



# Numonyx™ StrataFlash® Embedded Memory (P30)

## Datasheet

### Product Features

- High performance
  - 85 ns initial access
  - 52 MHz with zero wait states, 17ns clock-to-data output synchronous-burst read mode
  - 25 ns asynchronous-page read mode
  - 4-, 8-, 16-, and continuous-word burst mode
  - Buffered Enhanced Factory Programming (BEFP) at 5  $\mu$ s/byte (Typ)
  - 1.8 V buffered programming at 7  $\mu$ s/byte (Typ)
- Architecture
  - Multi-Level Cell Technology: Highest Density at Lowest Cost
  - Asymmetrically-blocked architecture
  - Four 32-KByte parameter blocks: top or bottom configuration
  - 128-KByte main blocks
- Voltage and Power
  - $V_{CC}$  (core) voltage: 1.7 V – 2.0 V
  - $V_{CCQ}$  (I/O) voltage: 1.7 V – 3.6 V
  - Standby current: 20 $\mu$ A (Typ) for 64-Mbit
  - 4-Word synchronous read current: 13 mA (Typ) at 40 MHz
- Quality and Reliability
  - Operating temperature: –40 °C to +85 °C
  - Minimum 100,000 erase cycles per block
  - ETOX™ VIII process technology
- 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 128-KByte blocks (top or bottom configuration)
    - Up to Full Array OTP Lockout
  - Absolute write protection:  $V_{PP} = V_{SS}$
  - Power-transition erase/program lockout
  - Individual zero-latency block locking
  - Individual block lock-down
- Software
  - 20  $\mu$ s (Typ) program suspend
  - 20  $\mu$ s (Typ) erase suspend
  - Numonyx™ Flash Data Integrator optimized
  - Basic Command Set and Extended Command Set compatible
  - Common Flash Interface capable
- Density and Packaging
  - 56- Lead TSOP package (64, 128, 256, 512- Mbit)
  - 64- Ball Numonyx™ Easy BGA package (64, 128, 256, 512- Mbit)
  - Numonyx™ QUAD+ SCSP (64, 128, 256, 512- Mbit)
  - 16-bit wide data bus



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH NUMONYX™ PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN NUMONYX'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, NUMONYX ASSUMES NO LIABILITY WHATSOEVER, AND NUMONYX DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF NUMONYX PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Numonyx products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Numonyx B.V. may make changes to specifications and product descriptions at any time, without notice.

Numonyx B.V. may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Numonyx reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Contact your local Numonyx sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Numonyx literature may be obtained by visiting Numonyx's website at <http://www.numonyx.com>.

Numonyx, the Numonyx logo, and StrataFlash are trademarks or registered trademarks of Numonyx B.V. or its subsidiaries in other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2007, Numonyx, B.V., All Rights Reserved.

## Contents

---

|             |   |    |
|-------------|---|----|
| <b>1.0</b>  | <b>Introduction</b> .....                             | 6  |
| 1.1         | Nomenclature .....                                    | 6  |
| 1.2         | Acronyms.....   | 6  |
| 1.3         | Conventions .....                                     | 7  |
| <b>2.0</b>  | <b>Functional Overview</b> .....                      | 8  |
| 2.1         | Virtual Chip Enable Description.....                  | 8  |
| <b>3.0</b>  | <b>Package Information</b> .....                      | 10 |
| 3.1         | 56-Lead TSOP.....                                     | 10 |
| 3.2         | 64-Ball Easy BGA Package.....                         | 11 |
| 3.3         | QUAD+ SCSP Packages.....                              | 13 |
| <b>4.0</b>  | <b>Ballout and Signal Descriptions</b> .....          | 16 |
| 4.1         | Signal Ballout .....                                  | 16 |
| 4.2         | Signal Descriptions.....                              | 19 |
| 4.3         | Dual-Die Configurations .....                         | 21 |
| 4.4         | Memory Maps .....                                     | 22 |
| <b>5.0</b>  | <b>Maximum Ratings and Operating Conditions</b> ..... | 25 |
| 5.1         | Absolute Maximum Ratings.....                         | 25 |
| 5.2         | Operating Conditions .....                            | 25 |
| <b>6.0</b>  | <b>Electrical Specifications</b> .....                | 26 |
| 6.1         | DC Current Characteristics.....                       | 26 |
| 6.2         | DC Voltage Characteristics.....                       | 27 |
| <b>7.0</b>  | <b>AC Characteristics</b> .....                       | 28 |
| 7.1         | AC Test Conditions .....                              | 28 |
| 7.2         | Capacitance.....                                      | 29 |
| 7.3         | AC Read Specifications.....                           | 29 |
| 7.4         | AC Write Specifications .....                         | 36 |
| 7.5         | Program and Erase Characteristics.....                | 39 |
| <b>8.0</b>  | <b>Power and Reset Specifications</b> .....           | 41 |
| 8.1         | Power-Up and Power-Down.....                          | 41 |
| 8.2         | Reset Specifications.....                             | 41 |
| 8.3         | Power Supply Decoupling .....                         | 42 |
| <b>9.0</b>  | <b>Device Operations</b> .....                        | 43 |
| 9.1         | Bus Operations .....                                  | 43 |
| 9.1.1       | Reads .....   | 43 |
| 9.1.2       | Writes.....   | 44 |
| 9.1.3       | Output Disable.....                                   | 44 |
| 9.1.4       | Standby .....   | 44 |
| 9.1.5       | Reset.....  | 44 |
| 9.2         | Device Commands.....                                  | 45 |
| 9.3         | Command Definitions.....                              | 46 |
| <b>10.0</b> | <b>Read Operations</b> .....                          | 48 |
| 10.1        | Asynchronous Page-Mode Read .....                     | 48 |
| 10.2        | Synchronous Burst-Mode Read.....                      | 48 |
| 10.3        | Read Configuration Register.....                      | 49 |
| 10.3.1      | Read Mode .....                                       | 50 |
| 10.3.2      | Latency Count.....                                    | 50 |

|             |   |           |
|-------------|---|-----------|
| 10.3.3      | WAIT Polarity .....                                     | 52        |
| 10.3.4      | Data Hold .....   | 53        |
| 10.3.5      | WAIT Delay .....  | 53        |
| 10.3.6      | Burst Sequence .....                                    | 54        |
| 10.3.7      | Clock Edge .....  | 54        |
| 10.3.8      | Burst Wrap .....  | 55        |
| 10.3.9      | Burst Length .....                                      | 55        |
| 10.3.10     | End of Word Line (EOWL) Considerations .....            | 55        |
| <b>11.0</b> | <b>Programming Operations .....</b>                     | <b>56</b> |
| 11.1        | Word Programming .....                                  | 56        |
| 11.1.1      | Factory Word Programming .....                          | 57        |
| 11.2        | Buffered Programming .....                              | 57        |
| 11.3        | Buffered Enhanced Factory Programming .....             | 58        |
| 11.3.1      | BEFP Requirements and Considerations .....              | 58        |
| 11.3.2      | BEFP Setup Phase .....                                  | 59        |
| 11.3.3      | BEFP Program/Verify Phase .....                         | 59        |
| 11.3.4      | BEFP Exit Phase .....                                   | 60        |
| 11.4        | Program Suspend .....                                   | 60        |
| 11.5        | Program Resume .....                                    | 60        |
| 11.6        | Program Protection .....                                | 61        |
| <b>12.0</b> | <b>Erase Operations .....</b>                           | <b>62</b> |
| 12.1        | Block Erase .....                                       | 62        |
| 12.2        | Erase Suspend .....                                     | 62        |
| 12.3        | Erase Resume .....                                      | 63        |
| 12.4        | Erase Protection .....                                  | 63        |
| <b>13.0</b> | <b>Security Modes .....</b>                             | <b>64</b> |
| 13.1        | Block Locking .....                                     | 64        |
| 13.1.1      | Lock Block .....  | 64        |
| 13.1.2      | Unlock Block .....                                      | 64        |
| 13.1.3      | Lock-Down Block .....                                   | 64        |
| 13.1.4      | Block Lock Status .....                                 | 65        |
| 13.1.5      | Block Locking During Suspend .....                      | 65        |
| 13.2        | Selectable One-Time Programmable Blocks .....           | 66        |
| 13.3        | Protection Registers .....                              | 66        |
| 13.3.1      | Reading the Protection Registers .....                  | 67        |
| 13.3.2      | Programming the Protection Registers .....              | 68        |
| 13.3.3      | Locking the Protection Registers .....                  | 68        |
| <b>14.0</b> | <b>Special Read States .....</b>                        | <b>69</b> |
| 14.1        | Read Status Register .....                              | 69        |
| 14.1.1      | Clear Status Register .....                             | 70        |
| 14.2        | Read Device Identifier .....                            | 70        |
| 14.3        | CFI Query .....   | 71        |
| <b>A</b>    | <b>Write State Machine .....</b>                        | <b>72</b> |
| <b>B</b>    | <b>Flowcharts .....</b>                                 | <b>79</b> |
| <b>C</b>    | <b>Common Flash Interface .....</b>                     | <b>87</b> |
| <b>D</b>    | <b>Additional Information .....</b>                     | <b>97</b> |
| <b>E</b>    | <b>Ordering Information for Discrete Products .....</b> | <b>98</b> |
| <b>F</b>    | <b>Ordering Information for SCSP Products .....</b>     | <b>99</b> |

