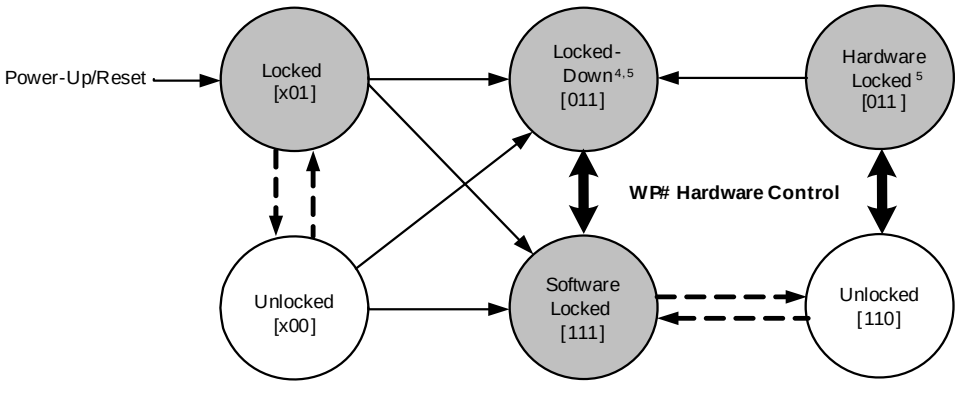
Nested lock or write commands during erase suspend can return ambiguous Status Register results. 60h followed by 01h commands lock a block. A Configuration Setup command (60h) followed by an invalid command produces a lock command Status Register error (SR.4 and SR.5 = 1). If this error occurs during erase suspend, SR.4 and SR.5 remain at 1 after the erase resumes. When erase completes, the previous locking command error hides the Status Register's erase errors. A similar situation occurs if a write operation error is nested within an erase suspend.

Table 14: Block Locking State Transitions

Current State				Erase / Write Allowed?(1)	Lock Command Input Result (Next State) (5)			WP# Toggle Result (Next State)	Locking Status Readout	
WP#	LAT 1	LAT 0	Name		UnLock	Lock	Lock-Down		D1	D0
0	0	0	Unlocked	Yes	000	001	011	100	0	0
0	0	1	Locked (default) (1)	No	000	001	011	101	0	1
0	1	0	Virtual lock-down (4)	No	011	011	011	110	1	1
0	1	1	Locked-Down	No	011	011	011	111	1	1
1	0	0	Unlocked	Yes	100	101	111	000	0	0
1	0	1	Locked	No	100	101	111	001	0	1
1	1	0	Lock-Down Disabled	Yes	110	111	111	010	1	0
1	1	1	Lock-Down Disabled	No	110	111	111	011	1	1

Notes:
1. Additional illegal states are shown but are not recommended for normal, non-erroneous operational modes.
2. "Erase/Write Allowed?" shows whether a block's current locking state allows erase or write.
3. At power-up or device reset, blocks default to locked state [001] if WP# = 0, the recommended default.
4. Blocks in "virtual lock-down" appear to be in locked-down state when WP#=V_{IL}. WP# = 1 changes [010] to unlocked state [110].
5. This column shows results of writing the four locking commands via WP# toggle from the current locking state.

Figure 6: Block Locking State Diagram



Notes
1. [a,b,c] represents [WP#, DQ1, DQ0]. X = Don't Care
2. DQ1 indicates Block Lock-Down status DQ1 = '0', Lock-Down has not bee issued to this block DQ1 = '1', Lock-Down has been issued to this block
3. DQ0 indicates block lock status. DQ0 = '0', block is unlocked. DQ0 = '1', block is locked.
4. Locked-down = Hardware + Software locked.
5. [011] states should be tracked system software to determine difference between Hardware Locked and Locked-Down states

10.2 Permanent One Time Programmable (OTP) Block Locking

The parameter blocks and first 3 main blocks for a bottom parameter device (or if device configured as a top parameter device this would be the last 3 main blocks and the parameter blocks) can be made OTP, so further write and erase operations to these blocks are disallowed, effectively permanently programming the blocks. This is achieved by programming bits 2, 3, 4, and 5 in the PR-LOCK0 register at offset 0x80 in ID Space. The OTP locking bit mapping may be seen in Table 15, "Selectable OTP Block Locking Feature" on page 34.

Bit 6 in the PR-LOCK0 register at offset 0x80 in ID space is defined as the Configuration Lock bit. When bit 6 is cleared (at zero), the device shall disable further programming of the OTP Lock bits, thereby effectively "freezing" their state. Putting bit 6 at zero shall not affect the ability to write any other bits in the non OTP regions or in the System Protection Registers. Reference Table 16, "Selectable OTP Block Locking Programming of PR-LOCK0" on page 34 for Configuration Lock bit (Bit 6 in PR-LOCK0) control of allowed states when other bits of the register are programmed.

The read operations of these permanently locked blocks are always supported regardless of the state of their corresponding Permanent Lock bits. Zero Latency Block Locking must be used until the block is permanently locked with the OTP Block Locking. Program and erase operations for these blocks remain fully supported until that block's Permanent Lock bit is cleared.

Program or erase operations to a permanently locked block returns a Status Register bit SR.1 error.

Programming of the Permanent OTP Block Locking bits is not allowed during Erase Suspend of a Permanent Lockable Block.

Note: The Selectable Block Locking will not be indicated in the Zero Latency Block Lock Status. See Section 10.1.6, "Block Lock Status" on page 31 for more information. Read PR-LOCK0 register to determine Block Lock Status for these blocks.

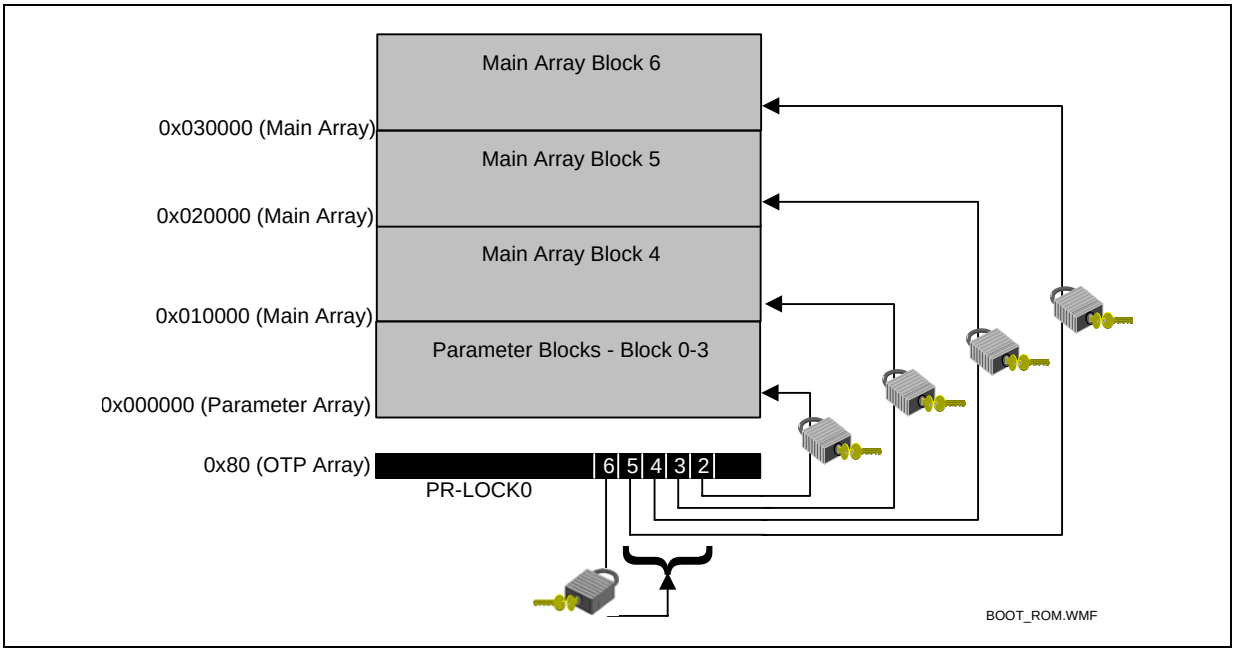
Table 15: Selectable OTP Block Locking Feature

Bit Number @ Offset 0x80 in CFI Space	Function When Set ('1b)	Function When Cleared ('0b)
2	Blocks not permanently locked	Write/erase disabled for all parameter blocks Bottom Boot - Blocks 0-3 Top Boot 128M - Blocks 127-130
3	Block not permanently locked	Write/erase disabled for first Main Block Bottom Boot - Block 4 Top Boot 128M - Block 126
4	Block not permanently locked	Write/erase disabled for second Main Block Bottom Boot - Block 5 Top Boot 128M - Block 125
5	Block not permanently locked	Write/erase disabled for third Main Block Bottom Boot - Block 6 Top Boot 128M - Block 124
6	Able to change PR-LOCK0[5:2] Bits	Program disabled for PR-LOCK0[5:2]

Table 16: Selectable OTP Block Locking Programming of PR-LOCK0

Bit 6	Program to [5:2]	Program to [1:0]	Status Register	Abort Program	Status of Data in 80H OTP Space
unlocked	don't care	don't care	no fail bits set	NO	Changed
locked	YES	YES	program fail/ lock fail	YES	No Change
locked	YES	NO	program fail/ lock fail	YES	No Change
locked	NO	YES	no fail bits set	NO	Changed

Figure 7: Selectable OTP Locking Illustration (Bottom Parameter Device Example)



10.2.1 WP# Lock-Down Control for Selectable OTP Lock Blocks

Once the block has been permanently locked with OTP bit, WP# at VIH does not override the lock down of the blocks those bits control.

10.2.2 Selectable OTP Locking Implementation Details

Clearing (write to “0”) any of the four Permanent Lock bits shall effectively cause the following commands to fail with a block locking error when issued to their corresponding blocks: Buffer Program command, Bit-Alterable Buffer Write command, Word Program command, Bit-Alterable Word Write command, and Erase command. No other commands shall be affected.

Programming the Permanent Lock bits or the Configuration Lock bit shall be done using the Protection Register Programming command. As with all bits in the CFI/OTP space once the Permanent Lock or the Configuration bits are programmed, they may not be erased (set) again.

11.0 Registers

11.1 Read Status Register

The device's Status Register displays program and erase operation status. A device's status can be read after writing the Read Status Register command. The Status Register can also be read following a Program, Erase, or Lock Block command sequence. Subsequent single reads from the device outputs its status until another valid command is written.

The last of OE# or CE# falling edge latches and updates the Status Register content. DQ[7:0] output is the Status Register bits; DQ[15:8] output 00h. See Table 17, "Status Register Definitions" on page 36.

Issuing a Read Status, Block Lock, Program, or Erase command to the device places it in the Read Status mode. Status Register bit SR.7 (DWS — Device Write Status) provides program/erase status of the device. Status Register bits SR.1-SR.6 present information about the WSM's program, erase, suspend, V_{PP}, and block-lock status mode.

Table 17: Status Register Definitions

DRS	ESS	ES	PS	VPPS	PSS	DPS	PRW
SR.7	**SR.6**	**SR.5**	**SR.4**	**SR.3**	**SR.2**	**SR.1**	**SR.0**
Status Register Bits			**Notes:**				
SR.7 = Device Write/Erase Status (DWS) 0 = Device WSM is Busy 1 = Device WSM is Ready			SR.7 indicates erase or program completion in the device. SR.1–6 are invalid while SR.7 = "0."				
SR.6 = Erase Suspend Status (ESS) 0 = Erase in progress/ completed 1 = Erase suspended			After issuing an Erase Suspend command, the WSM halts and sets (1) SR.7 and SR.6. SR.6 remains set until the device receives an Erase Resume command.				
SR.5 = Erase Status (ES) 0 = Successful erase 1 = Erase error			SR.5 is set (1) if an attempted erase failed. A Command Sequence Error is indicated when SR.4, SR.5 and SR.7 are set.				
SR.4 = Program Status (PS) 0 = Successful write 1 = Write error			SR.4 is set (1) if the WSM failed to program. A Command Sequence Error is indicated when SR.4, SR.5 and SR.7 are set.				
SR.3 = V_{PP} Status (VPPS) 0 = V_{PP} OK 1 = V_{PP} low detect, operation aborted			The WSM indicates the V_{PP} level after program or erase starts. SR.3 does not provide continuous V_{PP} feedback and isn't guaranteed when $V_{PP} < V_{PPLK}$				
SR.2 = Program Suspend Status (PSS) 0 = Write in progress/ completed 1 = Write suspended			After receiving a Write Suspend command, the WSM halts execution and sets (1) SR.7 & SR.2, which remains set until a Resume command is received.				
SR.1 = Device Protect Status (DPS) 0 = Unlocked 1 = Aborted erase/program attempt on locked block			If an erase or program operation is attempted to a locked block (if WP# = V_{IL}), the WSM sets (1) SR.1 and aborts the operation.				
SR.0 Super Page Write Status (PRW) 0 = Reserved 1 = Reserved			Reserved				

11.1.1 Clear Status Register Command

The Clear Status Register command clears the Status Register. The command functions independently of the applied V_{PP} voltage. The WSM can set (1) Status Register bits SR[7:0] and clear (0) bits 2, 6, and 7. Because bits 1, 3, 4 and 5 indicate various error conditions, they can only be cleared by the Clear Status Register command. By allowing system software to reset these bits, several operations (such as cumulatively programming several addresses or erasing multiple blocks in sequence) may be performed before reading the Status Register to determine error occurrence. The Status Register should be cleared before beginning another command or sequence. Device reset (RST# = V_{IL}) also clears the Status Register.

11.2 System Protection Registers

The device contains two 64-bit, and sixteen 128-bit individually lockable protection registers that can increase system security or hinder device substitution by containing values that mate the PCM component to the system's CPU or ASIC.

One 64-bit protection register is programmed at the Numonyx factory with an non-changeable unique 64-bit number. The other 64-bit and sixteen 128-bit protection registers are blank so customers can program them as desired. Once programmed, each customer segment can be locked to prevent further reprogramming.

11.2.1 Read Protection Register

The Read Identifier command allows Protection register data to be read 16 bits at a time from addresses shown in Table 9, "Read Identifier Table" on page 22. To read the Protection Register, first issue the Read Device Identifier command at Device Base Address to place the device in the Read Device Identifier mode. Next, perform a read operation at the device's base address plus the address offset corresponding to the register to be read. Table 9, "Read Identifier Table" on page 22 shows the address offsets of the Protection Registers and Lock Registers. Register data is read 16 bits at a time. Refer Appendix , "Protection Register Addressing" on page 39.

11.2.2 Program Protection Register

The Protection Program command should be issued followed by the data to be programed at the specified location. It programs the 64 user protection register 16 bits at a time. Table 9, "Read Identifier Table" on page 22 and in Table 18 on page 39 show allowable addresses. See also Figure 36, "Protection Register Programming Flowchart" on page 79. Addresses A[MAX:11] are ignored when programming the OTP, and OTP program will succeed if A[10:1] are within the prescribed protection addressing range; otherwise an error is indicated by SR4 = 1.

11.2.3 Lock Protection Register

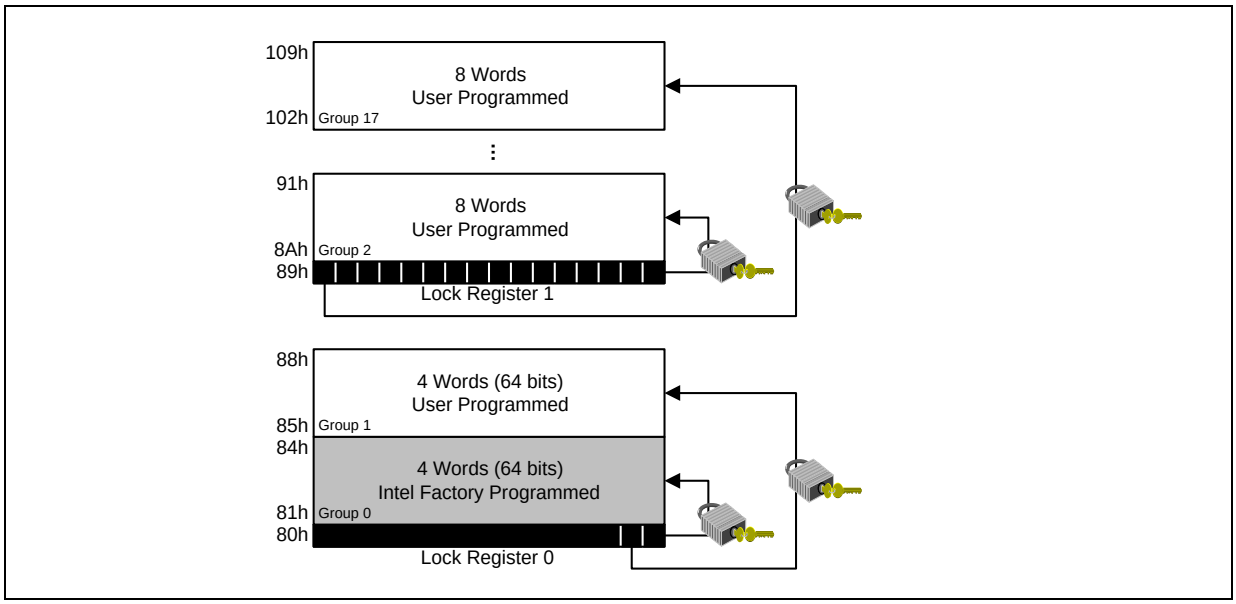
Each of the protection registers are lockable by programming their respective lock bits in the PR-LOCK0 or PR-LOCK1 registers. Bit 0 of the Lock-Register -0 is programmed by Numonyx to lock-in the unique device number. The physical address of the PR-LOCK0 register is 80h as seen in Figure 8, "Protection Register Memory Map" on page 38. Bit 1 of the Lock-Register -0 can be programmed by the user to lock the upper 64-bit portion. (Refer Table 18, "Protection Register Addressing" on page 39.). The bits in both PR-LOCK registers are made of "PCM cells" that may only be programmed to '0' and may not be altered.

Note: Bit0 of the Lock-Register, PR-LOCK0, is a don't care, so users must mask out this bit when reading PR-LOCK0 register. This number is guaranteed to persist through board attach.

For the 2K OTP space, there exists an additional 16-bit lock register called PR_LOCK1. Each bit in the PR_LOCK1 register locks a 128-bit segment of the 2K-OTP space. Therefore, the 16 128-bit segments of the 2K OTP space can be locked individually. Hence, any 128-bit segment can be first programmed and then locked using the protection program command followed by protection register data. The PR-LOCK1 register is physically located at the address 89h as shown in the Figure 8, “Protection Register Memory Map” on page 38.

After PR-LOCK register bits have been programmed, no further changes can be made to the protection registers' stored values. Protection Program commands written to a locked section result in a Status Register error (Program Error bit SR.4 and Lock Error bit SR.1 are set to 1). Once locked, Protection register states are not reversible.

Figure 8: Protection Register Memory Map



11.2.3.1 OTP Protection Register Addressing details

Table 18: Protection Register Addressing

Word	Use	ID Offset	A8	A7	A6	A5	A4	A3	A2	A1
LOCK	Both	DBA + 000080h	1	0	0	0	0	0	0	0
0	Numonyx	DBA + 000081h	1	0	0	0	0	0	0	1
1	Numonyx	DBA + 000082h	1	0	0	0	0	0	1	0
2	Numonyx	DBA + 000083h	1	0	0	0	0	0	1	1
3	Numonyx	DBA + 000084h	1	0	0	0	0	1	0	0
4	Customer	DBA + 000085h	1	0	0	0	0	1	0	1
5	Customer	DBA + 000086h	1	0	0	0	0	1	1	0
6	Customer	DBA + 000087h	1	0	0	0	0	1	1	1
7	Customer	DBA + 000088h	1	0	0	0	1	0	0	0

Note: Addresses A_9-A_{23} should be set to zero.