## Revision History

| Revision Date  | Revision | Description  |
|----------------|----------|--|
| April 2005     | -001     | Initial Release  |
| August 2005    | -002     | <p>Revised discrete memory maps in <a href="#">Section 4.4, "Memory Maps" on page 22</a></p> <p>Added memory maps for 512-Mbit top parameter devices in <a href="#">Section 4.4, "Memory Maps" on page 22</a></p> <p>Fixed size of Programming Region for 256-Mbit to be 8-Mbit in <a href="#">Section 4.4, "Memory Maps" on page 22</a> and <a href="#">Section 11.0, "Programming Operations" on page 56</a></p> <p>Removed power supply sequencing requirement in <a href="#">Section 8.1, "Power-Up and Power-Down" on page 41</a></p> <p>Updated conditions for <a href="#">Table 15, "Capacitance" on page 29</a></p> <p>Updated CFI table in <a href="#">Appendix C, "Common Flash Interface"</a></p>   |
| September 2005 | -003     | <p>Added note to <a href="#">Table 34, "Device ID codes" on page 71</a> for stacked Device ID codes</p> <p>Synchronous burst read operation is currently not supported for the TSOP package</p> <p>Updated 512-Mbit Easy BGA Ball Height (symbol A1) in <a href="#">Figure 2, "Easy BGA Mechanical Specifications" on page 11</a></p>  |
| November 2005  | -004     | Updated read access speed for 265M TSOP package  |
| February 2006  | -005     | Removed all references to 1 Gigabit.   |
| April 2006     | -006     | <ul style="list-style-type: none"> <li>• Added 52 MHz capabilities,</li> <li>• Added TSOP Package information for 512 Mb throughout the document,</li> <li>• Added <a href="#">Section 2.1, "Virtual Chip Enable Description" on page 8,</a></li> <li>• Modified figures in <a href="#">Section 4.3, "Dual-Die Configurations" on page 21,</a></li> <li>• Modified <a href="#">Table 9, "512-Mbit Top and Bottom Parameter Memory Map (Easy BGA and QUAD+ SCSP)" on page 23,</a></li> <li>• Modified Notes 5 &amp; 6 to Reset Specifications table in <a href="#">Section 8.2, "Reset Specifications" on page 41,</a></li> <li>• Added additional note on 512 Mb capability in <a href="#">Table 31, "Selectable OTP Block Mapping" on page 66.</a></li> </ul> |
| May 2006       | -007     | <ul style="list-style-type: none"> <li>• Updated the following tables to 52 MHz: <a href="#">Table 16, "AC Read Specifications for 64/128-Mbit Densities" on page 29</a> and <a href="#">Table 17, "AC Read Specifications for 256/512-Mbit Densities" on page 30.</a></li> </ul>  |
| May-2006       | -008     | <ul style="list-style-type: none"> <li>• Added notes 1, 2, and 3 to <a href="#">Table 15, "Capacitance" on page 29.</a></li> </ul>   |
| June - 2007    | -009     | <ul style="list-style-type: none"> <li>• Correct typos and add clarifications</li> <li>• Enabled specific burst operation on TSOP packages.</li> <li>• Updated device commands table.</li> <li>• Updaed description on synchronous burst operation.</li> <li>• Added EOWL description.</li> <li>• Updated flowcharts</li> </ul>  |
| November 2007  | -010     | <ul style="list-style-type: none"> <li>• Updated for 65nm lithography</li> <li>• Added W602 - Erase to Suspend</li> </ul>  |
| November 2007  | 11       | <ul style="list-style-type: none"> <li>• Applied Numonyx branding.</li> </ul>  |

## 1.0 Introduction

This document provides information about the Numonyx™ StrataFlash® Embedded Memory (P30) product and describes its features, operation, and specifications.

The Numonyx™ StrataFlash® Embedded Memory (P30) product is the latest generation of Numonyx™ StrataFlash® memory devices. Offered in 64-Mbit up through 512-Mbit densities, the P30 device brings reliable, two-bit-per-cell storage technology to the embedded flash market segment. Benefits include more density in less space, high-speed interface, lowest cost-per-bit NOR device, and support for code and data storage. Features include high-performance synchronous-burst read mode, fast asynchronous access times, low power, flexible security options, and three industry standard package choices. The P30 product family is manufactured using Intel\* 130 nm ETOX™ VIII process technology.

The P30 product family is also planned on the Intel\* 65nm process lithography. 65nm AC timing changes are noted in this datasheet, and should be taken into account for all new designs.

## 1.1 Nomenclature

|                          |   |
|--------------------------|---|
| 1.8 V:                   | $V_{CC}$ (core) voltage range of 1.7 V – 2.0 V  |
| 3.0 V:                   | $V_{CCQ}$ (I/O) voltage range of 1.7 V – 3.6 V  |
| 9.0 V:                   | $V_{pp}$ voltage range of 8.5 V – 9.5 V   |
| Block:                   | A group of bits, bytes, or words within the flash memory array that erase simultaneously when the Erase command is issued to the device. The P30 has two block sizes: 32-KByte and 128-KByte. |
| Main block:              | An array block that is usually used to store code and/or data. Main blocks are larger than parameter blocks.  |
| Parameter block:         | An array block that is usually used to store frequently changing data or small system parameters that traditionally would be stored in EEPROM.  |
| Top parameter device:    | A device with its parameter blocks located at the highest physical address of its memory map.   |
| Bottom parameter device: | A device with its parameter blocks located at the lowest physical address of its memory map.  |

## 1.2 Acronyms

|       |                                     |
|-------|-------------------------------------|
| BEFP: | Buffer Enhanced Factory Programming |
| CUI:  | Command User Interface              |
| MLC:  | Multi-Level Cell                    |
| OTP:  | One-Time Programmable               |
| PLR:  | Protection Lock Register            |
| PR:   | Protection Register                 |
| RCR:  | Read Configuration Register         |

## **P30**

|      |                         |
|------|-------------------------|
| RFU: | Reserved for Future Use |
| SR:  | Status Register         |
| WSM: | Write State Machine     |

## **1.3 Conventions**

|                   |  |
|-------------------|--|
| VCC:              | Signal or voltage connection   |
| V <sub>CC</sub> : | Signal or voltage level  |
| 0x:               | Hexadecimal number prefix  |
| 0b:               | Binary number prefix   |
| SR[4]:            | Denotes an individual register bit.  |
| A[15:0]:          | Denotes a group of similarly named signals, such as address or data bus.             |
| A5:               | Denotes one element of a signal group membership, such as an individual address bit. |
| Bit:              | Binary unit  |
| Byte:             | Eight bits   |
| Word:             | Two bytes, or sixteen bits   |
| Kbit:             | 1024 bits  |
| KByte:            | 1024 bytes   |
| KWord:            | 1024 words   |
| Mbit:             | 1,048,576 bits   |
| MByte:            | 1,048,576 bytes  |
| MWord:            | 1,048,576 words  |

## 2.0 Functional Overview

This section provides an overview of the features and capabilities of the P30.

The P30 family provides density upgrades from 64-Mbit through 512-Mbit. This family of devices provides high performance at low voltage on a 16-bit data bus. Individually erasable memory blocks are sized for optimum code and data storage.

Upon initial power up or return from reset, the device defaults to asynchronous page-mode read. Configuring the Read Configuration Register enables synchronous burst-mode reads. In synchronous burst mode, output data is synchronized with a user-supplied clock signal. A WAIT signal provides an easy CPU-to-flash memory synchronization.

In addition to the enhanced architecture and interface, the device incorporates technology that enables fast factory program and erase operations. Designed for low-voltage systems, the P30 supports read operations with  $V_{CC}$  at 1.8 V, and erase and program operations with  $V_{PP}$  at 1.8 V or 9.0 V. Buffered Enhanced Factory Programming (BEFP) provides the fastest flash array programming performance with  $V_{PP}$  at 9.0 V, which increases factory throughput. With  $V_{PP}$  at 1.8 V,  $V_{CC}$  and  $V_{PP}$  can be tied together for a simple, ultra low power design. In addition to voltage flexibility, a dedicated  $V_{PP}$  connection provides complete data protection when  $V_{PP} \leq V_{PPLK}$ .

A Command User Interface (CUI) is the interface between the system processor and all internal operations of the device. An internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for block erase and program. A Status Register indicates erase or program completion and any errors that may have occurred.

An industry-standard command sequence invokes program and erase automation. Each erase operation erases one block. The Erase Suspend feature allows system software to pause an erase cycle to read or program data in another block. Program Suspend allows system software to pause programming to read other locations. Data is programmed in word increments (16 bits).

The P30 protection register allows unique flash device identification that can be used to increase system security. The individual Block Lock feature provides zero-latency block locking and unlocking. In addition, the P30 device also has four pre-defined spaces in the main array that can be configured as One-Time Programmable (OTP).

## 2.1 Virtual Chip Enable Description

The P30 512Mbit devices employ a Virtual Chip Enable which combines two 256-Mbit die with a common chip enable, F1-CE# for QUAD+ packages or CE# for Easy BGA and TSOP packages. (Refer to [Figure 9 on page 21](#) and [Figure 10 on page 21](#)). Address A24 (Quad+ package) or A25 (Easy BGA and TSOP packages) is then used to select between the die pair with F1-CE# / CE# asserted depending upon the package option used. When chip enable is asserted and QUAD+ A24 (Easy BGA/TSOP A25) is low ( $V_{IL}$ ), the lower parameter die is selected; when chip enable is asserted and QUAD+ A24 (Easy BGA/TSOP A25) is high ( $V_{IH}$ ), the upper parameter die is selected. Refer to [Table 1](#) and [Table 2](#) for additional details.

**Table 1: Virtual Chip Enable Truth Table for 512 Mb (QUAD+ Package)**

| Die Selected    | F1-CE# | A24 |
|-----------------|--------|-----|
| Lower Param Die | L      | L   |
| Upper Param Die | L      | H   |



**Table 2: Virtual Chip Enable Truth Table for 512 Mb (Easy BGA & TSOP Packages)**

| Die Selected    | CE# | A25 |
|-----------------|-----|-----|
| Lower Param Die | L   | L   |
| Upper Param Die | L   | H   |

### 3.0 Package Information

#### 3.1 56-Lead TSOP

Figure 1: TSOP Mechanical Specifications

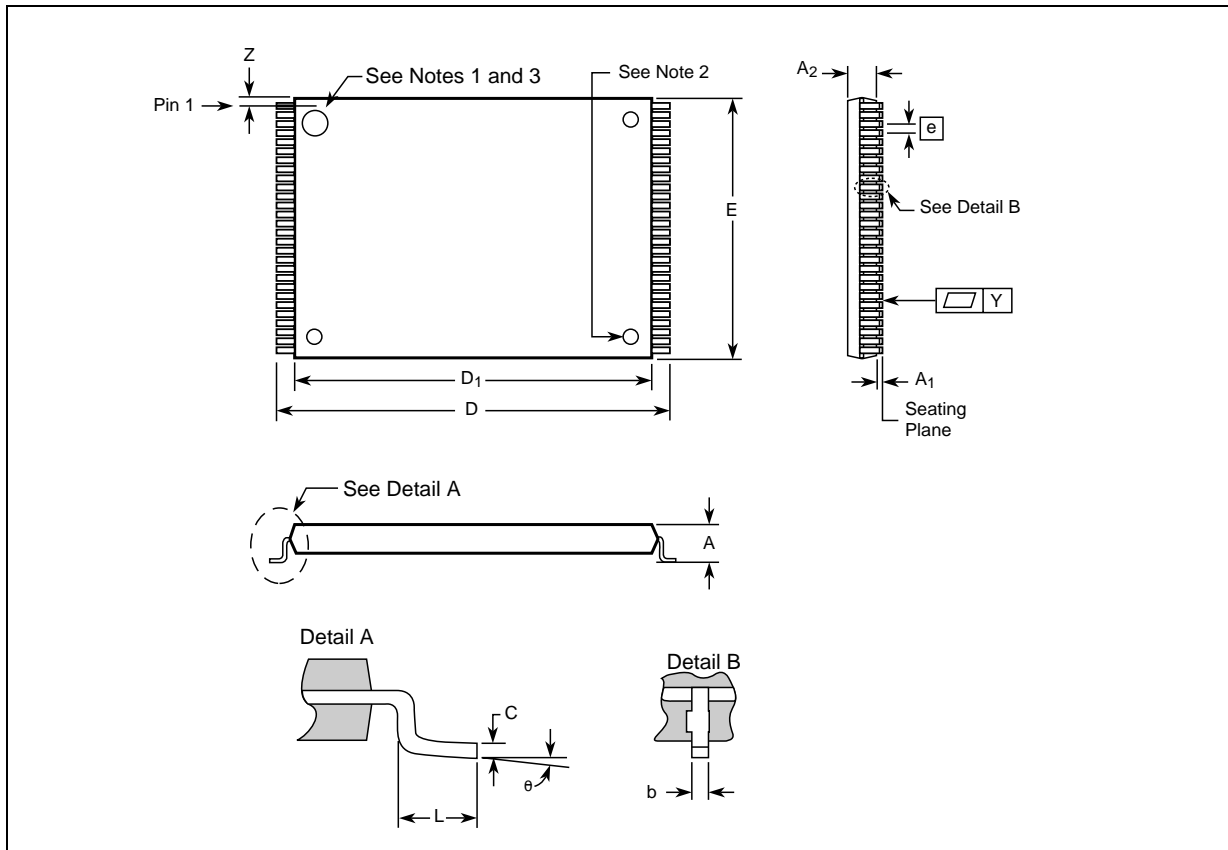


Table 3: TSOP Package Dimensions (Sheet 1 of 2)

| Product Information    | Symbol         | Millimeters |        |        | Inches |        |       | Notes |
|------------------------|----------------|-------------|--------|--------|--------|--------|-------|-------|
|                        |                | Min         | Nom    | Max    | Min    | Nom    | Max   |       |
| Package Height         | A              | -           | -      | 1.200  | -      | -      | 0.047 |       |
| Standoff               | A <sub>1</sub> | 0.050       | -      | -      | 0.002  | -      | -     |       |
| Package Body Thickness | A <sub>2</sub> | 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 |       |
| Lead Thickness         | c              | 0.100       | 0.150  | 0.200  | 0.004  | 0.006  | 0.008 |       |
| Package Body Length    | D <sub>1</sub> | 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 |       |