Table 19: 2K OTP Space Addressing

Word	Use	ID Offset	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
Lock	Customer	DBA+000089h	0	0	0	0	0	1	0	0	0	1	0	0	1
0	Customer	DBA+00008Ah	0	0	0	0	0	1	0	0	0	1	0	1	0
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
127	Customer	DBA+000109h	0	0	0	0	1	0	0	0	0	1	0	0	1

Note: DBA - Device Base Address. Typically this would start from Address 0.

12.0 Serial Peripheral Interface (SPI)

12.1 SPI Overview

A Serial Peripheral Interface has been added as a secondary interface on Numonyx® Omneo™ P8P PCM to enable low cost, low pin count on-board programming. This interface gives access to the P8P memory by using only seven signals, instead of a conventional parallel interface that may take 45 signals or more. The seven signals consist of six SPI-only signals plus one signal that is shared with the conventional interface.

When the SPI mode is enabled, all non-SPI P8P output signals are tri-stated, and all non-SPI P8P inputs signals are ignored (made “don't care”). When the conventional interface is enabled, the SPI-only output is tri-stated, and the SPI-only inputs are ignored (made “don't care”).

Note: The SPI interface can only be enable upon power-up, and to enable this interface the SERIAL pin must be tied to Vcc for the interface to be factional. Once the SPI interface is enable it is the only interface that can be accessed until the part is powered down.

The SPI mode may be disabled. Please contact Numonyx for more information.

12.2 SPI Signal Names

For P8P, the six additional SPI-only signals are implemented in addition to the power pins. V_{CC}, V_{CCQ}, and V_{PP} are valid power pins during Serial mode and must be connected during SPI mode operation. Four of the six additional SPI signals do not share functions with the regular interface. For pin and signal descriptions of all P8P pins see Table 5, “Ball/Pin Descriptions” on page 16. Two pins are shared between the interface modes: S# is the same pin as CE#, and HOLD# is the same pin as OE#. The signals that are unique to the SPI mode and require a separate connection are C, D, Q, and SERIAL.

12.3 SPI Memory Organization

The memory is organized as:

- 16,772,216 bytes (8 bits each)
- 128 sectors (128 Kbytes each)
- 131,072 pages (64 bytes each)

Each page can be individually programmed (bits are programmed from ‘1’ to ‘0’) or written (bit alterable: ‘1’ can be altered to ‘0’ and ‘0’ can be altered to ‘1’). The device is sector or bulk erasable (bits are erased from ‘0’ to ‘1’).

Table 6. Memory organization

Sector	Address range		Sector	Address range	
127	FE0000	FFFFFF	63	7E0000	7FFFFF
126	FC0000	FDFFFF	62	7C0000	7DFFFF
125	FA0000	FBFFFF	61	7A0000	7BFFFF
124	F80000	F9FFFF	60	780000	79FFFF
123	F60000	F7FFFF	59	760000	77FFFF
122	F40000	F5FFFF	58	740000	75FFFF
121	F20000	F3FFFF	57	720000	73FFFF
120	F00000	F1FFFF	56	700000	71FFFF
119	EE0000	EFFFFF	55	6E0000	6FFFFF
118	EC0000	EDFFFF	54	6C0000	6DFFFF
117	EA0000	EBFFFF	53	6A0000	6BFFFF
116	E80000	E9FFFF	52	680000	69FFFF
115	E60000	E7FFFF	51	660000	67FFFF
114	E40000	E5FFFF	50	640000	65FFFF
113	E20000	E3FFFF	49	620000	63FFFF
112	E00000	E1FFFF	48	600000	61FFFF
111	DE0000	DFFFFF	47	5E0000	5FFFFF
110	DC0000	DDFFFF	46	5C0000	5DFFFF
109	DA0000	DBFFFF	45	5A0000	5BFFFF
108	D80000	D9FFFF	44	580000	59FFFF
107	D60000	D7FFFF	43	560000	57FFFF
106	D40000	D5FFFF	42	540000	55FFFF
105	D20000	D3FFFF	41	520000	53FFFF
104	D00000	D1FFFF	40	500000	51FFFF
103	CE0000	CFFFFF	39	4E0000	4FFFFF
102	CC0000	CDFFFF	38	4C0000	4DFFFF

Table 6. Memory organization (Continued)

Sector	Address range		Sector	Address range	
101	CA0000	CBFFFF	37	4A0000	4BFFFF
100	C80000	C9FFFF	36	480000	49FFFF
99	C60000	C7FFFF	35	460000	47FFFF
98	C40000	C5FFFF	34	440000	45FFFF
97	C20000	C3FFFF	33	420000	43FFFF
96	C00000	C1FFFF	32	400000	41FFFF
95	BE0000	BFFFFF	31	3E0000	3FFFFF
94	BC0000	BDFFFF	30	3C0000	3DFFFF
93	BA0000	BBFFFF	29	3A0000	3BFFFF
92	B80000	B9FFFF	28	380000	39FFFF
91	B60000	B7FFFF	27	360000	37FFFF
90	B40000	B5FFFF	26	340000	35FFFF
89	B20000	B3FFFF	25	320000	33FFFF
88	B00000	B1FFFF	24	300000	31FFFF
87	AE0000	AFFFFF	23	2E0000	2FFFFF
86	AC0000	ADFFFF	22	2C0000	2DFFFF
85	AA0000	ABFFFF	21	2A0000	2BFFFF
84	A80000	A9FFFF	20	280000	29FFFF
83	A60000	A7FFFF	19	260000	27FFFF
82	A40000	A5FFFF	18	240000	25FFFF
81	A20000	A3FFFF	17	220000	23FFFF
80	A00000	A1FFFF	16	200000	21FFFF
79	9E0000	9FFFFF	15	1E0000	1FFFFF
78	9C0000	9DFFFF	14	1C0000	1DFFFF
77	9A0000	9BFFFF	13	1A0000	1BFFFF
76	980000	99FFFF	12	180000	19FFFF
75	960000	97FFFF	11	160000	17FFFF
74	940000	95FFFF	10	140000	15FFFF
73	920000	93FFFF	9	120000	13FFFF
72	900000	91FFFF	8	100000	11FFFF
71	8E0000	8FFFFF	7	0E0000	0FFFFF
70	8C0000	8DFFFF	6	0C0000	0DFFFF
69	8A0000	8BFFFF	5	0A0000	0BFFFF

Table 6. Memory organization (Continued)

Sector	Address range		Sector	Address range	
68	880000	89FFFF	4	080000	09FFFF
67	860000	87FFFF	3	060000	07FFFF
66	840000	85FFFF	2	040000	05FFFF
65	820000	83FFFF	1	020000	03FFFF
64	800000	81FFFF	0	000000	01FFFF

12.4 SPI Instruction

Serial data input D is sampled on the first rising edge of Serial Clock (C) after Chip Select (S#) is driven Low. Then, the one-byte instruction code must be shifted in to the device, most significant bit first, on serial data input DQ0, each bit being latched on the rising edges of Serial Clock (C).

The instruction set is listed in Table 20 on page 44.

Every instruction sequence starts with a one-byte instruction code. Depending on the instruction, this might be followed by address bytes, or by data bytes, or by both or none.

In the case of a read data bytes (READ), read data bytes at higher speed (FAST_READ), read status register (RDSR) or read identification (RDID) instruction, the shifted-in instruction sequence is followed by a data-out sequence. Chip Select (S#) can be driven High after any bit of the data-out sequence is being shifted out.

In the case of a page program (PP), sector erase (SE), write status register (WRSR), write enable (WREN), or write disable (WRDI), Chip Select (S#) must be driven High exactly at a byte boundary, otherwise the instruction is rejected, and is not executed. That is, Chip Select (S#) must driven High when the number of clock pulses after Chip Select (S#) being driven Low is an exact multiple of eight.

All attempts to access the memory array during a write status register cycle, program cycle erase cycle are ignored, and the internal write status register cycle, program cycle, erase cycle continues unaffected.

Note: Output Hi-Z is defined as the point where data out is no longer driven

Table 20: Instruction set

Instruction	Description	One-byte instruction code		Address bytes	Dummy bytes	Data bytes
WREN	Write enable	0000 0110	06h	0	0	0
WRDI	Write disable	0000 0100	04h	0	0	0
RDID	Read identification	1001 1111	9Fh	0	0	1 to 3
RDSR	Read status register	0000 0101	05h	0	0	1 to ∞
WRSR	Write status register	0000 0001	01h	0	0	1
READ	Read data bytes	0000 0011	03h	3	0	1 to ∞
FAST_READ	Read data bytes at higher speed	0000 1011	0Bh	3	1	1 to ∞
PP	Page program (Legacy Program)	0000 0010	02h	3	0	1 to 64
	Page program (Bit-alterable write)	0010 0010	22h	3	0	1 to 64
	Page program (On all 1's)	1101 0001	D1h	3	0	1 to 64
SE	Sector erase	1101 1000	D8h	3	0	0

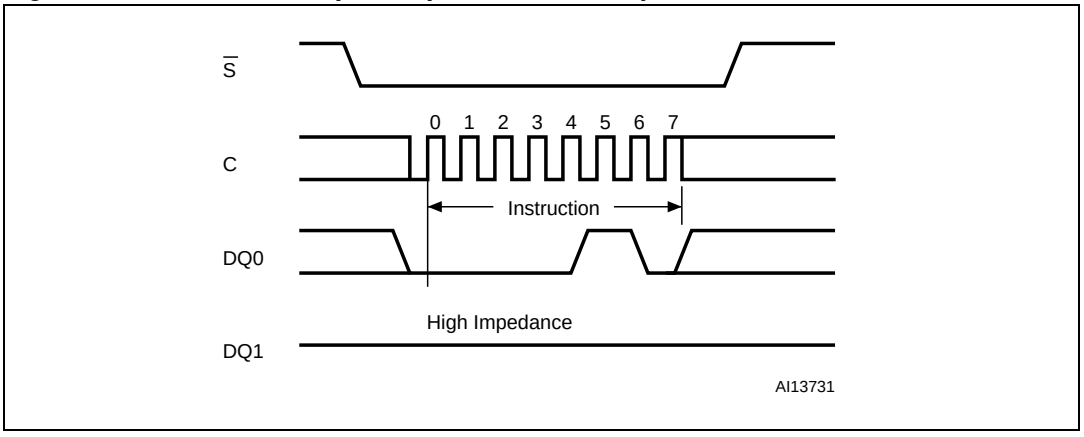
12.4.1 Write enable (WREN)

The write enable (WREN) instruction sets the write enable latch (WEL) bit.

The write enable latch (WEL) bit must be set prior to every page program (PP), sector erase (SE), or write status register (WRSR) instruction.

The write enable (WREN) instruction is entered by driving Chip Select ($\overline{S}$) Low, sending the instruction code, and then driving Chip Select ($\overline{S}$) High.

Figure 9: Write enable (WREN) instruction sequence



12.4.2 Write disable (WRDI)

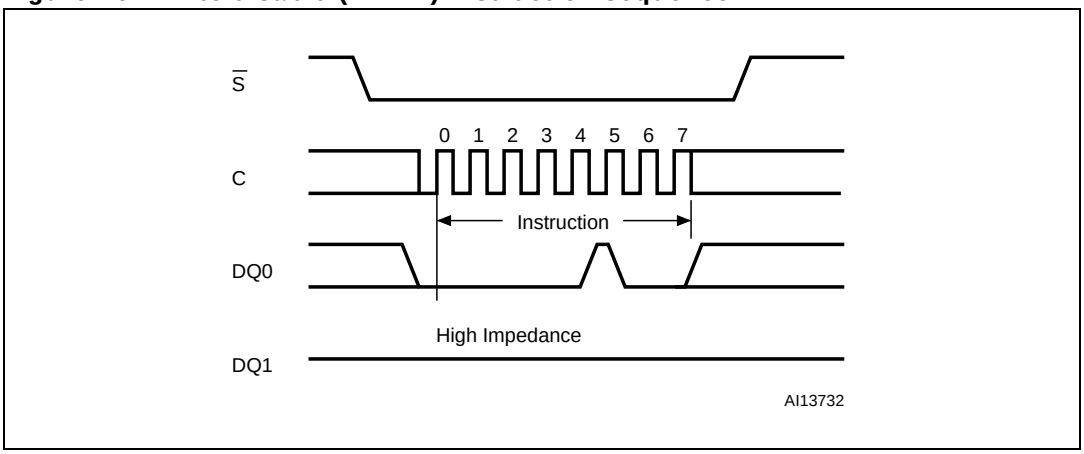
The write disable (WRDI) instruction resets the write enable latch (WEL) bit.

The write disable (WRDI) instruction is entered by driving Chip Select (S#) Low, sending the instruction code, and then driving Chip Select (S#) High.

The write enable latch (WEL) bit is reset under the following conditions:

- Power-up
- Write disable (WRDI) instruction completion
- Write status register (WRSR) instruction completion
- Page program (PP) instruction completion
- Sector erase (SE) instruction completion

Figure 10: Write disable (WRDI) instruction sequence



12.4.3 Read identification (RDID)

The read identification (RDID) instruction allows to read the device identification data:

- Manufacturer identification (1 byte)
- Device identification (2 bytes)

The manufacturer identification is assigned by JEDEC, and has the value 20h for Numonyx.

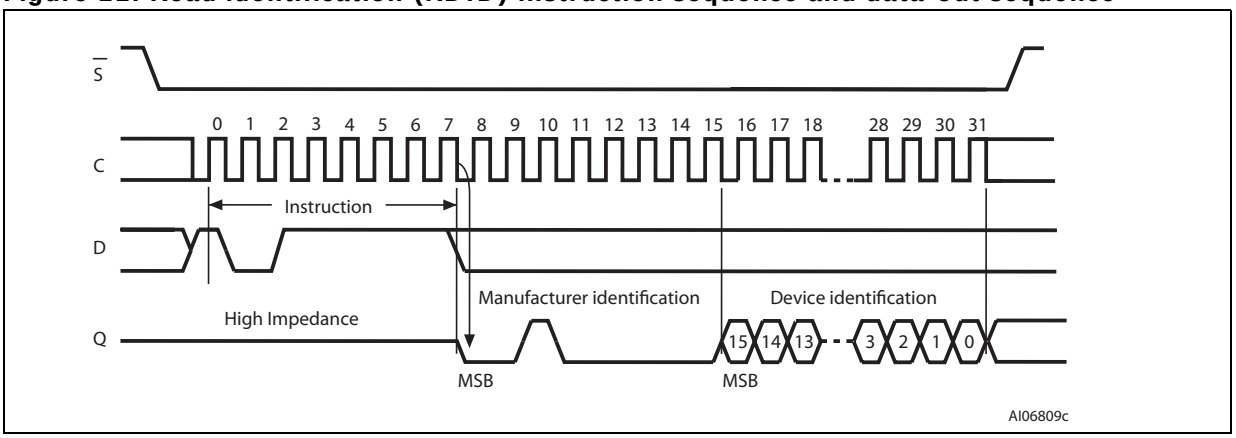
Any read identification (RDID) instruction while an erase or program cycle is in progress, is not decoded, and has no effect on the cycle that is in progress.

The device is first selected by driving Chip Select (S#) Low. Then, the 8-bit instruction code for the instruction is shifted in. After this, the 24-bit device identification stored in the memory will be shifted out on serial data output (DQ1). Each bit is shifted out during the falling edge of Serial Clock (C).

The read identification (RDID) instruction is terminated by driving Chip Select (S#) High at any time during data output.

When Chip Select (S#) is driven High, the device is put in the standby power mode. Once in the standby power mode, the device waits to be selected, so that it can receive, decode and execute instructions.

Figure 11: Read identification (RDID) instruction sequence and data-out sequence

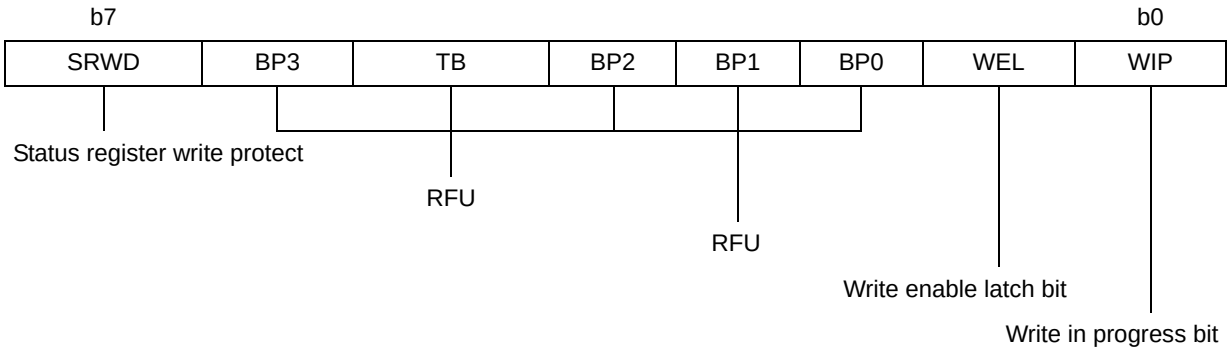


12.4.4 Read status register (RDSR)

The read status register (RDSR) instruction allows the status register to be read. The status register may be read at any time, even while a program, erase, write status register 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 instruction to the device. It is also possible to read the status register continuously, as shown in Figure 12 on page 49

RDSR is the only instruction accepted by the device while a program, erase, write status register operation is in progress.

Table 21: Status register format



The status and control bits of the status register are as follows:

12.4.4.1 WIP bit

The write in progress (WIP) bit indicates whether the memory is busy with a write status register, program, erase cycle. When set to ‘1’, such a cycle is in progress, when reset to ‘0’ no such cycle is in progress.

While WIP is ‘1’, RDSR is the only instruction the device will accept; all other instructions are ignored.

12.4.4.2 WEL bit

The write enable latch (WEL) bit indicates the status of the internal write enable latch. When set to ‘1’ the internal write enable latch is set, when set to ‘0’ the internal write enable latch is reset and no write status register, program, erase instruction is accepted.

12.4.4.3 BP3, BP2, BP1, BP0 bits

The block protect (BP3, BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against program (or write) and erase instructions. These bits are written with the write status register (WRSR) instruction. When one or more of the block protect (BP3, BP2, BP1, BP0) bits is set to ‘1’, the relevant memory area (as defined in Table 1) becomes protected against page program (PP), dual input fast program (DIFP), quad input fast program (QIFP), and sector erase (SE) instructions. The block protect (BP3, BP2, BP1, BP0) bits can be written provided that the hardware protected mode has not been set.The bulk erase (BE) instruction is executed if, and only if, all block protect (BP3, BP2, BP1, BP0) bits are 0.