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

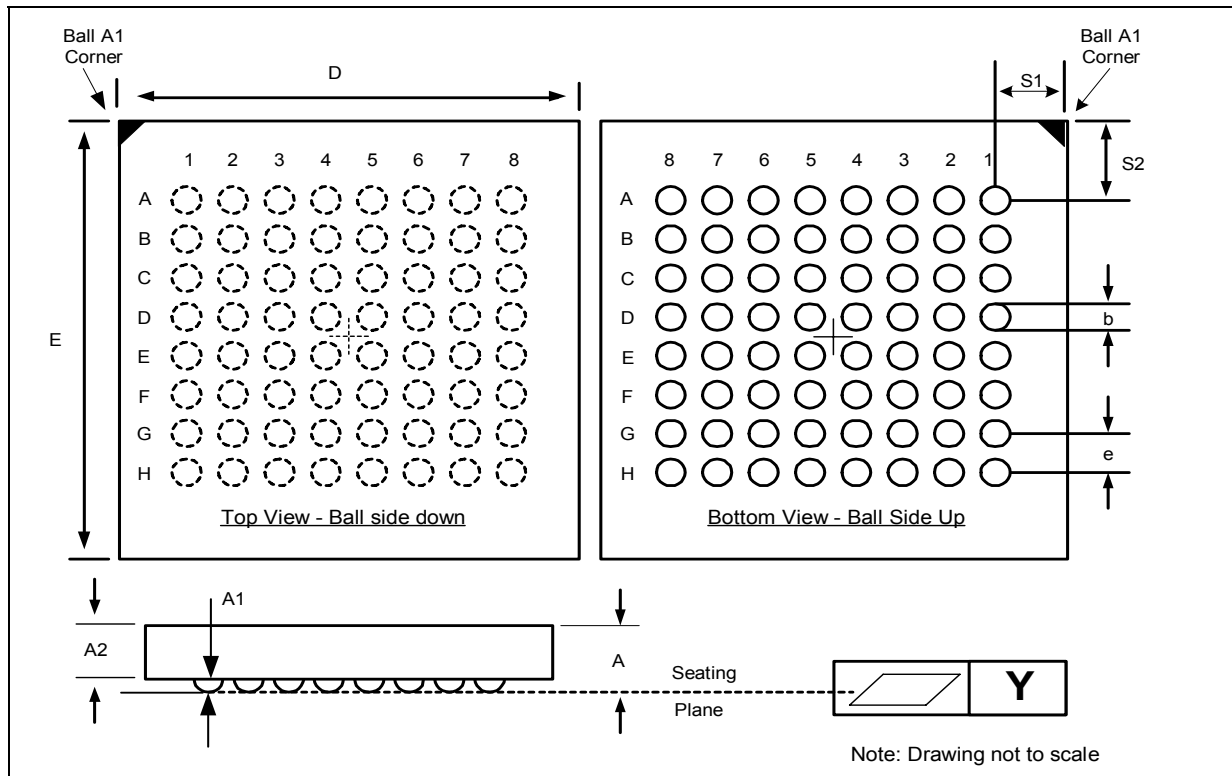
| Product Information       | Symbol | Millimeters |       |       | Inches |       |       | Notes |
|---------------------------|--------|-------------|-------|-------|--------|-------|-------|-------|
|                           |        | Min         | Nom   | Max   | Min    | Nom   | Max   |       |
| Lead Tip Length           | L      | 0.500       | 0.600 | 0.700 | 0.020  | 0.024 | 0.028 |       |
| Lead Count                | N      | -           | 56    | -     | -      | 56    | -     |       |
| Lead Tip Angle            | ý      | 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 |       |

**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.
4. Daisy Chain Evaluation Unit information is at Numonyx™ Flash Memory Packaging Technology <http://developer.Numonyx.com/design/flash/packtech>.

### 3.2 64-Ball Easy BGA Package

**Figure 2: Easy BGA Mechanical Specifications**



**Table 4: Easy BGA Package Dimensions**

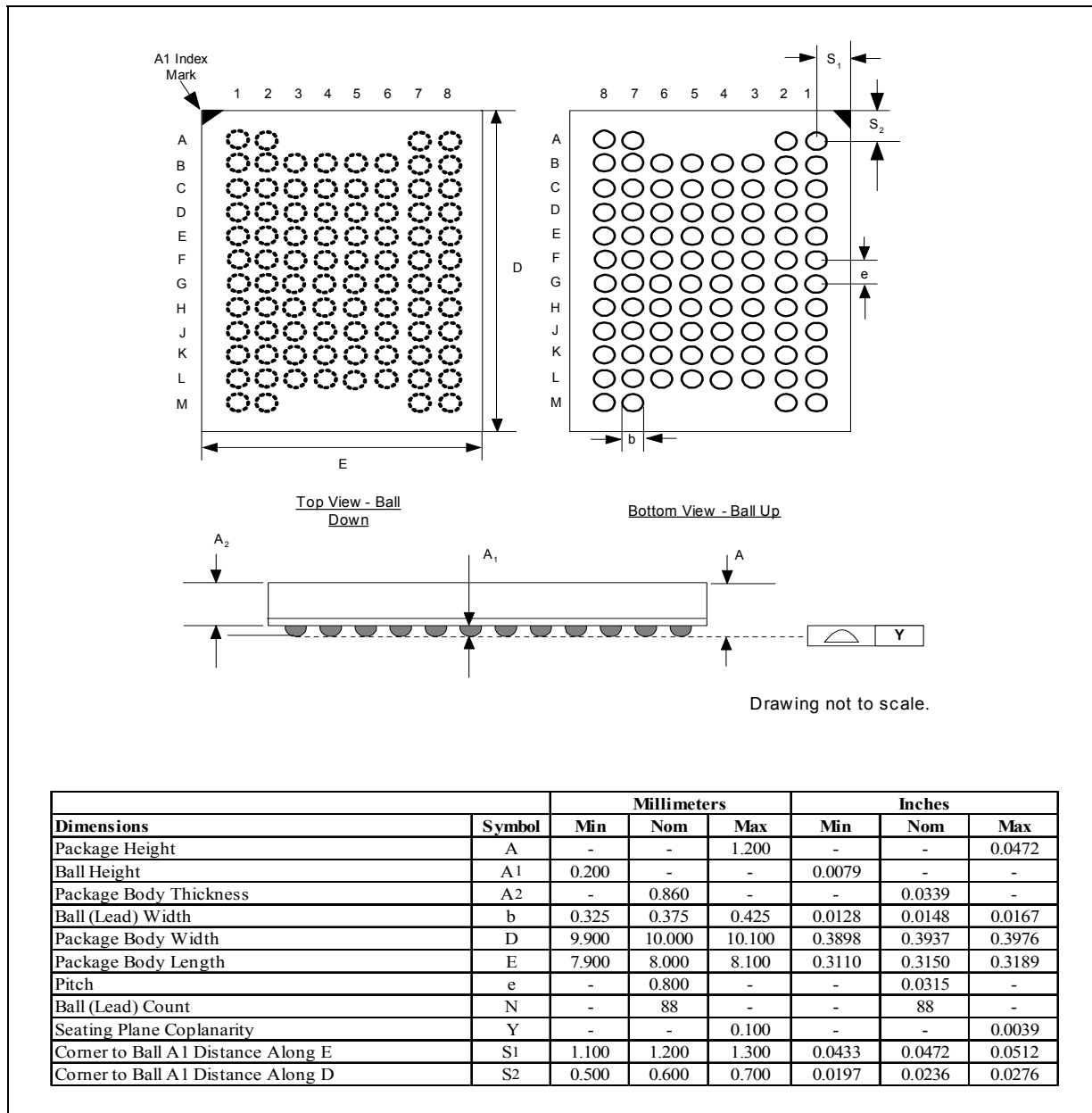
| Product Information                      | Symbol | Millimeters |        |        | Inches |        |        | Notes |
|--|--------|-------------|--------|--------|--------|--------|--------|-------|
|  |        | Min         | Nom    | Max    | Min    | Nom    | Max    |       |
| Package Height (64/128/256-Mbit)         | A      | -           | -      | 1.200  | -      | -      | 0.0472 |       |
| Package Height (512-Mbit)                | A      | -           | -      | 1.300  | -      | -      | 0.0512 |       |
| Ball Height                              | A1     | 0.250       | -      | -      | 0.0098 | -      | -      |       |
| Package Body Thickness (64/128/256-Mbit) | A2     | -           | 0.780  | -      | -      | 0.0307 | -      |       |
| Package Body Thickness (512-Mbit)        | A2     | -           | 0.910  | -      | -      | 0.0358 | -      |       |
| Ball (Lead) Width                        | b      | 0.330       | 0.430  | 0.530  | 0.0130 | 0.0169 | 0.0209 |       |
| Package Body Width                       | D      | 9.900       | 10.000 | 10.100 | 0.3898 | 0.3937 | 0.3976 |       |
| Package Body Length                      | E      | 12.900      | 13.000 | 13.100 | 0.5079 | 0.5118 | 0.5157 |       |
| Pitch                                    | [e]    | -           | 1.000  | -      | -      | 0.0394 | -      |       |
| Ball (Lead) Count                        | N      | -           | 64     | -      | -      | 64     | -      |       |
| Seating Plane Coplanarity                | Y      | -           | -      | 0.100  | -      | -      | 0.0039 |       |
| Corner to Ball A1 Distance Along D       | S1     | 1.400       | 1.500  | 1.600  | 0.0551 | 0.0591 | 0.0630 |       |
| Corner to Ball A1 Distance Along E       | S2     | 2.900       | 3.000  | 3.100  | 0.1142 | 0.1181 | 0.1220 |       |