Table 7. Protected area sizes

Status register contents					Memory content	
TB bit	**BP bit 3**	**BP bit 2**	**BP bit 1**	**BP bit 0**	**Protected area**	**Unprotected area**
0	0	0	0	0	none	All sectors[1] (Sectors 0 to 127)
0	0	0	0	1	Upper 128th (Sector 127)	Sectors 0 to 126
0	0	0	1	0	Upper 64th (Sectors 126 to 127)	Sectors 0 to 125
0	0	0	1	1	Upper 32nd (Sectors 124 to 127)	Sectors 0 to 123
0	0	1	0	0	Upper 16th (Sectors 120 to 127)	Sectors 0 to 119
0	0	1	0	1	Upper 8th (Sectors 112 to 127)	Sectors 0 to 111
0	0	1	1	0	Upper quarter (Sectors 96 to 127)	Sectors 0 to 95
0	0	1	1	1	Upper half (Sectors 64 to 127)	Sectors 0 to 63
0	1	X(2)	X(2)	X(2)	All sectors (Sectors 0 to 127)	None
1	0	0	0	0	none	All sectors(1) (Sectors 0 to 127)
1	0	0	0	1	Lower 128th (Sector 0)	Sectors 1 to 127
1	0	0	1	0	Lower 64th (Sectors 0 to 1)	Sectors 2 to 127
1	0	0	1	1	Lower 32nd (Sectors 0 to 3)	Sectors 4 to 127
1	0	1	0	0	Lower 16th (Sectors 0 to 7)	Sectors 8 to 127
1	0	1	0	1	Lower 8th (Sectors 0 to15)	Sectors 16 to 127
1	0	1	1	0	Lower 4th (Sectors 0 to 31)	Sectors 32 to 127
1	0	1	1	1	Lower half (Sectors 0 to 63)	Sectors 64 to 127
1	1	X(2)	X(2)	X(2)	All sectors (Sectors 0 to 127)	None

1. The device is ready to accept a bulk erase instruction if, and only if, all block protect (BP3, BP2, BP1, BP0) are 0
2. X can be 0 or 1

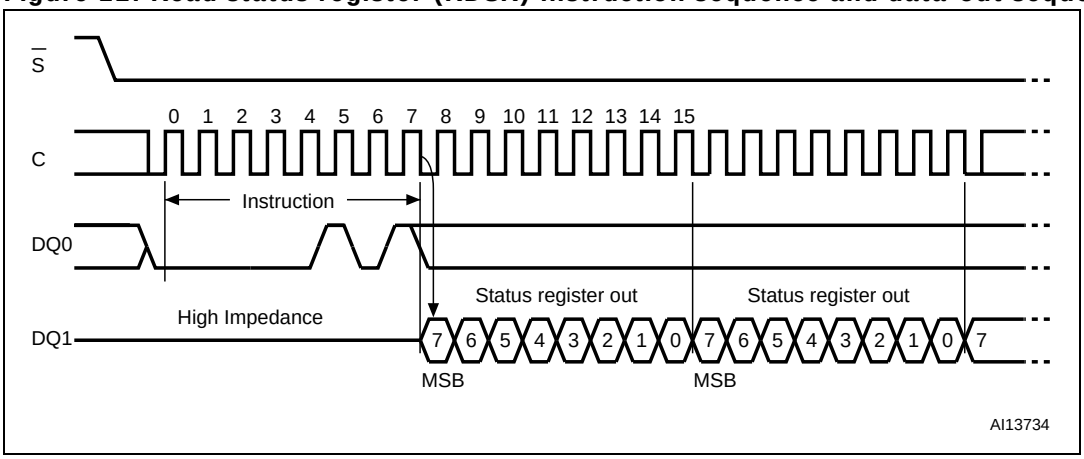
12.4.4.4 Top/bottom bit

Reads as 0

12.4.4.5 SRWD bit

The status register write disable (SRWD) bit is operated in conjunction with the write protect (W) signal. The status register write disable (SRWD) bit and the write protect (W) signal allow the device to be put in the hardware protected mode (when the status register write disable (SRWD) bit is set to '1', and write protect (W) is driven Low). In this mode, the non-volatile bits of the status register (SRWD, TB, BP3, BP2, BP1, BP0) become read-only bits and the write status register (WRSR) instruction is no longer accepted for execution.

Figure 12: Read status register (RDSR) instruction sequence and data-out sequence



12.4.5 Write status register (WRSR)

The write status register (WRSR) instruction allows new values to be written to the status register. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded and executed, the device sets the write enable latch (WEL).

The write status register (WRSR) instruction is entered by driving Chip Select (S#) Low, followed by the instruction code and the data byte on serial data input (DQ0).

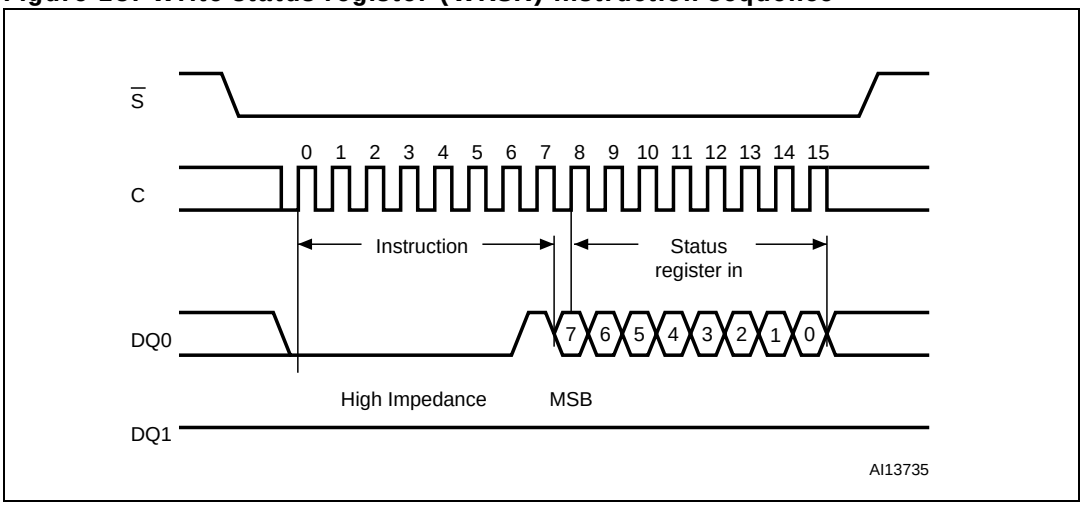
The write status register (WRSR) instruction has no effect on b1 and b0 of the status register.

Chip Select (S#) must be driven High after the eighth bit of the data byte has been latched in. If not, the write status register (WRSR) instruction is not executed. As soon as Chip Select (S#) is driven High, the self-timed write status register cycle (whose duration is t_W) is initiated. 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 write in progress (WIP) bit is 1 during the self-timed write status register cycle, and is 0 when it is completed. When the cycle is completed, the write enable latch (WEL) is reset.

The write status register (WRSR) instruction allows the user to change the values of the block protect (BP3, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only. The write status register (WRSR) instruction also allows the user to set and reset the status register write disable (SRWD) bit in accordance with the Write Protect (W) signal. The status register write disable (SRWD) bit and Write Protect (W) signal allow the device to be put in the hardware protected mode (HPM). The write status register (WRSR) instruction is not executed once the hardware protected mode (HPM) is entered.

Read Status Register (RDSR) is the only instruction accepted while WRSR operation is in progress; all other instructions are ignored.

Figure 13: Write status register (WRSR) instruction sequence



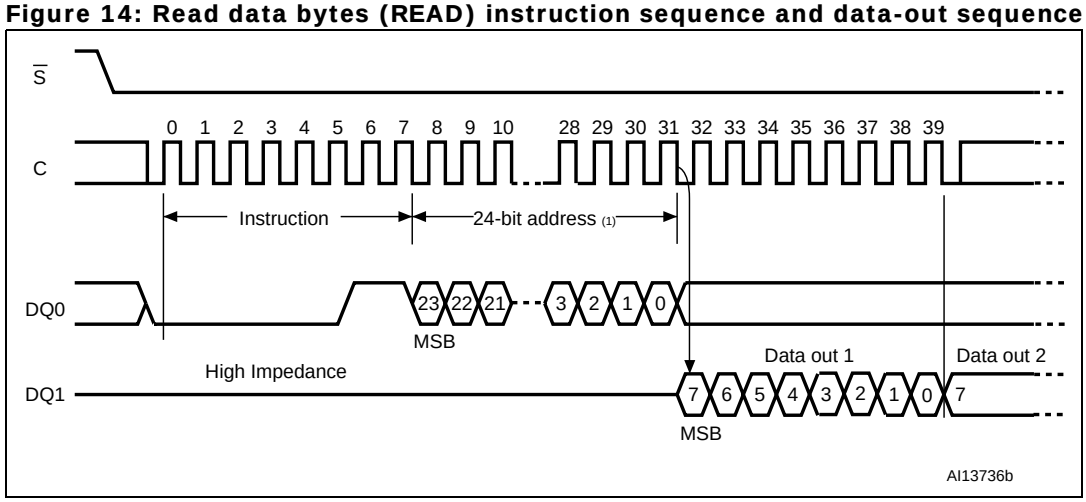
12.4.6 Read data bytes (READ)

The device is first selected by driving Chip Select (S#) Low. The instruction code for the read data bytes (READ) instruction is followed by a 3-byte address A[23:0], each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, is shifted out on serial data output (Q), each bit being shifted out, at a maximum frequency f_R, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single read data bytes (READ) instruction. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

The read data bytes (READ) instruction is terminated by driving Chip Select (S#) High. Chip Select (S#) can be driven High at any time during data output. Any read data bytes (READ) instruction, while an erase, program, write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

Figure 14: Read data bytes (READ) instruction sequence and data-out sequence



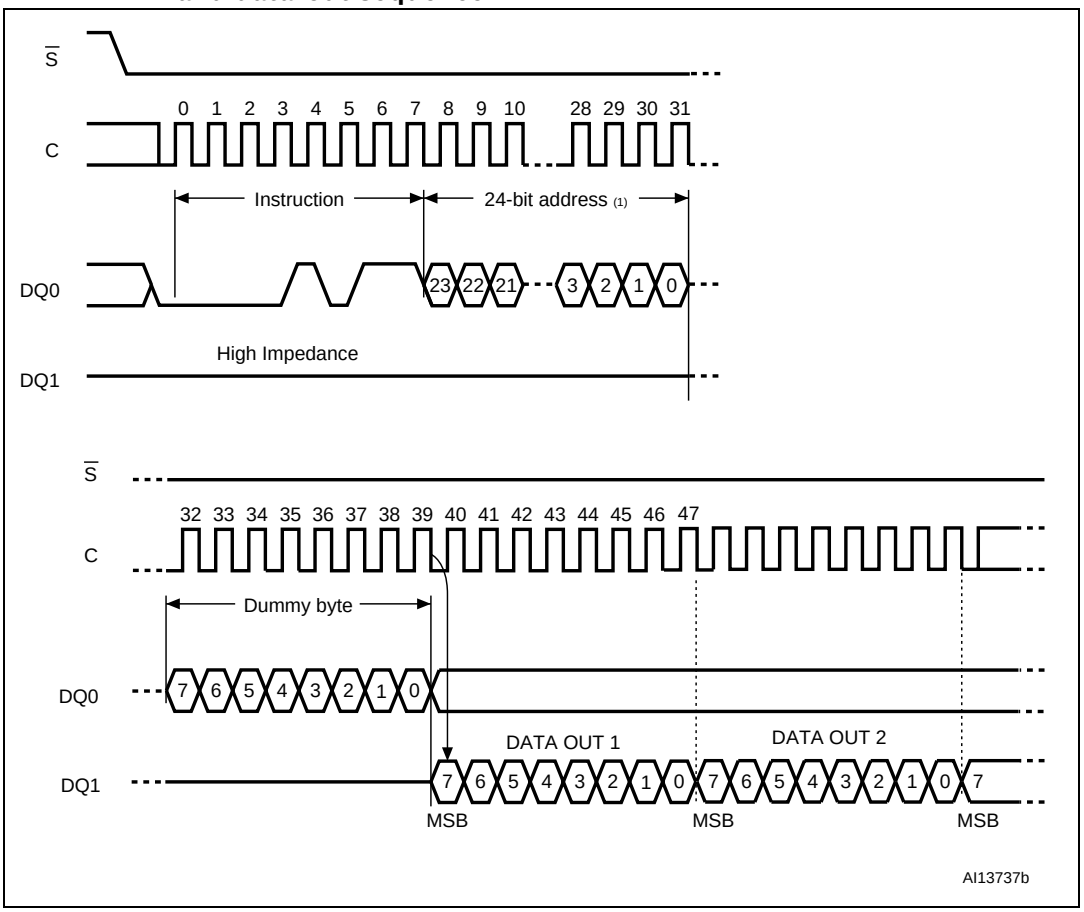
12.4.7 Read data bytes at higher speed (FAST_READ)

The device is first selected by driving Chip Select (S#) Low. The instruction code for the read data bytes at higher speed (FAST_READ) instruction is followed by a 3-byte address A[23:0] and a dummy byte, each bit being latched-in during the rising edge of Serial Clock (C). Then the memory contents, at that address, are shifted out on serial data output (Q) at a maximum frequency f_C, during the falling edge of Serial Clock (C).

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single read data bytes at higher speed (FAST_READ) instruction. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

The read data bytes at higher speed (FAST_READ) instruction is terminated by driving Chip Select (S#) High. Chip Select (S#) can be driven High at any time during data output. Any read data bytes at higher speed (FAST_READ) instruction, while an erase, program, write, or cycle is in progress, is rejected without having any effects on the cycle that is in progress.

Figure 15: Read data bytes at higher speed (FAST_READ) instruction sequence and data-out sequence



12.4.8 Page program (PP)

Note: This definition applies to all flavors of Page Program: Legacy Program, Bit-alterable.

The page program (PP) instruction allows bytes to be programmed/written in the memory. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded, the device sets the write enable latch (WEL).

The page program (PP) instruction is entered by driving Chip Select (S#) Low, followed by the instruction code, three address bytes and at least one data byte on serial data input (DQ0). If the 6 least significant address bits (A5-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same page (from the address whose 6 least significant bits (A5-A0) are all zero). Chip Select (S#) must be driven Low for the entire duration of the sequence.

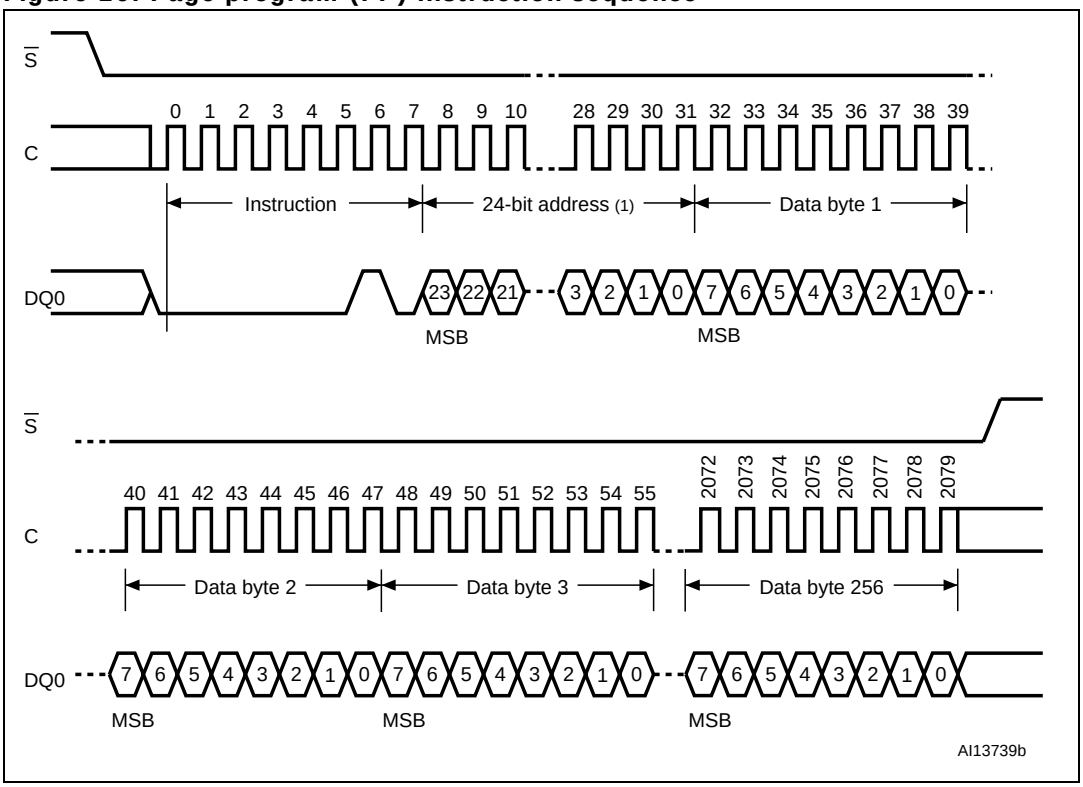
If more than 64 bytes are sent to the device, previously latched data are discarded and the last 64 data bytes are guaranteed to be programmed/written correctly within the same page. If less than 64 data bytes are sent to device, they are correctly programmed/written at the requested addresses without having any effects on the other bytes of the same page. (With Program on all 1s, the entire page should already have been set to all 1s (FFh).)

For optimized timings, it is recommended to use the page program (PP) instruction to program all consecutive targeted bytes in a single sequence versus using several page program (PP) sequences with each containing only a few bytes.

Chip Select (S#) must be driven High after the eighth bit of the last data byte has been latched in, otherwise the page program (PP) instruction is not executed.

As soon as Chip Select (S#) is driven High, the self-timed page program cycle (whose duration is t_{PP}) is initiated. While the page program cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The write in progress (WIP) bit is 1 during the self-timed page program cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the write enable latch (WEL) bit is reset. RDSR is the only instruction accepted while a Page Program operation is in progress; all other instructions are ignored.

Figure 16: Page program (PP) instruction sequence



12.4.9 Sector erase (SE)

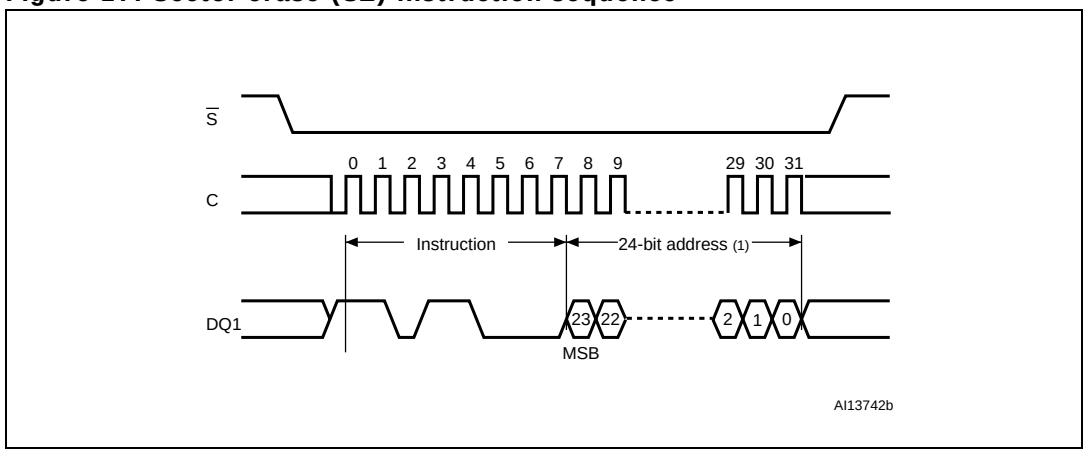
The sector erase (SE) instruction sets to '1' (FFh) all bits inside the chosen sector. Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded, the device sets the write enable latch (WEL).

The sector erase (SE) instruction is entered by driving Chip Select (S#) Low, followed by the instruction code, and three address bytes on serial data input (DQ0). Any address inside the sector is a valid address for the sector erase (SE) instruction. Chip Select (S#) must be driven Low for the entire duration of the sequence.

Chip Select (S#) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the sector erase (SE) instruction is not executed. As soon as Chip Select (S#) is driven High, the self-timed sector erase cycle (whose duration is t_{SE}) is initiated. While the sector erase cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The write in progress (WIP) bit is 1 during the self-timed sector erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the write enable latch (WEL) bit is reset. RDSR is the only instruction accepted while device is busy with erase operation; all other instructions are ignored.

A sector erase (SE) instruction applied to a page which is protected by the block protect (BP3, BP2, BP1, BP0) bits is not executed.