**Notes:**

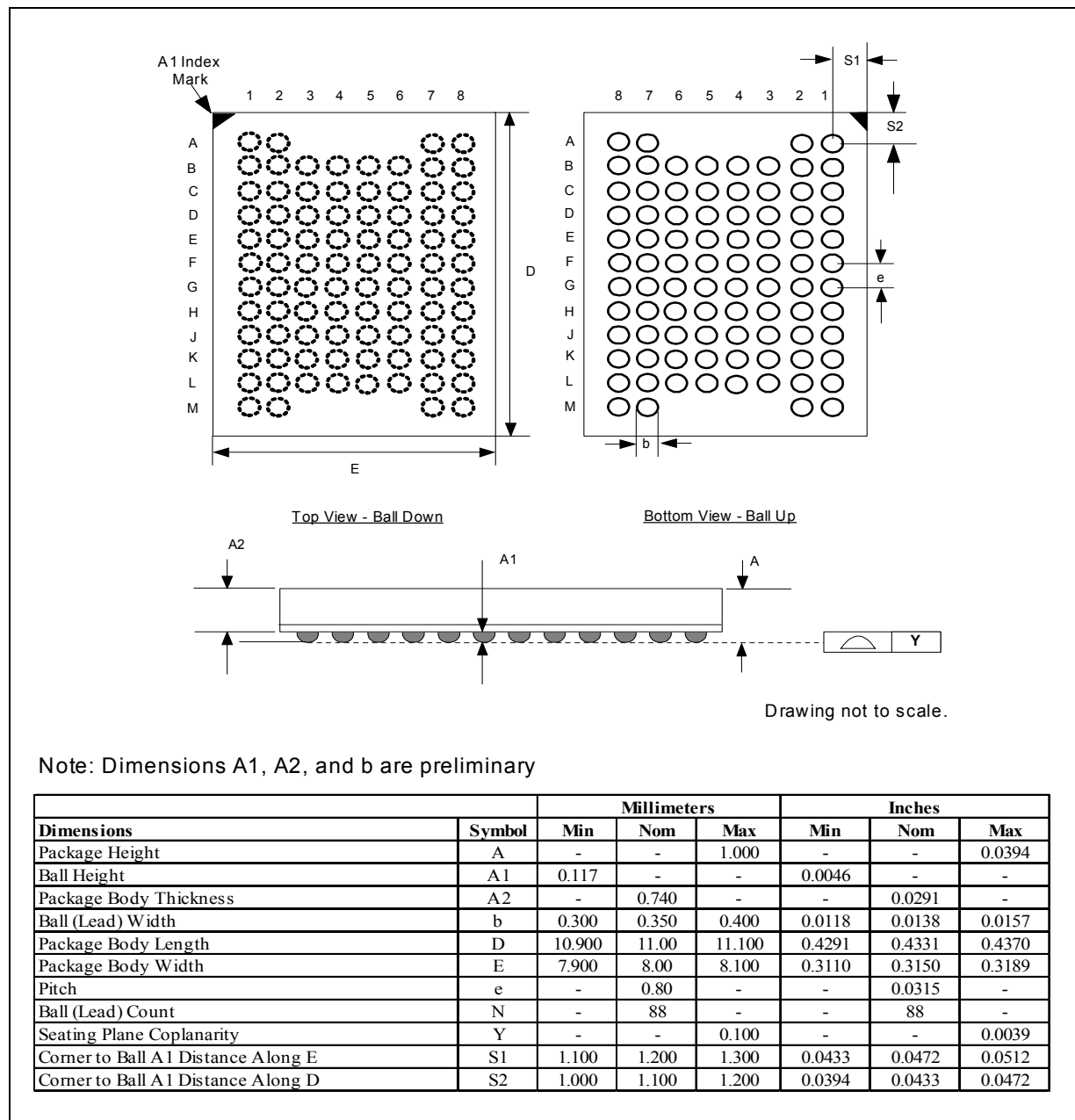
1. Daisy Chain Evaluation Unit information is at Numonyx™ Flash Memory Packaging Technology  
<http://developer.Numonyx.com/design/flash/packtech>.

### 3.3 QUAD+ SCSP Packages

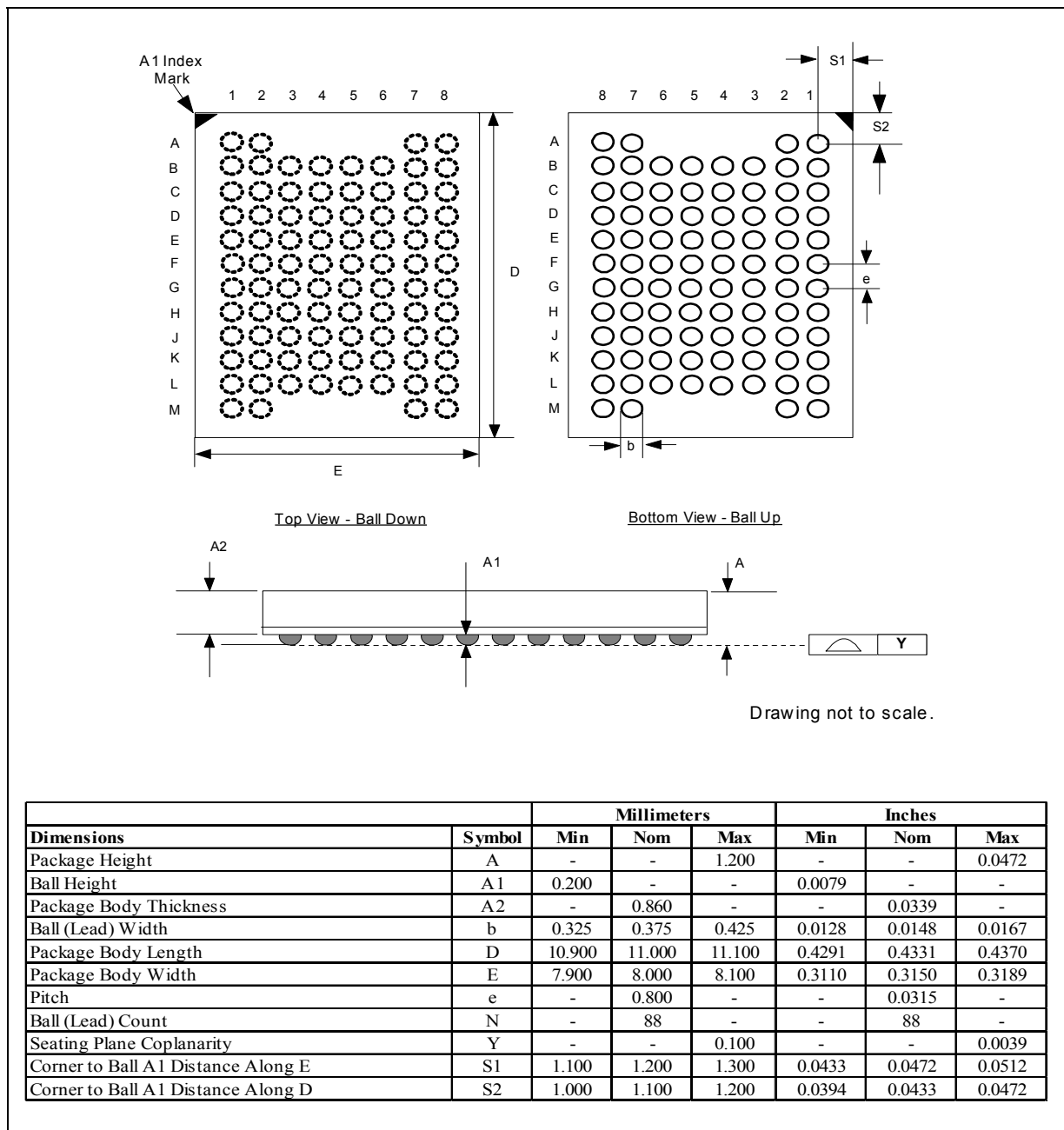
Figure 3: 64/128-Mbit, 88-ball (80 active) QUAD+ SCSP Specifications (8x10x1.2 mm)