Figure 17: Sector erase (SE) instruction sequence



13.0 Power and Reset Specification

13.1 Power-Up and Power-Down

Upon power-up the flash memory interface is defined by the SERIAL pin being at Vss (parallel) or Vcc (serial).

- During power-up if the SERIAL pin is at Vss the flash memory will be a x16 parallel interface.
- During power-up if the SERIAL pin is at Vcc the flash memory will be a SPI interface.

After the interface is defined it can not be changed until a full power-down is completed and a power-up sequence is reinitiated.

Power supply sequencing is not required if VPP is connected to VCC or VCCQ. Otherwise V_{CC} and V_{CCQ} should attain their minimum operating voltage before applying V_{PP}.

Power supply transitions should only occur when RST# is low. This protects the device from accidental programming or erasure during power transitions.

13.2 Reset Specifications

Asserting RST# during a system reset is important with automated program/erase devices because systems typically expect to read from flash memory when coming out of reset. If a CPU reset occurs without a flash memory reset, proper CPU initialization may not occur. This is because the flash memory may be providing status information, instead of array data as expected. Connect RST# to the same active low reset signal used for CPU initialization.

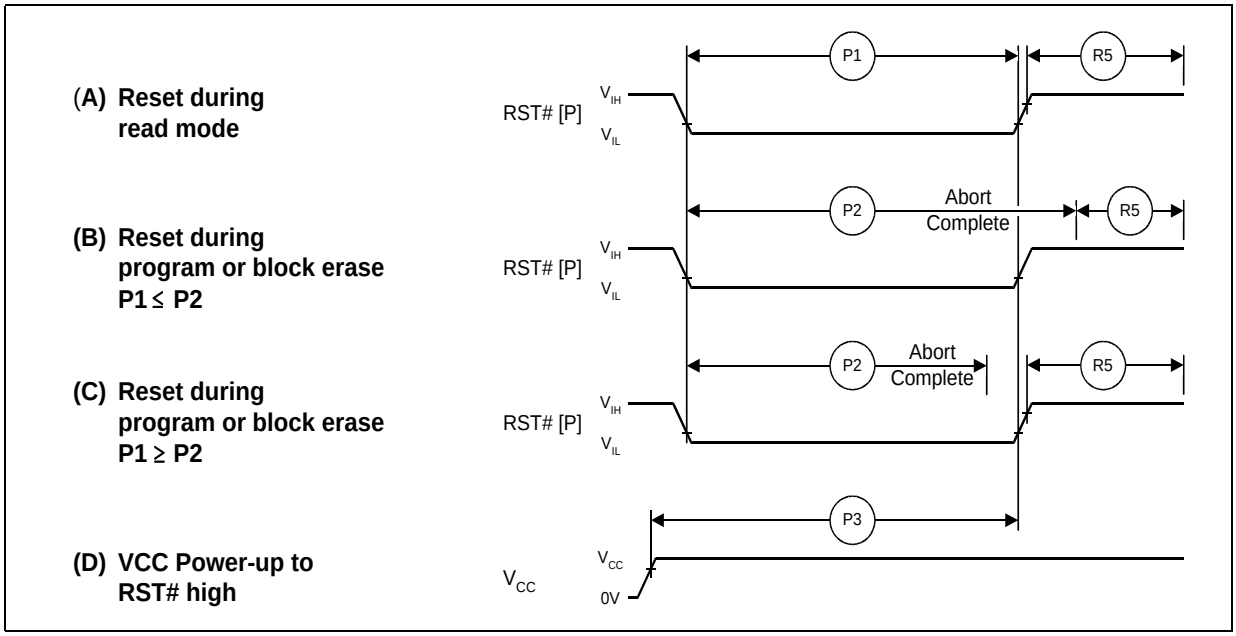
Also, because the device is disabled when RST# is asserted, it ignores its control inputs during power-up/down. Invalid bus conditions are masked, providing a level of memory protection.

Table 22: Power and Reset

Num	Symbol	Parameter(1)	Min	Max	Unit	Notes
P1	t_{PLPH}	RST# pulse width low	100	-	ns	1,2,3,4
P2	t_{PLRH}	RST# low to device reset during erase	-	40	us	1,3,4,7
		RST# low to device reset during program	-	40		1,3,4,7
P3	t_{VCCPH}	V_{CC} Power valid to RST# de-assertion (high)	100	-		1,4,5,6

Notes:
1. These specifications are valid for all device versions (packages and speeds).
2. The device may reset if t_{PLPH} is < t_{PLPH} MIN, but this is not guaranteed.
3. Not applicable if RST# is tied to Vcc.
4. Sampled, but not 100% tested.
5. When RST# is tied to the V_{CC} supply, device will not be ready until t_{VCCPH} after $V_{CC} \geq V_{CCMIN}$.
6. When RST# is tied to the V_{CCQ} supply, device will not be ready until t_{VCCPH} after $V_{CC} \geq V_{CCMIN}$.
7. Reset completes within t_{PLPH} if RST# is asserted while no erase or program operation is executing.

Figure 18: Reset Operation Waveforms



13.3 Power Supply Decoupling

Flash memory devices require careful power supply de-coupling. Three basic power supply current considerations are 1) standby current levels, 2) active current levels, and 3) transient peaks produced when CE# and OE# are asserted and deasserted.

When the device is accessed, many internal conditions change. Circuits within the device enable charge-pumps, and internal logic states change at high speed. All of these internal activities produce transient signals. Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and correct de-coupling capacitor selection suppress transient voltage peaks.

Flash memory devices draw their power from VCC, VPP, and VCCQ, each power connection should have a 0.1 µF ceramic capacitor to ground. High-frequency, inherently low-inductance capacitors should be placed as close as possible to package leads.

Additionally, for every eight devices used in the system, a 4.7 µF electrolytic capacitor should be placed between power and ground close to the devices. The bulk capacitor is meant to overcome voltage droop caused by PCB trace inductance.

14.0 Max Ratings and Operating Conditions

14.1 Absolute Maximum Ratings

Warning: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only.

Table 23: Absolute Maximum Ratings

Parameter	Maximum Rating
Voltage on any signal (except V_{CC}, Vccq, V_{PP})(1)	–2.0 V to +5.6V, <20ns
V_{PP} voltage(2,4)	–2.0 V to +5.6V, <20ns
V_{CC} voltage(2,4)	–2.0 V to +5.6V, <20ns
V_{CCQ} voltage(2,4,5)	–2.0 V to +5.6V, <20ns
Output short circuit current(3)	100 mA

Notes:
1. All specified voltages are with respect to Vss. During infrequent non-periodic transitions, the voltage potential between Vss and input/output pins may undershoot to -2.0v for periods <20 ns or overshoot to VCCQ + 2.0v for periods <20 ns.
2. During infrequent non-periodic transitions the voltage potential between Vss and the supplies may undershoot to -2.0v for periods <20 ns or overshoot to supply voltage (max) + 2.0v for periods <20 ns.
3. Output shorted for no more than one second. No more than one output shorted at a time.
4. For functional operating voltages, please refer to Section 27, "DC Voltage Characteristics" on page 61
5. Make sure that V_{CCQ} is less or equal to V_{CC} in value, otherwise the device fails to operate correctly" in the next revision of the datasheet.

14.2 Operating Conditions

Note: Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

Table 24: Operating Conditions

Symbol	Parameter		Min	Max	Units	Notes
T_C	Operating Temperature (115 ns)		0	+70	°C	1
T_C	Operating Temperature (135 ns)		-30	+85	°C	
V_{CC}	V_{CC} Supply Voltage		2.7	3.6	V	
V_{CCQ}	I/O Supply Voltage	CMOS inputs	1.7	3.6		2
		TTL inputs	2.4	3.6		
V_{PP}	V_{PP} Voltage Supply (Logic Level)		0.9	3.6		3

Notes:
1. T_C = Case Temperature.
2. V_{CCQ} = 1.7v - 3.6v range is intended for CMOS inputs and the 2.4v - 3.6v is intended for TTL inputs.
3. In typical operation VPP program voltage is V_{PPL}.

14.3 Endurance

Numonyx® Omneo™ P8P PCM endurance is different than traditional non-volatile memory. For PCM a “write cycle” is defined as any time a bit changes within a 32-byte page.

Table 25: Endurance

Parameter	Condition	Min	Units	Notes
Write Cycle	Main Block (V_{PP} = V_{PPH})	1,000,000	Cycles per 32-Byte Page	1
	Parameter Block (V_{PP} = V_{PPH})	1,000,000		

Notes:
1. In typical operation VPP program voltage is V_{PPL}.

15.0 Electrical Specifications

15.1 DC Current Characteristics

Table 26: DC Current Characteristics

Sym	Parameter[1]		Note	CMOS Inputs V_{CCQ} 1.7v - 3.6v		TTL Inputs V_{CCQ} 2.4v - 3.6v		Unit	Test Condition	
				Typ	Max	Typ	Max			
I_{LI}	Input Load		9		±1		±2	µA	$V_{CC} = V_{CCMAX}$ $V_{CCQ} = V_{CCQMAX}$ $V_{IN} = V_{CCQ}$ or GND	
I_{LO}	Output Leakage	DQ_{15-0}			±1		±10	µA	$V_{CC} = V_{CCMAX}$ $V_{CCQ} = V_{CCQMAX}$ $V_{IN} = V_{CCQ}$ or GND	
I_{CCS} I_{CCD}	V_{CC} Standby, Power Down	128-Mbit	11	80	160	80	160	µA	$V_{CC} = V_{CCMAX}$, $V_{CCQ} = V_{CCQMAX}$ CE# = V_{CCQ}, RST# = V_{CCQ} WP# = V_{IH} Must reach stated $I_{CCS} \leq 5$µS after CE# = V_{IH}	
I_{CCR}	Average V_{CC} Read	Asynchronous single word f = 5MHz (1 CLK)		30	42	30	42	mA	Internal 8 Word Read	$V_{CC} = V_{CCMAX}$ CE# = V_{IL} OE# = V_{IH} Inputs: V_{IH} or V_{IL}
		Page Mode f = 13 MHz (9 CLK)		15	20	15	20	mA	8 Word Read	
I_{CCW}, I_{CCE}	V_{CC} Write, V_{CC} Erase		3,4,5, 12	35	50	36	51	mA	program/erase in progress	
I_{CCWS} I_{CCES}	V_{CC} Write Suspend V_{CC} Erase Suspend		6	Refer to I_{CCS} for each density above.				µA	CE# = V_{CCQ}, suspend in progress	
I_{PPS} I_{PPWS} IPPES	V_{PP} Standby V_{PP} Write Suspend V_{PP} Erase Suspend		3	0.2	5	0.2	5	µA	$V_{PP} = V_{PPL}$, suspend in progress	
I_{PPR}	V_{PP} Read			2	15	2	15	µA	$V_{PP} \leq V_{CC}$	
I_{PPW}	V_{PP} Write		3	0.05	0.10	0.05	0.10	mA	write in progress	
I_{PPE}	V_{PP} Erase		3	0.05	0.10	0.05	0.10	mA	erase in progress	

Note: Refer Table 27 on page 61 for the Notes relevant to this table.

15.2 DC Voltage Characteristics

Table 27: DC Voltage Characteristics

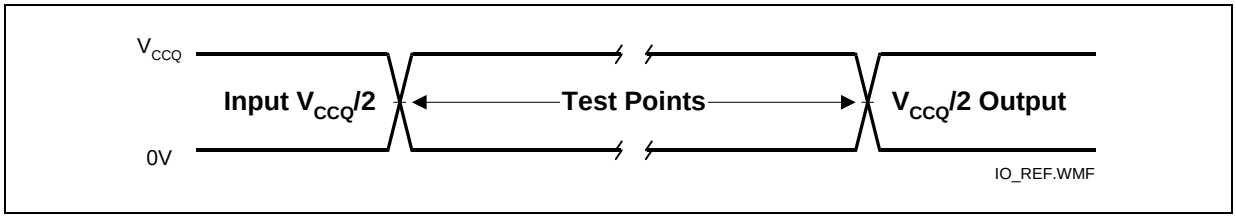
Sym	Parameter	Notes	CMOS Inputs V_{CCQ} 1.7v - 3.6v		TTL Inputs V_{CCQ} 2.4v - 3.6v		Unit	Test Condition
			Min	Max	Min	Max		
V_{IL}	Input Low	2	0	0.4	0	0.6	V	
V_{IH}	Input High	2	V_{CCQ} – 0.4	V_{CCQ}	2.0	V_{CCQ}	V	
V_{OL}	Output Low			0.1		0.1	V	$V_{CC} = V_{CCMIN}$ $V_{CCQ} = V_{CCQMIN}$ I_{OL} = 100 µA
V_{OH}	Output High		V_{CCQ} – 0.1		V_{CCQ} – 0.1		V	$V_{CC} = V_{CCMIN}$ $V_{CCQ} = V_{CCQMIN}$ I_{OH} = –100 µA
V_{PPLK}	V_{PP} Lock-Out	1		0.4		0.4	V	
V_{LKO}	V_{CC} Lock		1.5		1.5		V	
V_{LKOQ}	V_{CCQ} Lock		0.9		0.9		V	

Notes:
1. $V_{PP} <= V_{PPLK}$ inhibits erase and write operations. Don't use V_{PP} outside the valid range.
2. V_{IL} can undershoot to –1.0V for durations of 2 ns or less and V_{IH} can overshoot to $V_{CCQ(MAX)}$+1.0V for durations of 2 ns or less.

16.0 AC Characteristics

16.1 AC Test Conditions

Figure 19: AC Input/Output Reference Waveform



Note: AC test inputs are driven at V_{CCQ} for Logic "1" and 0.0 V for Logic "0." Input/output timing begins/ends at $V_{CCQ}/2$. Input rise and fall times (10% to 90%) < 5 ns. Worst case speed occurs at V_{CC} = V_{CC}Min.

Figure 20: Transient Equivalent Testing Load Circuit

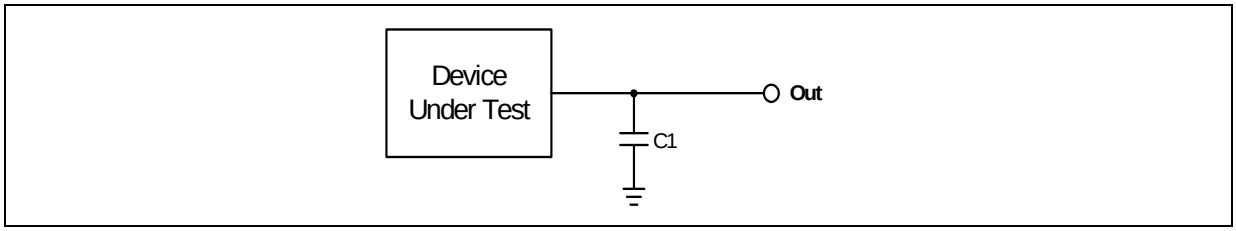


Table 28: Test configuration component value for worst case speed conditions

Test Configuration	C_L (pF) (includes jig capacitance)
V_{CCQMIN}	30

16.2 Capacitance

Table 29: Capacitance: T_A = +25°C, f = 1 MHz (Sampled, not 100% tested)

Symbol	Parameter[1]	Min	Typ	Max	Unit	Condition
C_{IN}	Input Capacitance	2	6	8	pF	V_{IN} = 0.0 V
C_{OUT}	Output Capacitance	2	4	7	pF	V_{OUT} = 0.0 V

16.3 AC Read Specifications

Table 30: AC Read Specifications

Num	Sym	Parameter[1]	Temp	0 to +70 C		-30 to +85 C		Units
			Note	Min	Max	Min	Max	
Asynchronous Specifications								
R1	t_{AVAV}	Read cycle time	1,4	115		135		ns
R2	t_{AVQV}	Address to output valid	1,4		115		135	ns
R3	t_{ELQV}	CE# low to output valid	1,4		115		135	ns
R4	t_{GLQV}	OE# low to output valid	1,2,4		25		25	ns
R5	t_{PHQV}	RST# high to output valid	1,4		150		150	ns
R6	t_{ELQX}	CE# low to output in low-Z	3,4	0		0		ns
R7	t_{GLQX}	OE# low to output in low-Z	1,2,3,4	0		0		ns
R8	t_{EHQZ}	CE# high to output in high-Z	1,3,4		24		24	ns
R9	t_{GHQZ}	OE# high to output in high-Z	1,3,4		24		24	ns
R10	t_{OH}	Output hold from first occurring address, CE#, or OE# change	1,3,4	0		0		ns
R11	t_{EHEL}	CE# pulse width high	1,4	20		20		ns
R108	t_{APA}	Page address access			25		25	ns

Notes:
1. See Figure 19 on page 62 for timing measurements and maximum allowable input slew rate.
2. OE# may be delayed by up to $t_{ELQV} - t_{GLQV}$ after CE#'s falling edge without impact to t_{ELQV}.
3. Sampled, not 100% tested.
4. All specs above apply to all densities.

Figure 21: Asynchronous Single-Word Read

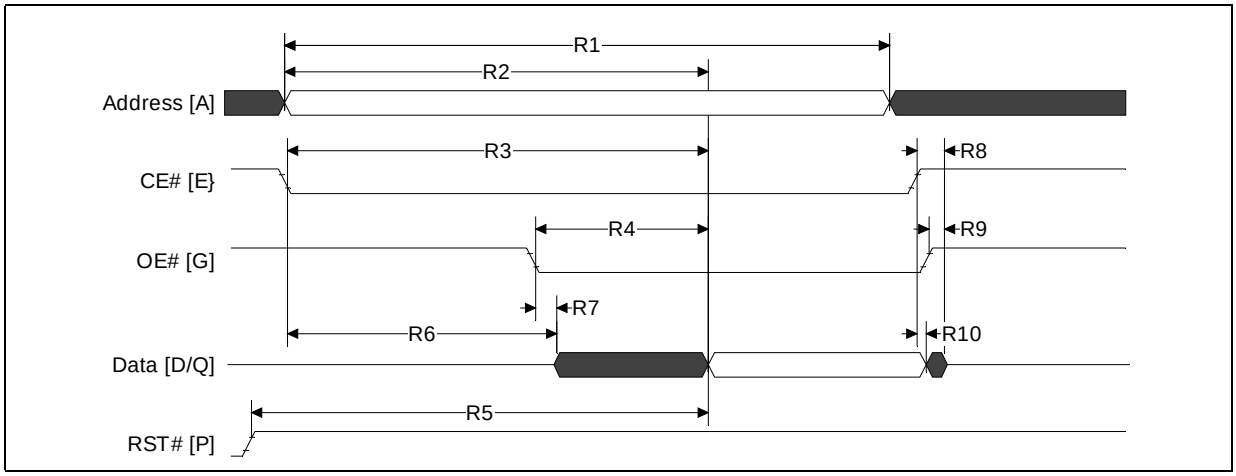
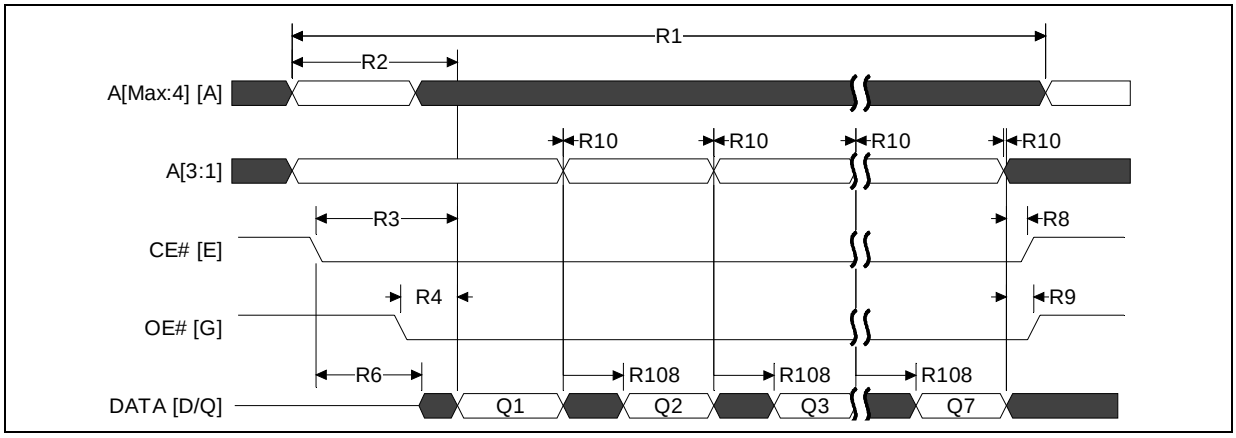


Figure 22: Asynchronous Page Mode Read Timing



16.4 AC Write Specifications

Table 31: AC Write Characteristics

Num	Sym	Parameter (1, 2)	Speed	All Speeds		Units
			Note	Min	Max	
W1	t_{PHWL}	RST# high recovery to WE# low	3	150		ns
W2	t_{ELWL}	CE# setup to WE# low	10	0		ns
W3	t_{WLWH}	WE# write pulse width low	4	50		ns
W4	t_{DVWH}	Data setup to WE# high		50		ns
W5	t_{AVWH}	Address valid setup to WE# high		50		ns
W6	t_{WHEH}	CE# hold from WE# high	10	0		ns
W7	t_{WHDX}	Data hold from WE# high		0		ns
W8	t_{WHAX}	Address hold from WE# high		0		ns
W9	t_{WHWL}	WE# pulse width high		20		ns
W10	t_{VPWH}	V_{PP} setup to WE# high	3,6	200		ns
W11	t_{QVVL}	V_{PP} hold from valid Status read	3,6	0		ns
W12	t_{QVBL}	WP# hold from valid Status read	3,6	0		ns
W13	t_{BHWH}	WP# setup to WE# high	3,6	200		ns
W14	t_{WHGL}	WE# high to OE# low	8	0		ns
W16	t_{WHQV}	WE# high to read valid	3, 5, 9	t_{AVQV}+35		ns
Write to Asynchronous Read Specifications						
W18	t_{WHAV}	WE# high to Address valid	3, 5, 7	0		ns

Notes:
1. Write timing characteristics during erase suspend are the same as write-only operations.
2. CE#- or WE#-high terminates a write operation.
3. Sampled, not 100% tested.
4. Write pulse width low (t_{WLWH} or t_{ELEH}) is defined from CE# or WE# low (whichever occurs last) to CE# or WE# high (whichever occurs first). Hence, $t_{WLWH} = t_{ELEH} = t_{WLEH} = t_{ELWH}$.
5. Write pulse width high (t_{WHWL} or t_{EHEL}) is defined from CE# or WE# high (whichever is first) to CE# or WE# low (whichever is last). Hence, $t_{WHWL} = t_{EHEL} = t_{WHEL} = t_{EHWL}$.
6. V_{PP} and WP# should be at a valid level without changing state until erase or program success is determined.
7. This spec is only applicable when transitioning from a write cycle to an asynchronous read.
8. When doing a read status operation following any command that alters the Status Register contents, W14 is 20ns.
9. Add 10ns if the write operation results in a block lock status change, for subsequent read operations to reflect this change.
10. Guaranteed by design.

Figure 23: Single-Word Write Timing

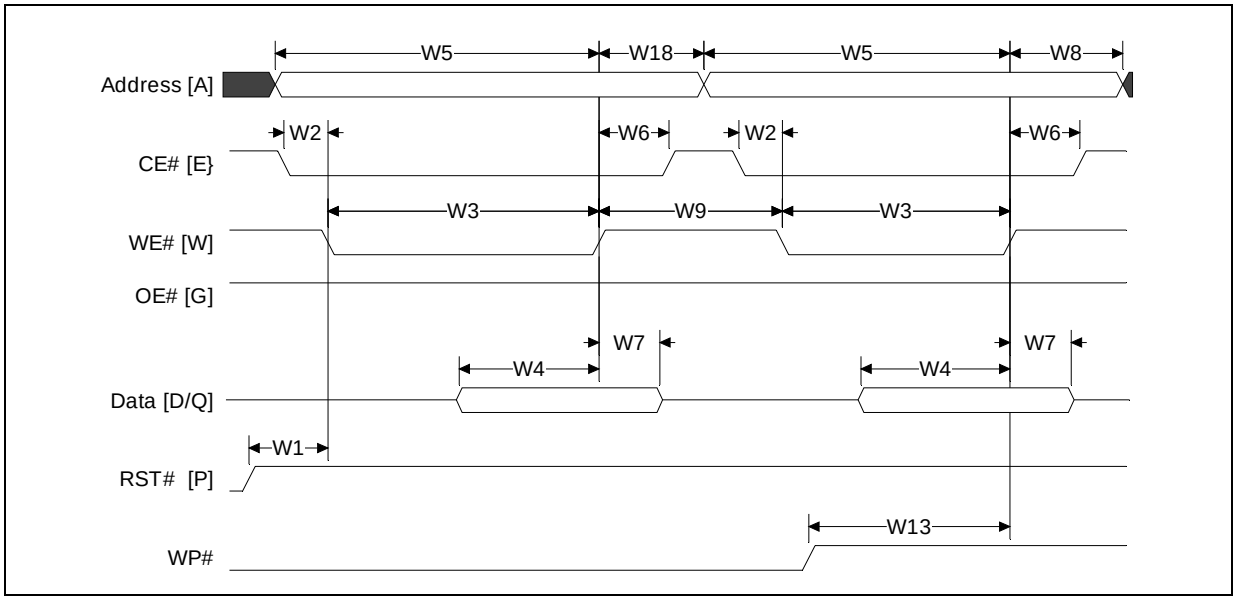
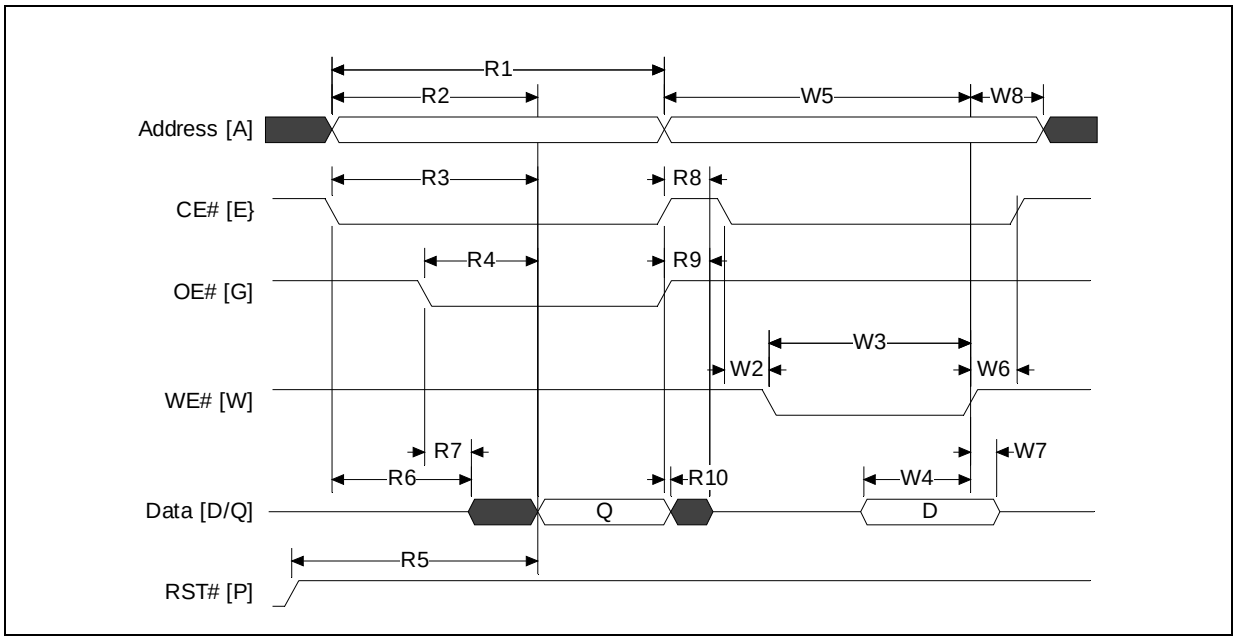
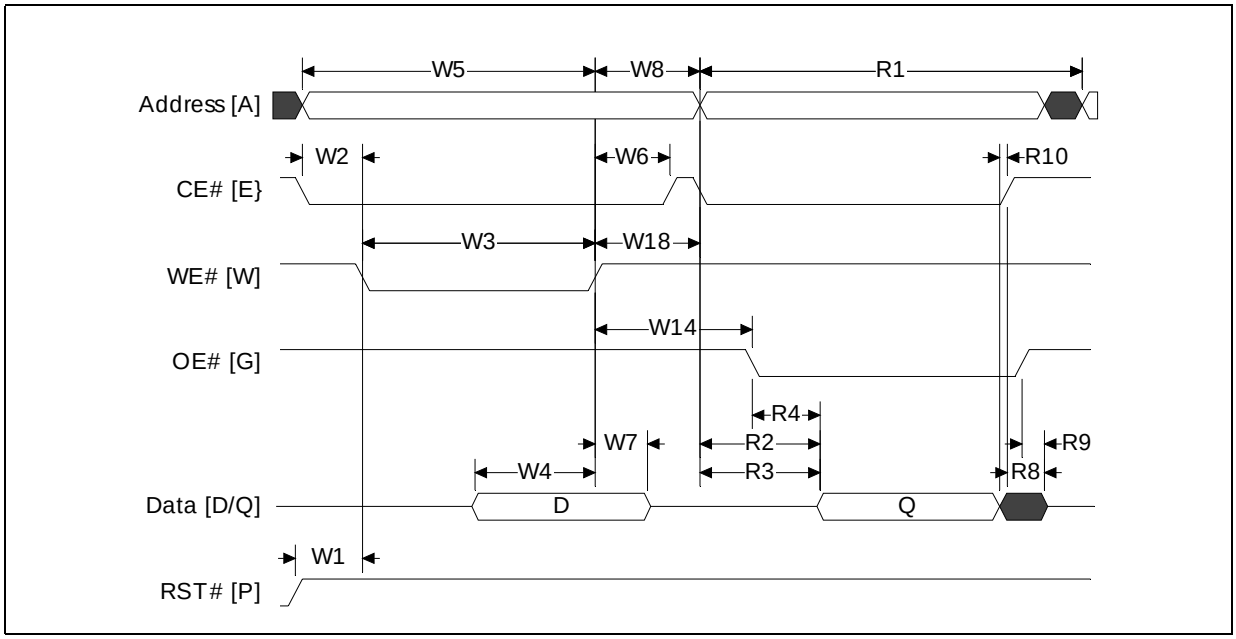


Figure 24: Asynchronous Read-to-Write Timing



Note: See sections 7.6 (AC Read Characteristics) and 7.7 (AC Write Characteristics) for the values of Rs and Ws.

Figure 25: Write-to-Asynchronous Read Timing



See sections 7.6 (AC Read Characteristics) and 7.7 (AC Write Characteristics) for the values of Rs and Ws.

16.5 SPI AC Specifications

Sym	Parameter	Speed	-All Speeds		Units
		Note	Min	Max	
F_C	Clock Frequency for all instructions except READ (0 to +70 C)		DC	50	MHz
F_C	Clock Frequency for all instructions except READ (-30 to +85 C)		DC	33	MHz
F_R	Clock Frequency for READ		DC	25	MHz
T_{CH}	Clock High Time	1	9		ns
T_{CL}	Clock Low Time	1	9		ns
T_{CLCH}	Clock Rise Time (peak to peak)	2, 3	0.1		V/ns
T_{CHCL}	Clock Fall Time (peak to peak)	2, 3	0.1		V/ns
T_{SLCH}	S# Active Setup Time (relative to C)		5		ns
T_{CHSL}	S# Active Hold Time (relative to C)		5		ns
T_{DVCH}	Data Input Setup Time		2		ns
T_{CHDX}	Data Input Hold Time		5		ns
T_{CHSH}	S# Active Hold Time (relative to C)		5		ns
T_{SHCH}	S# Inactive Hold Time (relative to C)		5		ns
T_{SHSL}	S# Deselect Time		100		ns
T_{SHQZ}	Output Disable Time	2		8	ns
T_{CLQV}	Clock Low to Output Valid			9	ns
T_{CLQX}	Output Hold Time		0		ns
T_{HLCH}	HOLD# Assertion Setup Time (relative to C)		5		ns
T_{CHHH}	HOLD# Assertion Hold Time (relative to C)		5		ns
T_{HHCH}	HOLD# De-assertion Setup Time (relative to C)		5		ns
T_{CHHL}	HOLD# De-assertion Hold Time (relative to C)		5		ns
T_{HHQX}	HOLD# De-assertion to Output Low-Z	2		10	ns
T_{HLQZ}	HOLD# De-assertion to Output High-Z	2		10	ns
T_{WHSL}	W# Setup Time	4	20		ns
T_{SHWL}	W# Hold Time	4	100		ns

Notes:
1. $T_{CH} + T_{CL}$ must be greater than or equal to $1/F_C$(max).
2. Sampled, not 100% tested.
3. Expressed as a slew-rate
4. Only applicable as a constraint for a WRSR instruction when SRWD is set to 1.

Figure 26: Serial Input Timing

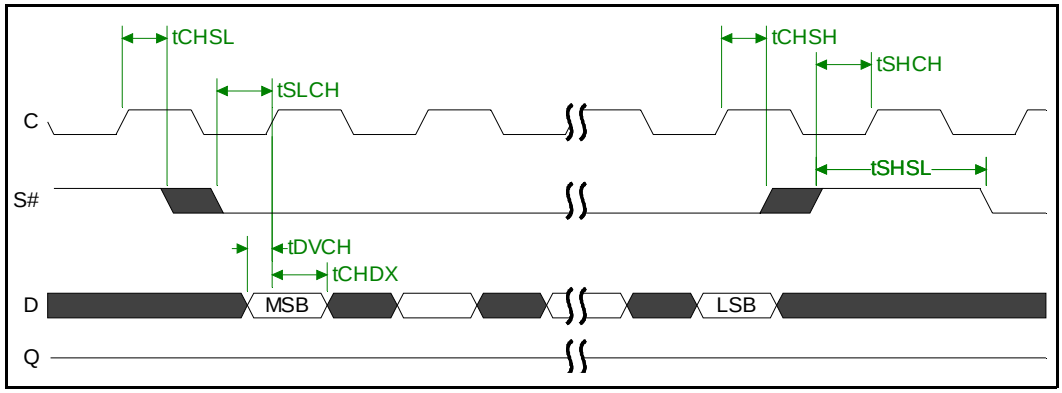


Figure 27: Write Protect Setup and Hold Timing during WRSR when SRWD=1

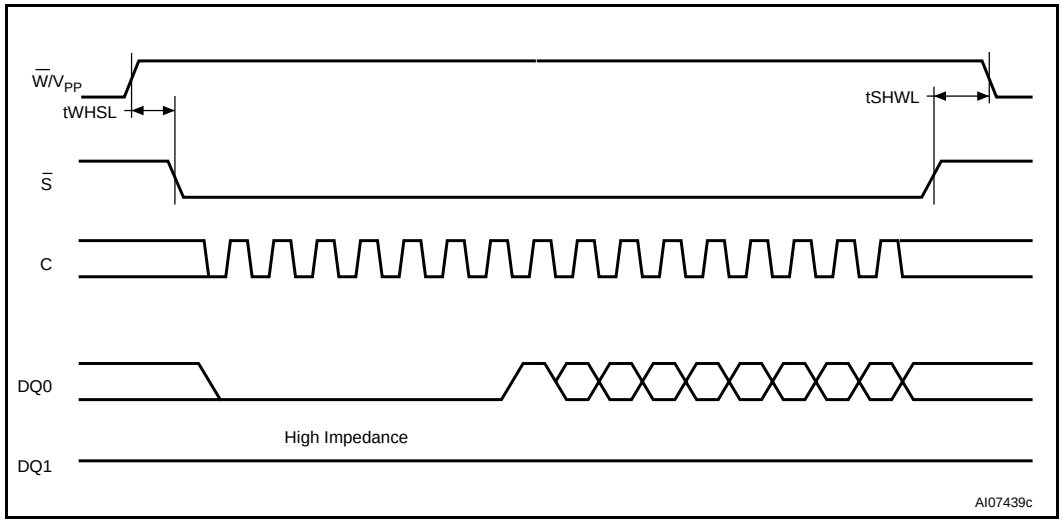


Figure 28: Hold Timing

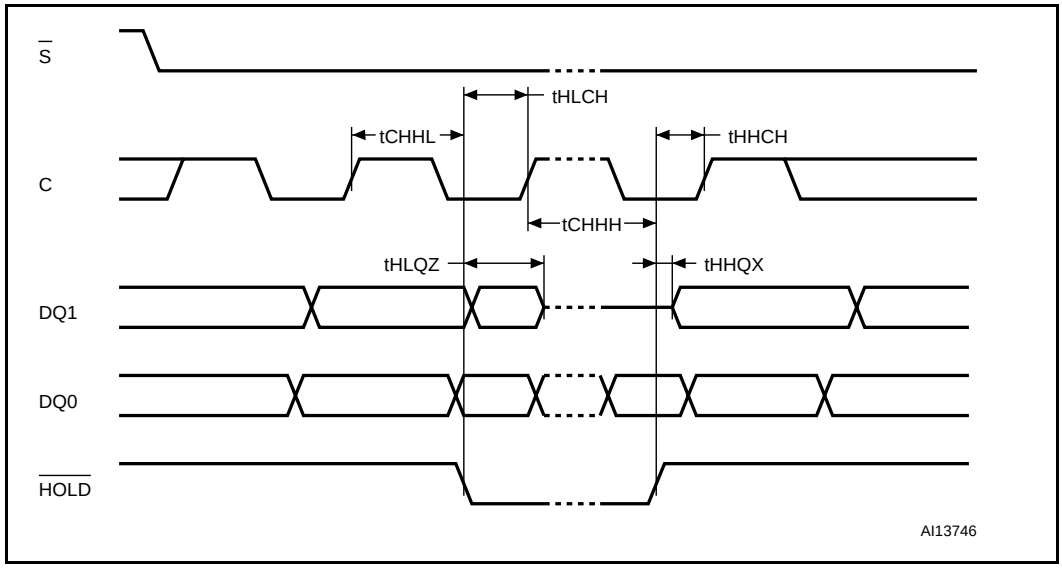
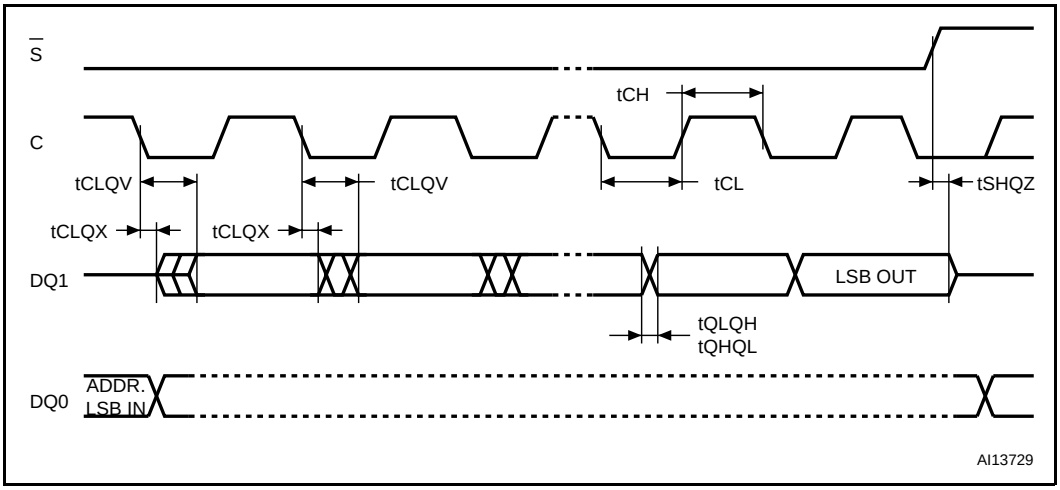


Figure 29: Output Timing - AI13729



17.0 Program and Erase Characteristics

Table 32: Program and Erase Specification

Operation(1)	Symbol	Parameter	Description	V_{PPL}(4,5)			Unit
				Min	Typ	Max	
Erasing and Suspending							
Erase to Suspend	W602 [3]	$t_{ERS/SUSP}$	Erase or Erase-Resume command to Erase-suspend command		500		µs
Erase Time	W500	$t_{ERS/PB}$	16-KW Parameter		100	200	ms
	W501	$t_{ERS/MB}$	64-KW Main		400	800	
Suspend Latency	W600	$t_{SUSP/P}$	Write suspend		35	60	µs
	W601	$t_{SUSP/E}$	Erase suspend		35	60	
Conventional Word Programming							
Program Time(6)	W200	$t_{PROG/W}$	Single word		60	120	µs
Buffered Programming							
Program Time	W200	$t_{PROG/W}$	Single word (Legacy Program & Bit-alterable Write)	5	120	240	µs
	W250	$t_{PROG/PB}$	One Buffer (64 Bytes/32 words) (Legacy Program & Bit-alterable Write)	4,5	120	360	µs
			One Buffer (64 Bytes/32 words) (Program on all 1s)		71	280	

Notes:
1. Typical values measured at T_A = +25 °C, typical voltages and 50% data pattern per word. Excludes system overhead. Performance numbers are valid for all speed versions. Sampled, but not 100% tested.
2. Averaged over entire device.
3. W602 is the minimum time between an initial block erase or erase resume command and the a subsequent erase suspend command. Violating the specification *repeatedly* during any particular block erase may cause erase failures in flash devices. This specification is required if the designer wishes to maintain compatibility with the P33 NOR flash device. However, it is not required with PCM.
4. These performance numbers are valid for all speed versions.
5. Sampled, not 100% tested.