**Figure 4: 256-Mbit, 88-ball (80 active) QUAD+ SCSP Specifications (8x11x1.0 mm)**



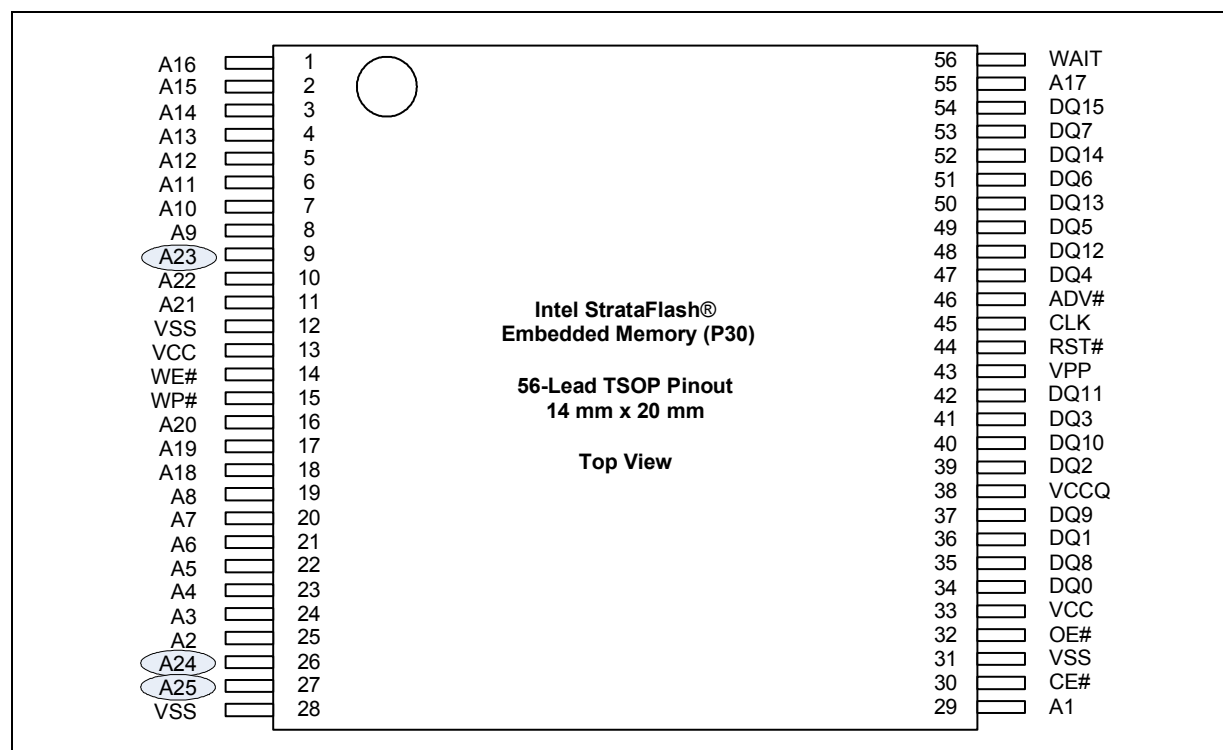
**Figure 5: 512-Mbit, 88-ball (80 active) QUAD+ SCSP Specifications (8x11x1.2 mm)**



## 4.0 Ballout and Signal Descriptions

### 4.1 Signal Ballout

**Figure 6: 56-Lead TSOP Pinout (64/128/256/512- Mbit)**

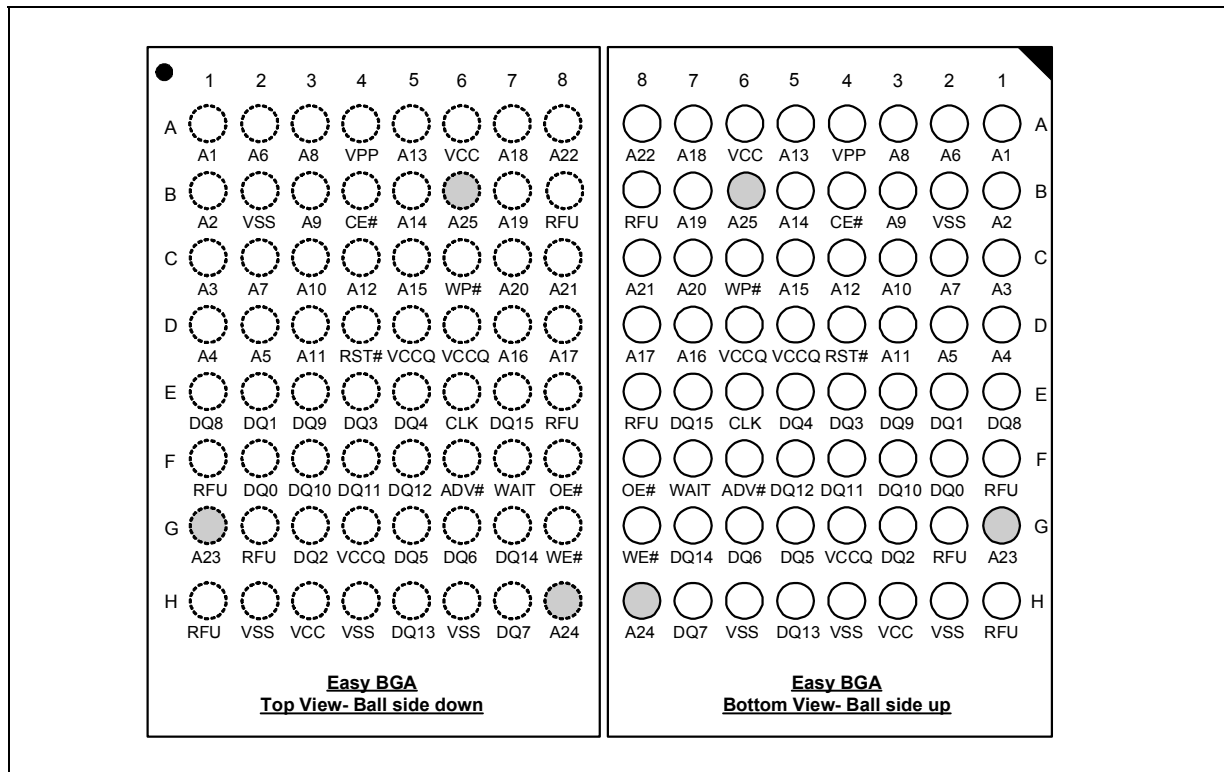


**Notes:**

1. A1 is the least significant address bit.
2. A23 is valid for 128-Mbit densities and above; otherwise, it is a no connect (NC).
3. A24 is valid for 256-Mbit densities; otherwise, it is a no connect (NC).
4. A25 is valid for 512-Mbit densities; otherwise, it is a no connect (NC).
5. Please refer to the latest specification update for synchronous read operation with the TSOP package. The synchronous read input signals (i.e. ADV# and CLK) should be tied off to support asynchronous reads. See [Section 4.2, "Signal Descriptions" on page 19](#).



**Figure 7: 64-Ball Easy BGA Ballout (64/128/256/512-Mbit)**



**Notes:**

1. A1 is the least significant address bit.
2. A23 is valid for 128-Mbit densities and above; otherwise, it is a no connect (NC).
3. A24 is valid for 256-Mbit densities and above; otherwise, it is a no connect (NC).
4. A25 is valid for 512-Mbit densities; otherwise, it is a no connect (NC).

Figure 8: 88-Ball (80-Active Ball) QUAD+ SCSP Ballout

|   | 1      | 2      | 3     | 4     | 5     | 6     | 7    | 8      |   |
|---|--------|--------|-------|-------|-------|-------|------|--------|---|
| A | DU     | DU     | Depop | Depop | Depop | Depop | DU   | DU     | A |
| B | A4     | A18    | A19   | VSS   | VCC   | VCC   | A21  | A11    | B |
| C | A5     | RFU    | A23   | VSS   | RFU   | CLK   | A22  | A12    | C |
| D | A3     | A17    | A24   | VPP   | RFU   | RFU   | A9   | A13    | D |
| E | A2     | A7     | RFU   | WP#   | ADV#  | A20   | A10  | A15    | E |
| F | A1     | A6     | RFU   | RST#  | WE#   | A8    | A14  | A16    | F |
| G | A0     | DQ8    | DQ2   | DQ10  | DQ5   | DQ13  | WAIT | F2-CE# | G |
| H | RFU    | DQ0    | DQ1   | DQ3   | DQ12  | DQ14  | DQ7  | F2-OE# | H |
| J | RFU    | F1-OE# | DQ9   | DQ11  | DQ4   | DQ6   | DQ15 | VCCQ   | J |
| K | F1-CE# | RFU    | RFU   | RFU   | RFU   | VCC   | VCCQ | RFU    | K |
| L | VSS    | VSS    | VCCQ  | VCC   | VSS   | VSS   | VSS  | VSS    | L |
| M | DU     | DU     | Depop | Depop | Depop | Depop | DU   | DU     | M |
|   | 1      | 2      | 3     | 4     | 5     | 6     | 7    | 8      |   |

**Notes:**

1. A22 is valid for 128-Mbit densities and above; otherwise, it is a no connect (NC).
2. A23 is valid for 256-Mbit densities and above; otherwise, it is a no connect (NC).
3. A24 is valid for 512-Mbit densities and above; otherwise, it is a no connect (NC).
4. F2-CE# and F2-OE# are no connect (NC) for all densities.