18.0 Ordering Information

This section defines all active line items that can be ordered.

Table 33: Active Line Item Ordering Table (0 to +70 °C)

Part Number	Description
NP8P128A13BSM60E	P8P 128Mb TSOP 14x20 Bottom Boot
NP8P128A13TSM60E	P8P 128Mb TSOP 14x20 Top Boot
NP8P128A13B1760E	P8P 128M leadfree 10x8x1.2 easyBGA Bottom Boot
NP8P128A13T1760E	P8P 128M leadfree 10x8x1.2 easyBGA Top Boot

Table 34: Active Line Item Ordering Table (-30 to +85 °C)

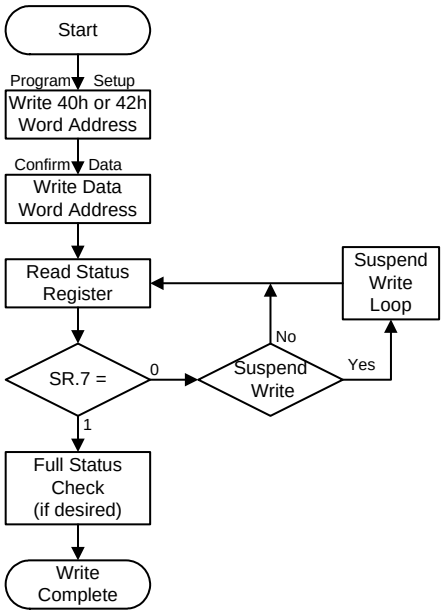
Part Number	Description
NP8P128AE3BSM60E	P8P 128Mb TSOP 14x20 Bottom Boot
NP8P128AE3TSM60E	P8P 128Mb TSOP 14x20 Top Boot
NP8P128AE3B1760E	P8P 128M leadfree 10x8x1.2 easyBGA Bottom Boot
NP8P128AE3T1760E	P8P 128M leadfree 10x8x1.2 easyBGA Top Boot

Appendix A Supplemental Reference Information

A.1 Flow Charts

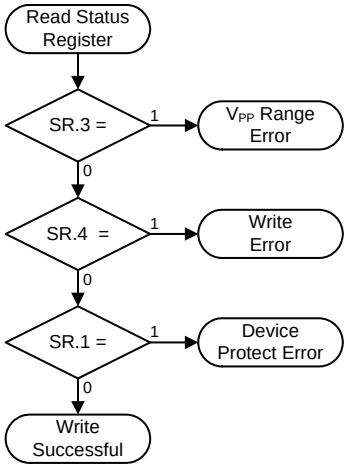
Figure 30: Word Programming or Bit Alterable Write Flowchart

WORD PROGRAM or BIT ALTERABLE WORD WRITE PROCEDURE



Bus Operation	Command	Comments
Write	Program/ Write Setup	Data = 40h or 42h (Bit Alterable) Addr = Location to write (WA)
Write	Data	Data = Data to be written (WD) Addr = Location to be written (WA)
Read		Status register data. Initiate a read cycle to update status register
Standby		Check SR.7 1 = WSM ready 0 = WSM busy
Repeat for subsequent write operations. Full status register check can be done after each write or after a sequence of write operations. Write FFh after the last operation to enter read array mode.		

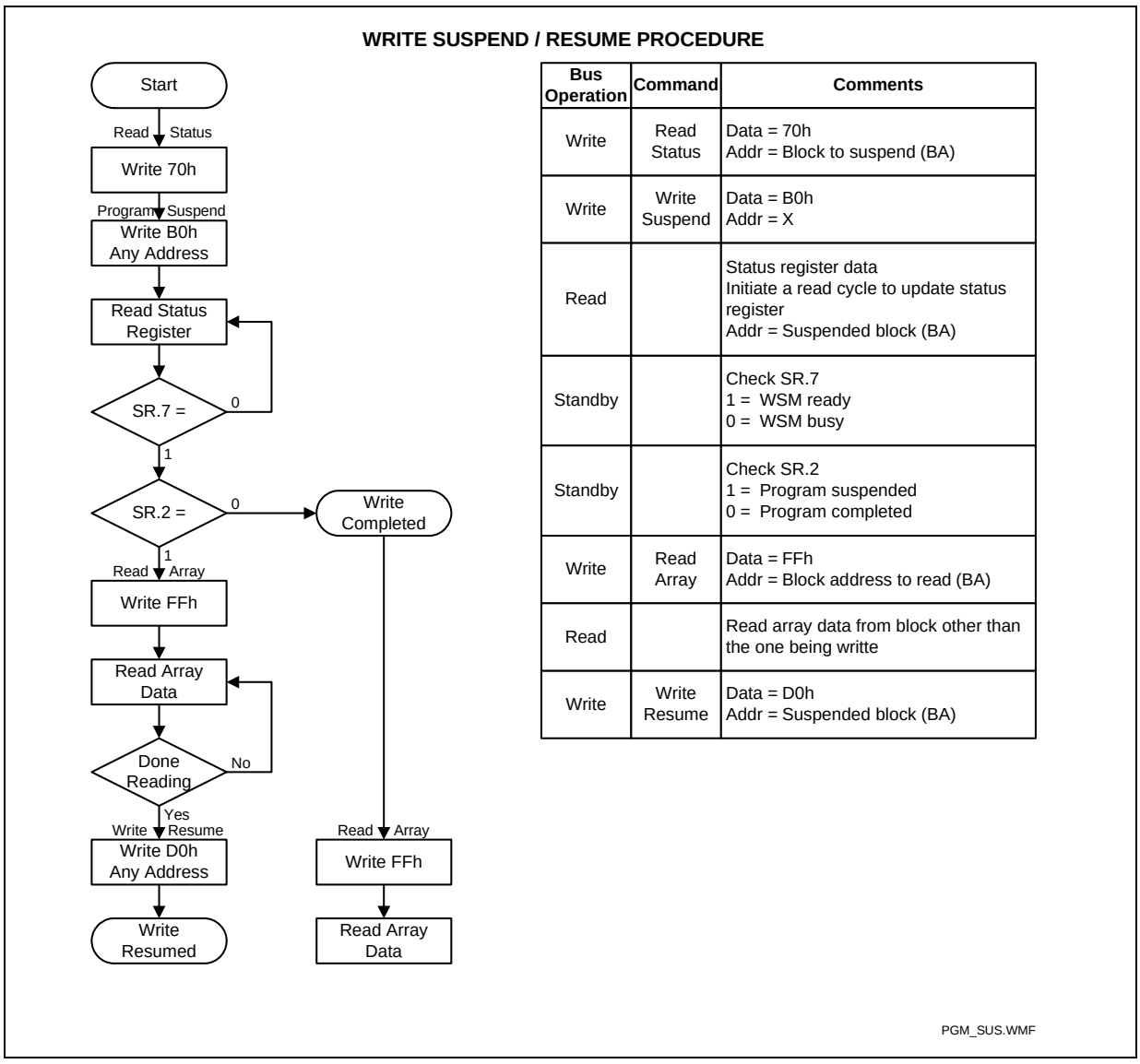
FULL WRITE STATUS CHECK PROCEDURE



Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} error
Standby		Check SR.4 1 = Data write error
Standby		Check SR.1 1 = Attempted write to locked block Write aborted
SR.3 MUST be cleared before the Write State Machine will allow further write attempts Only the Clear Staus Register command clears SR.1, 3, 4. If an error is detected, clear the status register before attempting a write retry or other error recovery.		

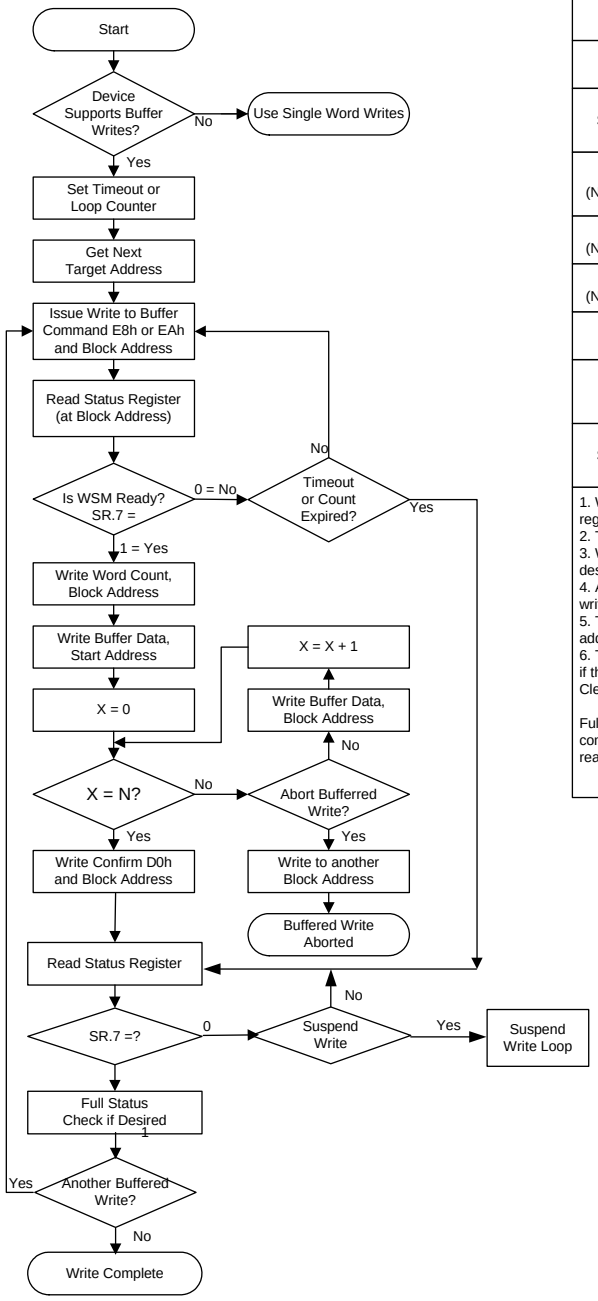
PGM_WRD.WMF

Figure 31: Write Suspend/Resume Flowchart



Bus Operation	Command	Comments
Write	Read Status	Data = 70h Addr = Block to suspend (BA)
Write	Write Suspend	Data = B0h Addr = X
Read		Status register data Initiate a read cycle to update status register Addr = Suspended block (BA)
Standby		Check SR.7 1 = WSM ready 0 = WSM busy
Standby		Check SR.2 1 = Program suspended 0 = Program completed
Write	Read Array	Data = FFh Addr = Block address to read (BA)
Read		Read array data from block other than the one being writte
Write	Write Resume	Data = D0h Addr = Suspended block (BA)

Figure 32: Buffer Program or Bit Alterable Buffer Write Flowchart

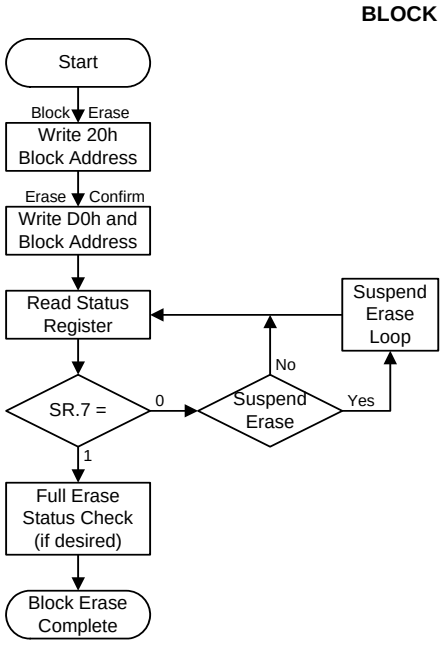


Bus Operation	Command	Comments
Write	Write to Buffer	Data = E8H or EAH (Bit Alterable) Addr = Block Address
Read		SR.7 = Valid Addr = Block Address
Standby		Check SR.7 1 = Device WSM is Busy 0 = Device WSM is Ready
Write (Notes 1, 2)		Data = N-1 = Word Count N = 0 corresponds to count = 1 Addr = Block Address
Write (Notes 3, 4)		Data = Write Buffer Data Addr = Start Address
Write (Notes 5, 6)		Data = Write Buffer Data Addr = Block Address
Write	Write Confirm	Data = D0H Addr = Block Address
Read		Status register Data CE# and OE# low updates SR Addr = Block Address
Standby		Check SR.7 1 = WSM Ready 0 = WSM Busy

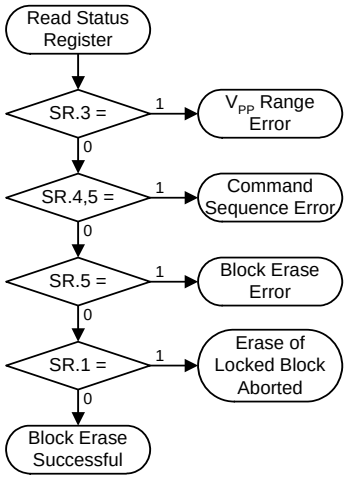
1. Word count values on DQ_0-DQ_7 are loaded into the Count register. Count ranges for this device are N = 0000h to 0001Fh.
2. The device outputs the status register when read.
3. Write Buffer contents will be written at the device start address or destination flash address.
4. Align the start address on a Write Buffer boundary for maximum write performance (i.e., A_5–A_1 of the start address = 0).
5. The device aborts the Buffered Program command if the current address is outside the original block address.
6. The Status register indicates an "improper command sequence" if the Buffered Program command is aborted. Follow this with a Clear Status Register command.

Full status check can be done after all erase and write sequences complete. Write FFh after the last operation to reset the device to read array mode.

Figure 33: Block Erase Flowchart



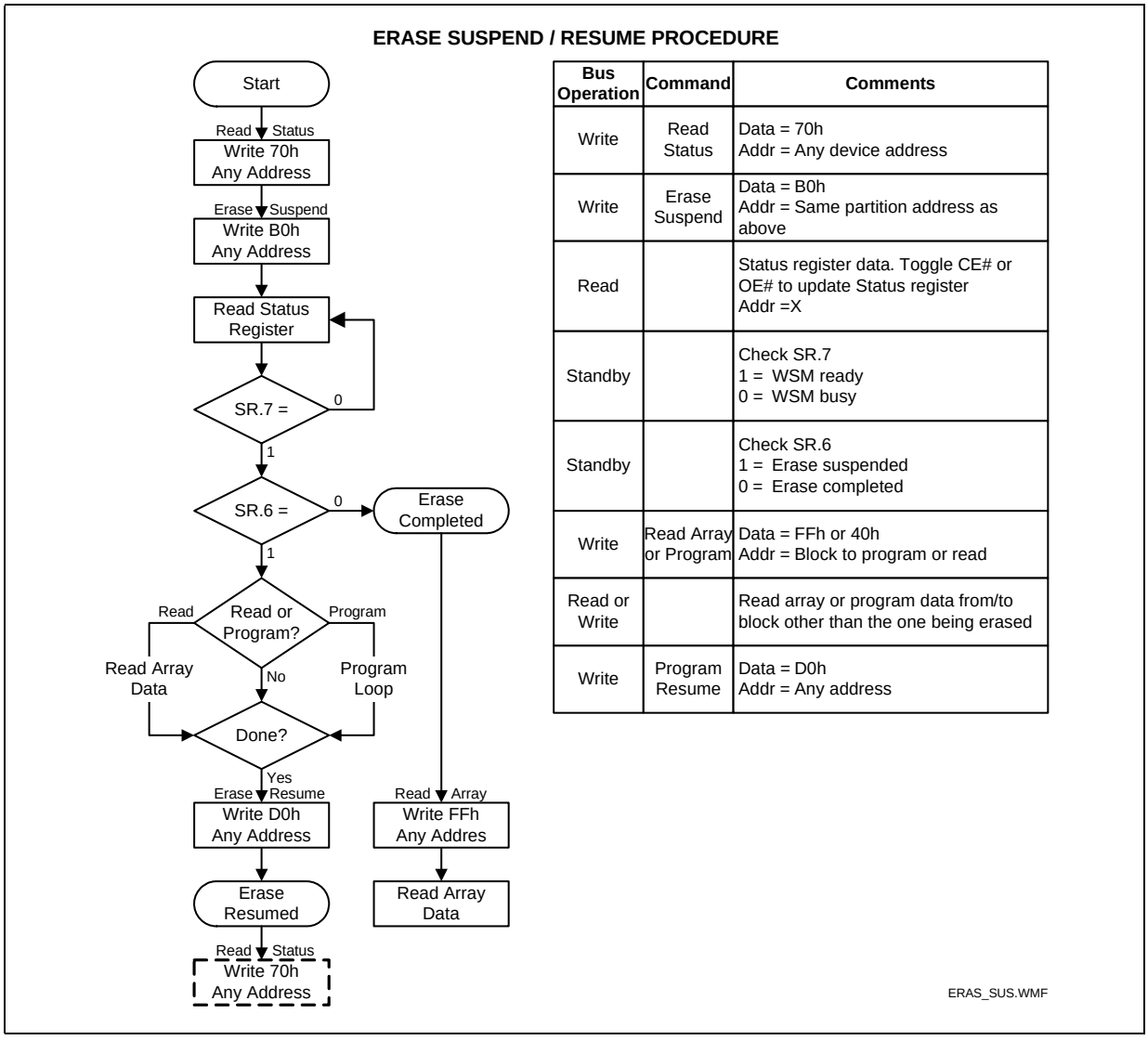
Bus Operation	Command	Comments
Write	Block Erase Setup	Data = 20h Addr = Block to be erased (BA)
Write	Erase Confirm	Data = D0h Addr = Block to be erased (BA)
Read		Status register data. Toggle CE# or OE# to update Status register data
Standby		Check SR.7 1 = WSM ready 0 = WSM busy
Repeat for subsequent block erasures. Full Status register check can be done after each block erase or after a sequence of block erasures. Write FFh after the last operation to enter read array mode.		



Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} error
Standby		Check SR.4,5 Both 1 = Command sequence error
Standby		Check SR.5 1 = Block erase error
Standby		Check SR.1 1 = Attempted erase of locked block Erase aborted
Only the Clear Staus Register command clears SR.1, 3, 4, 5. If an error is detected, clear the Status register before attempting an erase retry or other error recovery.		

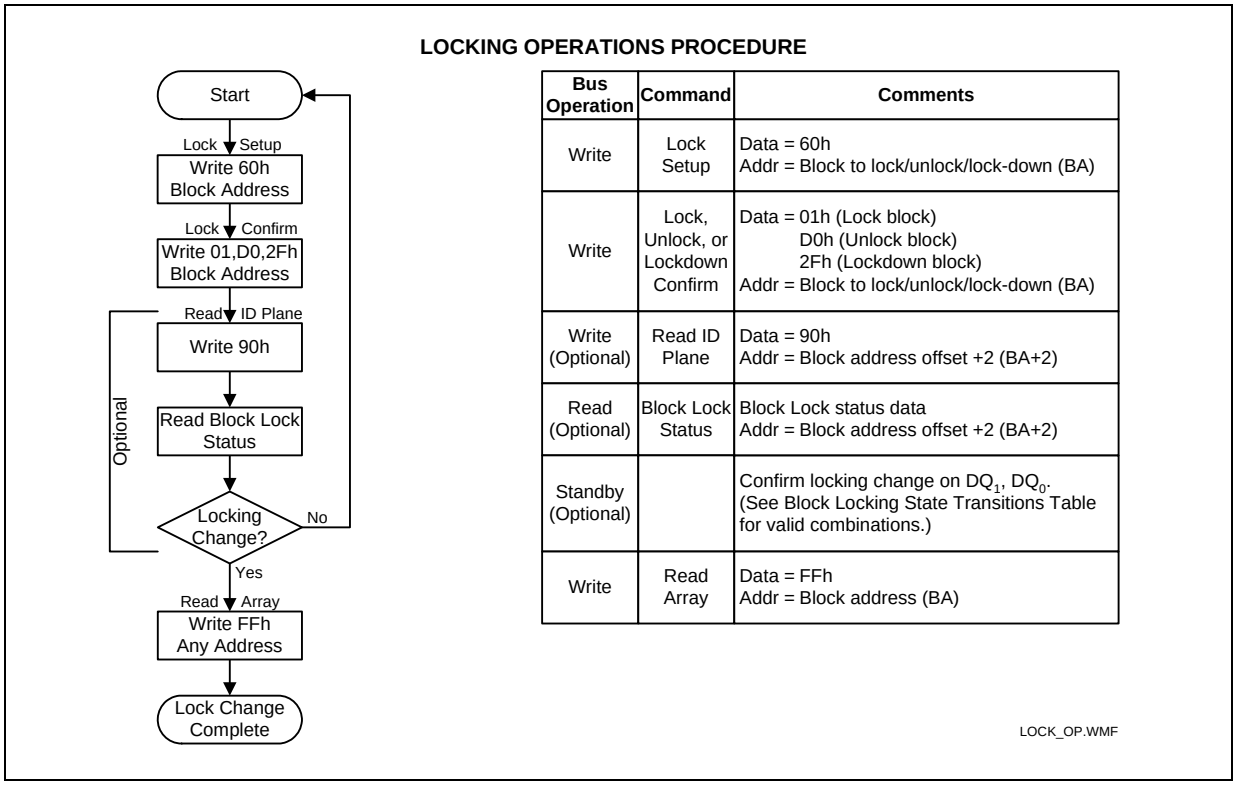
ERAS_BLK.WMF

Figure 34: Erase Suspend / Resume Flowchart



Bus Operation	Command	Comments
Write	Read Status	Data = 70h Addr = Any device address
Write	Erase Suspend	Data = B0h Addr = Same partition address as above
Read		Status register data. Toggle CE# or OE# to update Status register Addr =X
Standby		Check SR.7 1 = WSM ready 0 = WSM busy
Standby		Check SR.6 1 = Erase suspended 0 = Erase completed
Write	Read Array or Program	Data = FFh or 40h Addr = Block to program or read
Read or Write		Read array or program data from/to block other than the one being erased
Write	Program Resume	Data = D0h Addr = Any address

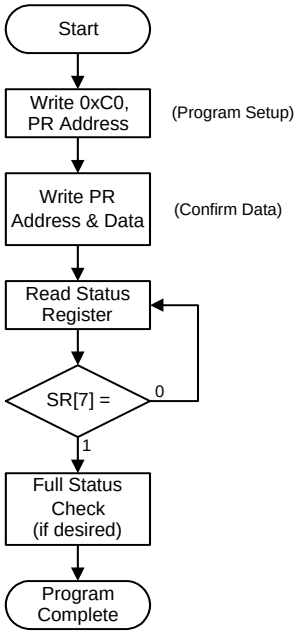
Figure 35: Locking Operations Flowchart



Bus Operation	Command	Comments
Write	Lock Setup	Data = 60h Addr = Block to lock/unlock/lock-down (BA)
Write	Lock, Unlock, or Lockdown Confirm	Data = 01h (Lock block) D0h (Unlock block) 2Fh (Lockdown block) Addr = Block to lock/unlock/lock-down (BA)
Write (Optional)	Read ID Plane	Data = 90h Addr = Block address offset +2 (BA+2)
Read (Optional)	Block Lock Status	Block Lock status data Addr = Block address offset +2 (BA+2)
Standby (Optional)		Confirm locking change on DQ_1, DQ_0. (See Block Locking State Transitions Table for valid combinations.)
Write	Read Array	Data = FFh Addr = Block address (BA)

Figure 36: Protection Register Programming Flowchart

PROTECTION REGISTER PROGRAMMING PROCEDURE



Bus Operation	Command	Comments
Write	Program PR Setup	Data = 0xC0 Addr = First Location to Program
Write	Protection Program	Data = Data to Program Addr = Location to Program
Read	None	Status Register Data.
Idle	None	Check SR[7]: 1 = WSM Ready 0 = WSM Busy

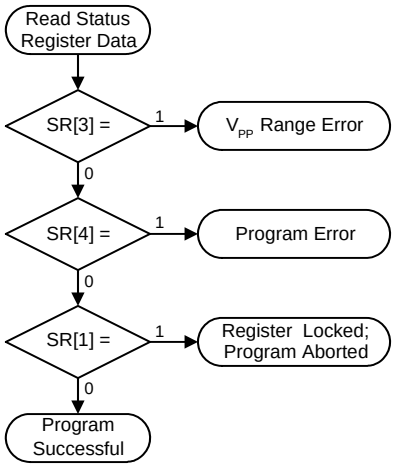
Program Protection Register operation addresses must be within the Protection Register address space. Addresses outside this space will return an error.

Repeat for subsequent programming operations.

Full Status Register check can be done after each program, or after a sequence of program operations.

Write 0xFF after the last operation to set Read Array state.

FULL STATUS CHECK PROCEDURE



Bus Operation	Command	Comments
Idle	None	Check SR[3]: 1 =V_{PP} Range Error
Idle	None	Check SR[4]: 1 =Programming Error
Idle	None	Check SR[1]: 1 =Block locked; operation aborted

Only the Clear Staus Register command clears SR[1, 3, 4].

If an error is detected, clear the Status register before attempting a program retry or other error recovery.

A.2 Write State Machine

Figure 37, "Write State Machine — Next State Table (sheet 1)" on page 80 to Figure 40, "Write State Machine — Output Next State Table (Sheet 4)" on page 83 shows the command state transitions based on incoming commands.

Figure 37: Write State Machine — Next State Table (sheet 1)

Current Chip State (6)		Command Input to Chip and resulting *Chip* Next State												
		Read Array (2) (FFH)	Word Program (3,4) (10H/40H)	Bit Alterable Word Write (42H)	Write to Buffered Program (BP) (E8H)	Bit Alterable Write to Buffer (EAH)	Streaming Mode Entry (SM Entry) (4AH)	Streaming Mode Exit (SM Exit) (4FH)	Erase Setup (3,4) (20H)	BE Confirm, P/E Resume, ULB, Confirm (7) (D0H)	BP / Prg / Erase Suspend (B0H)	Read Status (70H)	Clear Status Register (5) (50H)	Read ID/Query (90H, 98H)	Lock, Unlock, Lock-down, CR setup (4) (60H)
Ready		Ready	Program Setup		BP Setup		SM Entry Setup	SM Exit Setup	Erase Setup	Ready					Lock/CR Setup
SM Ready		SM Ready	Program Setup		BP Setup		SM Ready	SM Exit Setup	Erase Setup	SM Ready					Lock/CR Setup
Lock/CR Setup		Ready (Lock Error [Botch])								Ready (Unlock Block)	Ready (Lock Error [Botch])				
OTP	Setup	OTP Busy													
	Busy														
Word Program	Setup	Word Program Busy													
	Busy	Word Program Busy									Word Pgm Suspend	Word Program Busy			
	Suspend	Word Program Suspend								Word Pgm Busy	Word Program Suspend				
BP	Setup	BP Load 1													
	BP Load 1 (8)	BP Confirm if Data load complete; ELSE BP Load 2													
	BP Load 2 (8)	BP Confirm if Data load complete; ELSE BP Load 2													
	BP Confirm	Ready (Error [Botch])								BP Busy	Ready (Error [Botch])				
	BP Busy	BP Busy									BP Suspend	BP Busy			
	BP Suspend	BP Suspend								BP Busy	BP Suspend				
Erase	Setup	Ready (Error [Botch])								Erase Busy	Ready (Error [Botch])				
	Busy	Erase Busy									Erase Suspend	Erase Busy			
	Suspend	Erase Suspend	Word Program Setup in Erase Suspend		BP Setup in Erase Suspend				Erase Suspend	Erase Busy	Erase Suspend				Lock/CR Setup in Erase Suspend
Word Program in Erase Suspend	Setup	Word Program Busy in Erase Suspend													
	Busy	Word Program Busy in Erase Suspend									Word Program Suspend in Erase Suspend	Word Program Busy in Erase Suspend Busy			
	Suspend	Word Program Suspend in Erase Suspend								Word Pgm Busy in Erase Suspend	Word Program Suspend in Erase Suspend				
BP in Erase Suspend	Setup	BP Load 1 in Erase Suspend													
	BP Load 1 (8)	BP Confirm in Erase Suspend if Data load complete; ELSE BP Load 2													
	BP Load 2 (8)	BP Confirm in Erase Suspend if Data load complete; ELSE BP Load 2													
	BP Confirm	Erase Suspend (Error [Botch BP])								BP Busy in Erase Suspend	Ready (Error [Botch BP] in Erase Suspend)				
	BP Busy	BP Busy in Erase Suspend									BP Suspend in Erase Suspend	BP Busy in Erase Suspend			
	BP Suspend	BP Suspend in Erase Suspend								BP Busy in Erase Suspend	BP Suspend in Erase Suspend				
Lock/CR Setup in Erase Suspend		Erase Suspend (Lock Error [Botch])								Erase Suspend (Unlock Blk)	Erase Suspend (Lock Error [Botch])				
SM Entry	Setup	Ready (Error [Botch])								SM Entry Busy	Ready (Error [Botch])				
	Busy	SM Entry Busy									SM Ready	SM Entry Busy			
SM Exit	Setup	Ready (Error [Botch])								SM Exit Busy	Ready (Error [Botch])				
	Busy	SM Exit Busy									Ready	SM Exit Busy			

Figure 38: Write State Machine — Next State Table (Sheet 2)

Command Input to Chip and resulting Chip Next State						
OTP Setup (4) (C0H)	Lock Block Confirm (7) (01H)	Lock-Down Block Confirm (7) (2FH)	Write RCR/ECR Confirm (7) (03H,04H)	Block Address (WA0) (9) (XXXXH)	Illegal Cmds+U48 (1) (all other codes)	WSM Operation Completes
OTP Setup	Ready					
SM Ready						N/A
Ready (Lock Error [Botch])	Ready (Lock Block)	Ready (Lock Down Blk)	Ready (Set CR)	Ready (Lock Error [Botch])		
OTP Busy						Ready
Word Program Busy						N/A
Word Program Busy						Ready
Word Program Suspend						N/A
BP Load 1						
BP Confirm if Data load complete; ELSE BP Load 2						
BP Confirm if Data load complete; ELSE BP Load 2				Ready	BP Confirm if Data load complete; ELSE BP Load 2	
Ready (Error [Botch])				Ready (Error [Botch]) (Proceed if unlocked or Lock error)	Ready (Error [Botch])	
BP Busy						Ready
BP suspend						N/A
Ready (Error [Botch])						
Erase Busy						Ready
Erase Suspend						N/A
Word Program Busy in Erase Suspend						
Word Program Busy in Erase Suspend Busy						Erase Suspend
Word Program Suspend in Erase Suspend						N/A
BP Load 1 in Erase Suspend						
BP Confirm in Erase Suspend if Data load complete; ELSE BP Load 2				Ready	BP Confirm in Erase Suspend if Data load complete; ELSE BP Load 2	
BP Confirm in Erase Suspend if Data load complete; ELSE BP Load 2				Ready	BP Confirm in Erase Suspend if Data load complete; ELSE BP Load 2	
Ready (Error [Botch BP] in Erase Suspend)				Ready (Error [Botch]) (Proceed if unlocked or Lock error)	Ready (Error [Botch BP] in Erase Suspend)	
BP Busy in Erase Suspend						Erase Suspend
BP Suspend in Erase Suspend						N/A
Erase Suspend (Error [Botch])	Erase Suspend (Lock Blk)	Erase Suspend (Lock Down Blk)	Erase Suspend (Set CR)	Erase Suspend (Lock Error [Botch])		N/A
Ready (Error [Botch])						
SM Entry Busy						Ready
Ready (Error [Botch])						N/A
SM Exit Busy						Ready

Figure 39: Write State Machine — Output Next State Table (Sheet 3)

	Command Input to Chip and resulting *Output* Mux Next State													
Current chip state	Read Array (2) (FFH)	Word Program Setup (3,4) (10H/40H)	Bit Alterable Word Write (42H)	Write to Buffered Program (BP) (E8H)	Bit Alterable Write to Buffer (EAH)	Streaming Mode Entry (SM Entry) (4AH)	Streaming Mode Exit (SM Exit) (4FH)	Erase Setup (3,4) (20H)	BE Confirm, P/E Resume, ULB Confirm (7) (D0H)	Program/ Erase Suspend (B0H)	Read Status (70H)	Clear Status Register (5) (50H)	Read ID/Query (90H, 98H)	Lock, Unlock, Lock-down, CR setup (4) (60H)
Erase Setup, OTP Setup, BP: Setup, Load 1, Load 2, Confirm, Word Pgm Setup, SM Entry Setup, SM Exit Setup	Status Read													
Lock/CR Setup, Lock/CR Setup in Erase Susp														
OTP Busy	Read Array	Status Read							Output mux does not change.		Status Read	Ready Array	Status Read	Status Read
Ready, SM Ready Erase Suspend, BP Suspend BP Busy, Word Program Busy, Erase Busy, BP Busy BP Busy in Erase Suspend Word Pgm Suspend, Word Pgm Busy in Erase Suspend, Pgm Suspend In Erase Suspend BP Suspend in Erase Suspend SM Entry Busy SM Exit Busy													ID Read	

Figure 40: Write State Machine — Output Next State Table (Sheet 4)

	Command Input to Chip and resulting *Output* Mux Next State						
Current chip state	OTP Setup (4) (C0H)	Lock Block Confirm (7) (01H)	Lock-Down Block Confirm (7) (2FH)	Write RCR/ECR Confirm (7) (03H,04H)	Block Address (WA0) (FFFFH)	Illegal Cmds (1) (all other codes)	WSM Operation Completes
Erase Setup, OTP Setup, BP: Setup, Load 1, Load 2, Confirm, Word Pgm Setup, SM Entry Setup, SM Exit Setup	Status Read						
Lock/CR Setup, Lock/CR Setup in Erase Susp	Status Read			Ready Array	Status Read		
OTP Busy	Status Read	Output mux does not change.			Ready Array	Read Array	
Ready, SM Ready Erase Suspend, BP Suspend							
BP Busy, Word Program Busy, Erase Busy, BP Busy BP Busy in Erase Suspend Word Pgm Suspend, Word Pgm Busy in Erase Suspend, Pgm Suspend In Erase Suspend BP Suspend in Erase Suspend SM Entry Busy SM Exit Busy							

Notes:

1. "Illegal commands" include commands outside of the allowed command set (allowed commands: 40H [pgm], 20H [erase], etc.)
2. If a "Read Array" is attempted while the device is busy writing or erasing, the result will be invalid data. The ID and Query data are located at different locations in the address map.
3. 1st and 2nd cycles of "2 cycles write commands" must be given to the same device address, or unexpected results will occur.
4. The 2nd cycle of the following 2 cycle commands will be ignored by the user interface: Word Program Setup, Erase Setup, OTP Setup, and Lock/Unlock/Lock-down/CR setup when issued in an "illegal condition". Illegal conditions are such as "pgm setup while busy", "erase setup while busy", “Word program suspend”, etc. Thus for example the second cycle of an erase command issued in program suspend will NOT resume the program operation.
5. The Clear Status command only clears the error bits in the Status Register if the device is not in the following modes: 1. WSM running (Pgm Busy, Erase Busy, Pgm Busy In Erase Suspend, OTP Busy, modes); 2. Suspend states (Pgm Suspend, Pgm Suspend In Erase Suspend)
6. The "current state" is that of the "device"
7. Confirm commands (Lock Block, Unlock Block, Lock-Down Block) perform the operation and then move to the Ready State.
8. Buffered programming will botch when a different block address (as compared to address given with E8 command) is written during the BP Load1 and BP Load2 states
9. WA0 refers to the block address latched during the first write cycle of the current operation

A.3 Common Flash Interface

The Numonyx® Omneo™ P8P PCM device borrows from the existing standards established for flash memory, and supports the use of the Common Flash Interface (CFI). This appendix defines the data structure or "database" returned by the CFI Query command. System software should parse this structure to gain critical information such as block size, density, x16, and electrical specifications. Once this information has been obtained, the software will know which command sets to use to enable PCM writes, block erases, and otherwise control the PCM component. The Query is part of an overall specification for multiple command set and control interface descriptions called Common Flash Interface, or CFI.

A.3.1 Query Structure Output

The Query database allows system software to obtain information for controlling the PCM device. This section describes the device's CFI-compliant interface that allows access to Query data.

Query data are presented on the lowest-order data outputs (DQ_{7-0}) only. The numerical offset value is the address relative to the maximum bus width supported by the device. On this family of devices, the Query table device starting address is a 10h, which is a word address for x16 devices.

For a word-wide (x16) device, the first two Query-structure bytes, ASCII "Q" and "R," appear on the low byte at word addresses 10h and 11h. This CFI-compliant device outputs 00h data on upper bytes. The device outputs ASCII "Q" in the low byte (DQ_{7-0}) and 00h in the high byte (DQ_{15-8}).

At Query addresses containing two or more bytes of information, the least significant data byte is presented at the lower address, and the most significant data byte is presented at the higher address.