## 4.2 Signal Descriptions

This section has signal descriptions for the various P30 packages.

**Table 5: TSOP and Easy BGA Signal Descriptions (Sheet 1 of 2)**

| Symbol   | Type         | Name and Function   |
|----------|--------------|---|
| A[MAX:1] | Input        | <b>ADDRESS INPUTS:</b> Device address inputs. 64-Mbit: A[22:1]; 128-Mbit: A[23:1]; 256-Mbit: A[24:1]; 512-Mbit: A[25:1]. <i>Note:</i> The virtual selection of the 256-Mbit "Top parameter" die in the dual-die 512-Mbit configuration is accomplished by setting A[25] high ( $V_{IH}$ ).  |
| DQ[15:0] | Input/Output | <b>DATA INPUT/OUTPUTS:</b> Inputs data and commands during write cycles; outputs data during memory, Status Register, Protection Register, and Read Configuration Register reads. Data balls float when the CE# or OE# are deasserted. Data is internally latched during writes.  |
| ADV#     | Input        | <b>ADDRESS VALID:</b> Active low input. During synchronous read operations, addresses are latched on the rising edge of ADV#, or on the next valid CLK edge with ADV# low, whichever occurs first. In asynchronous mode, the address is latched when ADV# going high or continuously flows through if ADV# is held low.<br>WARNING: Designs not using ADV# must tie it to VSS to allow addresses to flow through.   |
| CE#      | Input        | <b>FLASH CHIP ENABLE:</b> Active low input. CE# low selects the associated flash memory die. When asserted, flash internal control logic, input buffers, decoders, and sense amplifiers are active. When deasserted, the associated flash die is deselected, power is reduced to standby levels, data and WAIT outputs are placed in high-Z state.<br>WARNING: Chip enable must be driven high when device is not in use.   |
| CLK      | Input        | <b>CLOCK:</b> Synchronizes the device with the system's bus frequency in synchronous-read mode. During synchronous read operations, addresses are latched on the rising edge of ADV#, or on the next valid CLK edge with ADV# low, whichever occurs first.<br>WARNING: Designs not using CLK for synchronous read mode must tie it to VCCQ or VSS.  |
| OE#      | Input        | <b>OUTPUT ENABLE:</b> Active low input. OE# low enables the device's output data buffers during read cycles. OE# high places the data outputs and WAIT in High-Z.   |
| RST#     | Input        | <b>RESET:</b> Active low input. RST# resets internal automation and inhibits write operations. This provides data protection during power transitions. RST# high enables normal operation. Exit from reset places the device in asynchronous read array mode.   |
| WAIT     | Output       | <b>WAIT:</b> Indicates data valid in synchronous array or non-array burst reads. Read Configuration Register bit 10 (RCR[10], WT) determines its polarity when asserted. WAIT's active output is $V_{OL}$ or $V_{OH}$ when CE# and OE# are $V_{IL}$ . WAIT is high-Z if CE# or OE# is $V_{IH}$ . <ul style="list-style-type: none"> <li>In synchronous array or non-array read modes, WAIT indicates invalid data when asserted and valid data when deasserted.</li> <li>In asynchronous page mode, and all write modes, WAIT is deasserted.</li> </ul>   |
| WE#      | Input        | <b>WRITE ENABLE:</b> Active low input. WE# controls writes to the device. Address and data are latched on the rising edge of WE#.   |
| WP#      | Input        | <b>WRITE PROTECT:</b> Active low input. WP# low enables the lock-down mechanism. Blocks in lock-down cannot be unlocked with the Unlock command. WP# high overrides the lock-down function enabling blocks to be erased or programmed using software commands.  |
| VPP      | Power/ Input | <b>Erase and Program Power:</b> A valid voltage on this pin allows erasing or programming. Memory contents cannot be altered when $V_{PP} \leq V_{PPLK}$ . Block erase and program at invalid $V_{PP}$ voltages should not be attempted.<br>Set $V_{PP} = V_{PPL}$ for in-system program and erase operations. To accommodate resistor or diode drops from the system supply, the $V_{IH}$ level of $V_{PP}$ can be as low as $V_{PPL}$ min. $V_{PP}$ must remain above $V_{PPL}$ min to perform in-system flash modification. $V_{PP}$ may be 0 V during read operations.<br>$V_{PPH}$ can be applied to main blocks for 1000 cycles maximum and to parameter blocks for 2500 cycles. $V_{PP}$ can be connected to 9 V for a cumulative total not to exceed 80 hours. Extended use of this pin at 9 V may reduce block cycling capability. |
| VCC      | Power        | <b>Device Core Power Supply:</b> Core (logic) source voltage. Writes to the flash array are inhibited when $V_{CC} \leq V_{LKO}$ . Operations at invalid $V_{CC}$ voltages should not be attempted.   |
| VCCQ     | Power        | <b>Output Power Supply:</b> Output-driver source voltage.   |
| VSS      | Power        | <b>Ground:</b> Connect to system ground. Do not float any VSS connection.   |

**Table 5: TSOP and Easy BGA Signal Descriptions (Sheet 2 of 2)**

| Symbol | Type | Name and Function   |
|--------|------|---|
| RFU    | —    | <b>Reserved for Future Use:</b> Reserved by Numonyx for future device functionality and enhancement. These should be treated in the same way as a Do Not Use (DU) signal. |
| DU     | —    | <b>Do Not Use:</b> Do not connect to any other signal, or power supply; must be left floating.  |
| NC     | —    | <b>No Connect:</b> No internal connection; can be driven or floated.  |