In all of the following tables, addresses and data are represented in hexadecimal notation, so the "h" suffix has been dropped. In addition, since the upper byte of word-wide devices is always "00h," the leading "00" has been dropped from the table notation and only the lower byte value is shown. Any x16 device outputs can be assumed to have 00h on the upper byte in this mode.

Table 35: Summary of Query Structure Output as a Function of Device and Model

Device	Hex Offset	Hex Code	ASCII Value
Device Addresses	00010: 00011: 00012:	51 52 59	"Q" "R" "Y"

Table 36: Example of Query Structure Output of x16- Devices

Word Addressing:			Byte Addressing:		
Offset	**Hex Code**	**Value**	**Offset**	**Hex Code**	**Value**
A_X–A_0	D_{15}–D_0		A_X–A_0	D_7–D_0	
00010h	0051	"Q"	00010h	51	"Q"
00011h	0052	"R"	00011h	52	"R"
00012h	0059	"Y"	00012h	59	"Y"
00013h	P_ID_{LO}	PrVendor	00013h	P_ID_{LO}	PrVendor
00014h	P_ID_{HI}	ID #	00014h	P_ID_{LO}	ID #
00015h	P_{LO}	PrVendor	00015h	P_ID_{HI}	ID #
00016h	P_{HI}	TblAdr	00016h	...	...
00017h	A_ID_{LO}	AltVendor	00017h		
00018h	A_ID_{HI}	ID #	00018h		
...	...	...	...		

A.3.2 Query Structure Overview

The Query command causes the PCM component to display the Common Flash Interface (CFI) Query structure or "database." The structure sub-sections and address locations are summarized below.

Offset	Sub-Section Name	Description(1)
00000h		Manufacturer Code
00001h		Device Code
(BA+2)h(2)	Block Status register	Block-specific information
00004-Fh	Reserved	Reserved for vendor-specific information
00010h	CFI query identification string	Command set ID and vendor data offset
0001Bh	System interface information	Device timing & voltage information
00027h	Device geometry definition	Flash device layout
P(3)	Primary Intel-specific Extended Query Table	Vendor-defined additional information specific to the Primary Vendor Algorithm

Notes:
1. Refer to the Query Structure Output section and offset 28h for the detailed definition of offset address as a function of device bus width and mode.
2. BA = Block Address beginning location (i.e., 08000h is block 1's beginning location when the block size is 32K-word).
3. Offset 15 defines "P" which points to the Primary Numonyx-specific Extended Query Table.

A.3.3 CFI Query Identification String

The Identification String provides verification that the component supports the Common Flash Interface specification. It also indicates the specification version and supported vendor-specified command set(s).

Table 37: Block Status Register

Offset	Length	Description	Add.	Value
(BA+2)h[1]	1	Block Lock Status Register	BA+2	--00 or --01
		BSR.0 Block lock status 0 = Unlocked 1 = Locked	BA+2	(bit 0): 0 or 1
		BSR.1 Block lock-down status 0 = Not locked down 1 = Locked down	BA+2	(bit 1): 0 or 1
		BSR.4 EFA Block lock status 0 = Unlocked 1 = Locked	BA+2	(bit 4): 0 or 1
		BSR.5 EFA Block lock-down status 0 = Not locked down 1 = Locked down	BA+2	(bit 5): 0 or 1
		BSR 2–3, 6-7: Reserved for future use	BA+2	(bit 2–3, 6-7): 0

Table 38: CFI Identification

Offset	Length	Description	Add.	Hex Code	Value
10h	3	Query-unique ASCII string "QRY"	10: 11: 12:	--51 --52 --59	"Q" "R" "Y"
13h	2	Primary vendor command set and control interface ID code. 16-bit ID code for vendor-specified algorithms	13: 14:	--01 --00	
15h	2	Extended Query Table primary algorithm address	15: 16:	--0A --01	
17h	2	Alternate vendor command set and control interface ID code. 0000h means no second vendor-specified algorithm exists	17: 18:	--00 --00	
19h	2	Secondary algorithm Extended Query Table address. 0000h means none exists	19: 1A:	--00 --00	

Table 39: System Interface Information

Offset	Length	Description	Add.	Hex Code	Value
1Bh	1	V_{CC} logic supply minimum program/erase voltage bits 0–3 BCD 100 mV bits 4–7 BCD volts	1B:	--27	2.7V
1Ch	1	V_{CC} logic supply maximum program/erase voltage bits 0–3 BCD 100 mV bits 4–7 BCD volts	1C:	--36	3.6V
1Dh	1	V_{PP} [programming] supply minimum program/erase voltage bits 0–3 BCD 100 mV bits 4–7 HEX volts	1D:	--09	0.9V
1Eh	1	V_{PP} [programming] supply maximum program/erase voltage bits 0–3 BCD 100 mV bits 4–7 HEX volts	1E:	--36	3.6V
1Fh	1	"n" such that typical single word program time-out = 2^n µ-sec	1F:	--08	256µs
20h	1	"n" such that typical full buffer write time-out = 2^n µ-sec	20:	--09	512µs
21h	1	"n" such that typical block erase time-out = 2^n m-sec	21:	--0A	1s
22h	1	"n" such that typical full chip erase time-out = 2^n m-sec	22:	--00	NA
23h	1	"n" such that maximum word program time-out = 2^n times typical	23:	--01	512µs
24h	1	"n" such that maximum buffer write time-out = 2^n times typical	24:	--01	1024µs
25h	1	"n" such that maximum block erase time-out = 2^n times typical	25:	--02	4s
26h	1	"n" such that maximum chip erase time-out = 2^n times typical	26:	--00	NA

Table 40: Device Geometry Definition

Offset	Length	Description	Add.	Hex Code	Value
27h	1	"n" such that device size = 2^n in number of bytes	27:	See table below	
28h	2	Flash device interface code assignment: "n" such that n+1 specifies the bit field that represents the flash device width capabilities as described in the table: 7 6 5 4 3 2 1 0 — — — — x64 x32 x16 x8 15 14 13 12 11 10 9 8 — — — — — — — —	28: 29:	--01 --00	x16
2Ah	2	"n" such that maximum number of bytes in write buffer = 2^n	2A: 2B:	--06 --00	64
2Ch	1	Number of erase block regions (x) within device: 1. x = 0 means no erase blocking; the device erases in bulk 2. x specifies the number of device regions with one or more contiguous same-size erase blocks. 3. Symmetrically blocked partitions have one blocking region	2C:	See table below	
2Dh	4	Erase Block Region 1 Information bits 0–15 = y, y+1 = number of identical-size erase blocks bits 16–31 = z, region erase block(s) size are z x 256 bytes	2D: 2E: 2F: 30:	See table below	
31h	4	Erase Block Region 2 Information bits 0–15 = y, y+1 = number of identical-size erase blocks bits 16–31 = z, region erase block(s) size are z x 256 bytes	31: 32: 33: 34:	See table below	
35h	4	Reserved for future erase block region information	35: 36: 37: 38:	See table below	

Device Geometry Definition

Address	128 Mbit	
	–B	–T
27:	--18	--18
28:	01:	01:
29:	00:	00:
2A:	06:	06:
2B:	00:	00:
2C:	--02	--02
2D:	--03	--7E
2E:	--00	--00
2F:	--80	--00
30:	--00	--02
31:	--7E	--03
32:	--00	--00
33:	--00	--80
34:	--02	--00
35:	--00	--00
36:	--00	--00
37:	--00	--00
38:	--00	--00

A.3.4 Numonyx-Specific Extended Query Table

Table 41: Primary Vendor-Specific Extended Query

Offset[1] P = 10Ah	Length	Description (Optional flash features and commands)	Add.	Hex Code	Value
(P+0)h	3	Primary extended query table	10A	--50	"P"
(P+1)h		Unique ASCII string "PRI"	10B:	--52	"R"
(P+2)h			10C:	--49	"I"
(P+3)h	1	Major version number, ASCII	10D:	--31	"1"
(P+4)h	1	Minor version number, ASCII	10E:	--34	"4"
(P+5)h	4	Optional feature and command support (1=yes, 0=no)	10F:	--E6	
(P+6)h		*bits 10–31 are reserved; undefined bits are "0." If bit 31 is*	110:	--00	
(P+7)h		*"1" then another 31 bit field of Optional features follows at*	111:	--00	
(P+8)h		*the end of the bit–30 field.*	112:	--00	
		bit 0 Chip erase supported	bit 0 = 0		No
		bit 1 Suspend erase supported	bit 1 = 1		Yes
		bit 2 Suspend program supported	bit 2 = 1		Yes
		bit 3 Legacy lock/unlock supported	bit 3 = 0		No
		bit 4 Queued erase supported	bit 4 = 0		No
		bit 5 Instant individual block locking supported	bit 5 = 1		Yes
		bit 6 Protection bits supported	bit 6 = 1		Yes
		bit 7 Pagemode read supported	bit 7 = 1		Yes
		bit 8 Synchronous read supported	bit 8 = 0		No
		bit 9 Simultaneous operations supported	bit 9 = 0		No
		bit 10 Extended Flash Array Blocks supported	bit 10 = 0		No
		bit 30 CFI Link(s) to follow	bit 30 = 0		No
		bit 31 Another "Optional Features" field to follow	bit 31 = 0		No
(P+9)h	1	Supported functions after suspend: read Array, Status, Query	113:	--01	
		Other supported operations are:			
		bits 1–7 reserved; undefined bits are "0"			
		bit 0 Program supported after erase suspend	bit 0 = 1		Yes
(P+A)h	2	Block status register mask	114:	--03	
(P+B)h		*bits 2–15 are Reserved; undefined bits are "0"*	115:	--00	
		bit 0 Block Lock-Bit Status register active	bit 0 = 1		Yes
		bit 1 Block Lock-Down Bit Status active	bit 1 = 1		Yes
		bit 4 EFA Block Lock-Bit Status register active	bit 4 = 0		No
		bit 5 EFA Block Lock-Down Bit Status active	bit 5 = 0		No
(P+C)h	1	V_{CC} logic supply highest performance program/erase voltage	116:	--33	3.3V
		bits 0–3 BCD value in 100 mV			
		bits 4–7 BCD value in volts			
(P+D)h	1	V_{PP} optimum program/erase supply voltage	117:	--33	3.3V
		bits 0–3 BCD value in 100 mV			
		bits 4–7 HEX value in volts			

Table 42: Protection Register Information

Offset[1] P = 10Ah	Length	Description (Optional flash features and commands)	Add.	Hex Code	Value
(P+E)h	1	Number of Protection register fields in JEDEC ID space. "00h," indicates that 256 protection fields are available	118:	--02	2
(P+F)h	4	Protection Field 1: Protection Description	119:	--80	80h
(P+10)h		This field describes user-available One Time Programmable	11A:	--00	00h
(P+11)h		(OTP) Protection register bytes. Some are pre-programmed	11B:	--03	8 byte
(P+12)h		with device-unique serial numbers. Others are user programmable. Bits 0–15 point to the Protection register Lock byte, the section's first byte. The following bytes are factory pre-programmed and user-programmable. bits 0–7 = Lock/bytes Jedec-plane physical low address bits 8–15 = Lock/bytes Jedec-plane physical high address bits 16–23 = "n" such that 2^n = factory pre-programmed bytes bits 24–31 = "n" such that 2^n = user programmable bytes	11C:	--03	8 byte
(P+13)h	10	Protection Field 2: Protection Description	11D:	--89	89h
(P+14)h		Bits 0–31 point to the Protection register physical Lock-word address in the	11E:	--00	00h
(P+15)h		Jedec-plane.	11F:	--00	00h
(P+16)h		Following bytes are factory or user-programmable.	120:	--00	00h
(P+17)h		bits 32–39 = "n" ¬ n = factory pgm'd groups (low byte)	121:	--00	0
(P+18)h		bits 40–47 = "n" ∪ n = factory pgm'd groups (high byte)	122:	--00	0
(P+19)h		bits 48–55 = "n" \ 2n = factory programmable bytes/group	123:	--00	0
(P+1A)h		bits 56–63 = "n" ¬ n = user pgm'd groups (low byte)	124:	--10	16
(P+1B)h		bits 64–71 = "n" ¬ n = user pgm'd groups (high byte)	125:	--00	0
(P+1C)h		bits 72–79 = "n" ¬ 2^n = user programmable bytes/group	126:	--04	16

Table 43: Read Information

Offset[1] P = 10Ah	Length	Description (Optional flash features and commands)	Add.	Hex Code	Value
(P+1D)h	1	Page Mode Read capability bits 0–7 = "n" such that 2^n HEX value represents the number of read-page bytes. See offset 28h for device word width to determine page-mode data output width. 00h indicates no read page buffer.	127:	--04	16 byte
(P+1E)h	1	Number of synchronous mode read configuration fields that follow. 00h indicates no burst capability.	128:	--00	0

Table 44: Partition and Erase-block Region Information

Offset[1] P = 10Ah Bottom	Offset[1] P = 10Ah Top	Description (Optional flash features and commands)	See table below Len	Address Bot	Address Top
(P+1F)h	(P+1F)h	Number of device hardware-partition regions within the device. x = 0: a single hardware partition device (no fields follow). x specifies the number of device partition regions containing one or more contiguous erase block regions.	1	129:	129:
		Partition Region 1 Information			
(P+20)h	(P+20)h	Data size of this Parition Region Information field	**2**	12A:	12A
(P+21)h	(P+21)h	(# addressable locations, including this field)		12B	12B
(P+22)h	(P+22)h	Number of identical partitions within the partition region	2	12C:	12C:
(P+23)h	(P+23)h			12D:	12D:
(P+24)h	(P+24)h	Number of program or erase operations allowed in a partition bits 0–3 = number of simultaneous Program operations bits 4–7 = number of simultaneous Erase operations	1	12E:	12E:
(P+25)h	(P+25)h	Simultaneous program or erase operations allowed in other partitions while a partition in this region is in Program mode bits 0–3 = number of simultaneous Program operations bits 4–7 = number of simultaneous Erase operations	1	12F:	12F:
(P+26)h	(P+26)h	Simultaneous program or erase operations allowed in other partitions while a partition in this region is in Erase mode bits 0–3 = number of simultaneous Program operations bits 4–7 = number of simultaneous Erase operations	1	130:	130:
(P+27)h	(P+27)h	Types of erase block regions in this Partition Region. x = 0 = no erase blocking; the Partition Region erases in bulk x = number of erase block regions w/ contiguous same-size erase blocks. Symmetrically blocked partitions have one blocking region. Partition size = (Type 1 blocks)x(Type 1 block sizes) + (Type 2 blocks)x(Type 2 block sizes) +…+ (Type n blocks)x(Type n block sizes)	1	131:	131:
(P+28)h	(P+28)h	Partition Region 1 Erase Block Type 1 Information	4	132:	132:
(P+29)h	(P+29)h	bits 0–15 = y, y+1 = # identical-size erase blks in a partition		133:	133:
(P+2A)h	(P+2A)h	bits 16–31 = z, region erase block(s) size are z x 256 bytes		134:	134:
(P+2B)h	(P+2B)h			135:	135:
(P+2C)h	(P+2C)h	Partition 1 (Erase Block Type 1)	2	136:	136:
(P+2D)h	(P+2D)h	Block erase cycles x 1000		137:	137:
(P+2E)h	(P+2E)h	Partition 1 (erase block Type 1) bits per cell; internal EDAC bits 0–3 = bits per cell in erase region bit 4 = internal EDAC used (1=yes, 0=no) bits 5–7 = reserve for future use	1	138:	138:
(P+2F)h	(P+2F)h	Partition 1 (erase block Type 1) page mode and synchronous mode capabilities defined in Table 10. bit 0 = page-mode host reads permitted (1=yes, 0=no) bit 1 = synchronous host reads permitted (1=yes, 0=no) bit 2 = synchronous host writes permitted (1=yes, 0=no) bits 3–7 = reserved for future use	1	139:	139:

Table 45: Partition and Erase-block Region Information

		Partition Region 1 (Erase Block Type 1) Programming Region Information	6		
(P+30)h	(P+30)h	bits 0–7 = x, 2^x = Programming Region aligned size (**bytes**)		13A:	13A:
(P+31)h	(P+31)h	bits 8–14 = Reserved; bit 15 = Legacy flash operation (ignore 0:7)		13B:	13B:
(P+32)h	(P+32)h	bits 16–23 = y = Control Mode **valid** size in bytes		13C:	13C:
(P+33)h	(P+33)h	bits 24-31 = Reserved		13D:	13D:
(P+34)h	(P+34)h	bits 32-39 = z = Control Mode **invalid** size in bytes		13E:	13E:
(P+35)h	(P+35)h	bits 40-46 = Reserved; bit 47 = Legacy flash operation (ignore 23:16 & 39:32)		13F:	13F:
(P+36)h	(P+36)h	Partition Region 1 Erase Block Type 2 Information	4	140:	140:
(P+37)h	(P+37)h	bits 0–15 = y, y+1 = # identical-size erase blks in a partition		141:	141:
(P+38)h	(P+38)h	bits 16–31 = z, region erase block(s) size are z x 256 bytes		142:	142:
(P+39)h	(P+39)h			143:	143:
(P+3A)h	(P+3A)h	Partition 1 (Erase Block Type 2)	2	144:	144:
(P+3B)h	(P+3B)h	Block erase cycles x 1000		145:	145:
(P+3C)h	(P+3C)h	Partition 1 (erase block Type 2) bits per cell; internal EDAC bits 0–3 = bits per cell in erase region bit 4 = internal EDAC used (1=yes, 0=no) bits 5–7 = reserve for future use	1	146:	146:
(P+3D)h	(P+3D)h	Partition 1 (erase block Type 2) page mode and synchronous mode capabilities defined in Table 10. bit 0 = page-mode host reads permitted (1=yes, 0=no) bit 1 = synchronous host reads permitted (1=yes, 0=no) bit 2 = synchronous host writes permitted (1=yes, 0=no) bits 3–7 = reserved for future use	1	147:	147:
		Partition Region 1 (Erase Block Type 2) Programming Region Information	6		
(P+3E)h	(P+3E)h	bits 0–7 = x, 2^x = Programming Region aligned size (**bytes**)		148:	148:
(P+3F)h	(P+3F)h	bits 8–14 = Reserved; bit 15 = Legacy flash operation (ignore 0:7)		149:	149:
(P+40)h	(P+40)h	bits 16–23 = y = Control Mode **valid** size in bytes		14A:	14A:
(P+41)h	(P+41)h	bits 24-31 = Reserved		14B:	14B:
(P+42)h	(P+42)h	bits 32-39 = z = Control Mode **invalid** size in bytes		14C:	14C:
(P+43)h	(P+43)h	bits 40-46 = Reserved; bit 47 = Legacy flash operation (ignore 23:16 & 39:32)		14D:	14D:

Table 46: Partition and Erase-block Region Information

Partition and Erase-block Region Information

Address	128 Mbit	
	–B	–T
129:	--01	--01
12A:	--24	--24
12B:	--00	--00
12C:	--01	--01
12D:	--00	--00
12E:	--11	--11
12F:	--00	--00
130:	--00	--00
131:	--02	--02
132:	--03	--7E
133:	--00	--00
134:	--80	--00
135:	--00	--02
136:	--64	--64
137:	--00	--00
138:	--01	--01
139:	--01	--01
13A:	--00	--00
13B:	--80	--80
13C:	--00	--00
13D:	--00	--00
13E:	--00	--00
13F:	--80	--80
140:	--7E	--03
141:	--00	--00
142:	--00	--80
143:	--02	--00
144:	--64	--64
145:	--00	--00
146:	--01	--01
147:	--01	--01
148:	--00	--00
149:	--80	--80
14A:	--00	--00
14B:	--00	--00
14C:	--00	--00
14D:	--80	--80