**Table 6: QUAD+ SCSP Signal Descriptions (Sheet 1 of 2)**

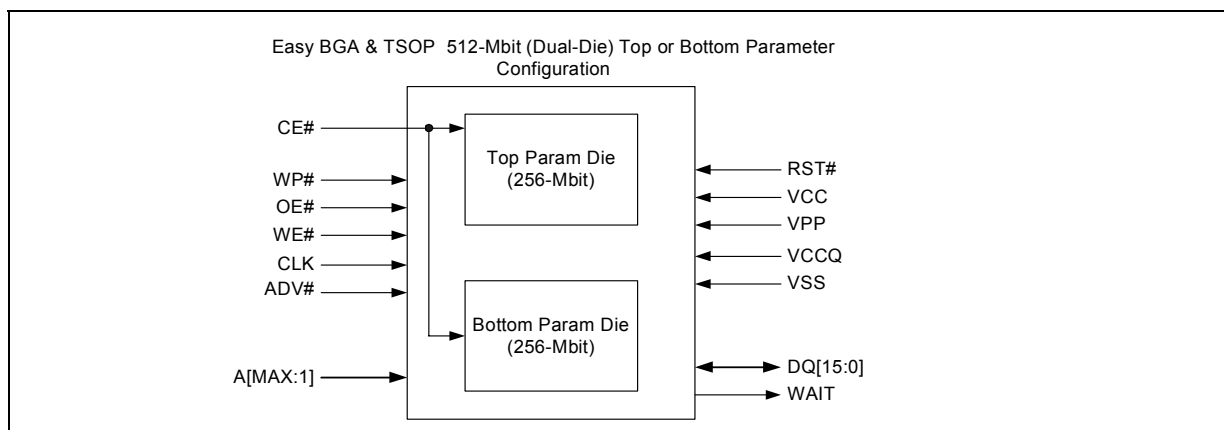
| Symbol   | Type         | Name and Function   |
|----------|--------------|---|
| A[MAX:0] | Input        | <b>ADDRESS INPUTS:</b> Device address inputs. 64-Mbit: A[21:0]; 128-Mbit: A[22:0]; 256-Mbit: A[23:0]; 512-Mbit: A[24:0]. <i>Note:</i> The virtual selection of the 256-Mbit "Top parameter" die in the dual-die 512-Mbit configuration is accomplished by setting A[25] high ( $V_{IH}$ ).  |
| DQ[15:0] | Input/Output | <b>DATA INPUT/OUTPUTS:</b> Inputs data and commands during write cycles; outputs data during memory, Status Register, Protection Register, and Read Configuration Register reads. Data balls float when the CE# or OE# are deasserted. Data is internally latched during writes.  |
| ADV#     | Input        | <b>ADDRESS VALID:</b> Active low input. During synchronous read operations, addresses are latched on the rising edge of ADV#, or on the next valid CLK edge with ADV# low, whichever occurs first. In asynchronous mode, the address is latched when ADV# going high or continuously flows through if ADV# is held low.<br>WARNING: Designs not using ADV# must tie it to VSS to allow addresses to flow through.   |
| F1-CE#   | Input        | <b>FLASH CHIP ENABLE:</b> Active low input. CE# low selects the associated flash memory die. When asserted, flash internal control logic, input buffers, decoders, and sense amplifiers are active. When deasserted, the associated flash die is deselected, power is reduced to standby levels, data and WAIT outputs are placed in high-Z state.<br>WARNING: Chip enable must be driven high when device is not in use.   |
| CLK      | Input        | <b>CLOCK:</b> Synchronizes the device with the system's bus frequency in synchronous-read mode. During synchronous read operations, addresses are latched on the rising edge of ADV#, or on the next valid CLK edge with ADV# low, whichever occurs first.<br>WARNING: Designs not using CLK for synchronous read mode must tie it to VCCQ or VSS.  |
| F1-OE#   | Input        | <b>OUTPUT ENABLE:</b> Active low input. OE# low enables the device's output data buffers during read cycles. OE# high places the data outputs and WAIT in High-Z.   |
| RST#     | Input        | <b>RESET:</b> Active low input. RST# resets internal automation and inhibits write operations. This provides data protection during power transitions. RST# high enables normal operation. Exit from reset places the device in asynchronous read array mode.   |
| WAIT     | Output       | <b>WAIT:</b> Indicates data valid in synchronous array or non-array burst reads. Read Configuration Register bit 10 (RCR[10], WT) determines its polarity when asserted. WAIT's active output is $V_{OL}$ or $V_{OH}$ when CE# and OE# are $V_{IL}$ . WAIT is high-Z if CE# or OE# is $V_{IH}$ . <ul style="list-style-type: none"> <li>In synchronous array or non-array read modes, WAIT indicates invalid data when asserted and valid data when deasserted.</li> <li>In asynchronous page mode, and all write modes, WAIT is deasserted.</li> </ul>   |
| WE#      | Input        | <b>WRITE ENABLE:</b> Active low input. WE# controls writes to the device. Address and data are latched on the rising edge of WE#.   |
| WP#      | Input        | <b>WRITE PROTECT:</b> Active low input. WP# low enables the lock-down mechanism. Blocks in lock-down cannot be unlocked with the Unlock command. WP# high overrides the lock-down function enabling blocks to be erased or programmed using software commands.  |
| VPP      | Power/ Input | <b>Erase and Program Power:</b> A valid voltage on this pin allows erasing or programming. Memory contents cannot be altered when $V_{pp} \leq V_{ppLK}$ . Block erase and program at invalid $V_{pp}$ voltages should not be attempted.<br>Set $V_{pp} = V_{ppL}$ for in-system program and erase operations. To accommodate resistor or diode drops from the system supply, the $V_{IH}$ level of $V_{pp}$ can be as low as $V_{ppL}$ min. $V_{pp}$ must remain above $V_{ppL}$ min to perform in-system flash modification. VPP may be 0 V during read operations.<br>$V_{ppH}$ can be applied to main blocks for 1000 cycles maximum and to parameter blocks for 2500 cycles. VPP can be connected to 9 V for a cumulative total not to exceed 80 hours. Extended use of this pin at 9 V may reduce block cycling capability. |

**Table 6: QUAD+ SCSP Signal Descriptions (Sheet 2 of 2)**

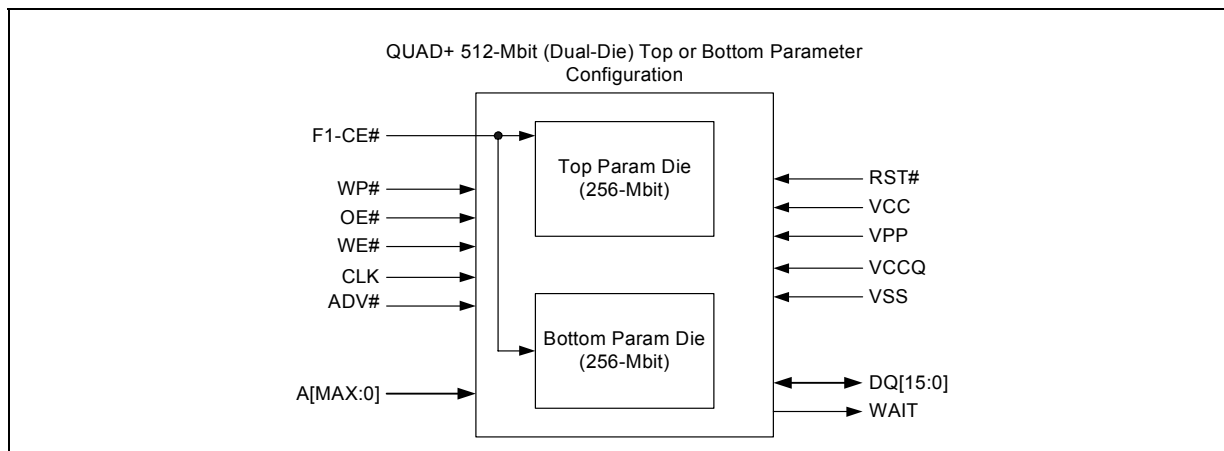
| Symbol | Type  | Name and Function   |
|--------|-------|---|
| VCC    | Power | <b>Device Core Power Supply:</b> Core (logic) source voltage. Writes to the flash array are inhibited when $V_{CC} \leq V_{LKO}$ . Operations at invalid $V_{CC}$ voltages should not be attempted. |
| VCCQ   | Power | <b>Output Power Supply:</b> Output-driver source voltage.   |
| VSS    | Power | <b>Ground:</b> Connect to system ground. Do not float any VSS connection.   |
| RFU    | —     | <b>Reserved for Future Use:</b> Reserved by Numonyx for future device functionality and enhancement. These should be treated in the same way as a Do Not Use (DU) signal.                           |
| DU     | —     | <b>Do Not Use:</b> Do not connect to any other signal, or power supply; must be left floating.  |
| NC     | —     | <b>No Connect:</b> No internal connection; can be driven or floated.  |

### 4.3 Dual-Die Configurations

**Figure 9: 512-Mbit Easy BGA and TSOP Top or Bottom Parameter Block Diagram**



**Figure 10: 512-Mbit QUAD+ SCSP Top or Bottom Parameter Block Diagram**



Note:  $A_{max} = V_{ih}$  selects the Top parameter Die;  $A_{max} = V_{il}$  selects the Bottom Parameter Die.

## 4.4 Memory Maps

Table 7 through Table 9 show the P30 memory maps. The memory array is divided into multiple 8-Mbit Programming Regions (see Section 11.0, "Programming Operations" on page 56).

**Table 7: Discrete Top Parameter Memory Maps (all packages)**

|                           | Size (KB) | Blk             | 64-Mbit         |
|---------------------------|-----------|-----------------|-----------------|
| One Programming Region    | 32        | 66              | 3FC000 - 3FFFFF |
|                           | ⋮         | ⋮               | ⋮               |
|                           | 32        | 63              | 3F0000 - 3F3FFF |
|                           | 128       | 62              | 3E0000 - 3EFFFF |
|                           | ⋮         | ⋮               | ⋮               |
| Seven Programming Regions | 128       | 56              | 380000 - 38FFFF |
|                           | 128       | 55              | 370000 - 37FFFF |
|                           | 128       | 54              | 360000 - 36FFFF |
|                           | ⋮         | ⋮               | ⋮               |
|                           | 128       | 1               | 010000 - 01FFFF |
| 128                       | 0         | 000000 - 00FFFF |                 |

|                             | Size (KB) | Blk             | 128-Mbit        |
|-----------------------------|-----------|-----------------|-----------------|
| One Programming Region      | 32        | 130             | 7FC000 - 7FFFFF |
|                             | ⋮         | ⋮               | ⋮               |
|                             | 32        | 127             | 7F0000 - 7F3FFF |
|                             | 128       | 126             | 7E0000 - 7EFFFF |
|                             | ⋮         | ⋮               | ⋮               |
| Fifteen Programming Regions | 128       | 120             | 780000 - 78FFFF |
|                             | 128       | 119             | 770000 - 77FFFF |
|                             | 128       | 118             | 760000 - 76FFFF |
|                             | ⋮         | ⋮               | ⋮               |
|                             | 128       | 1               | 010000 - 01FFFF |
| 128                         | 0         | 000000 - 00FFFF |                 |

|                                | Size (KB) | Blk             | 256-Mbit          |
|--------------------------------|-----------|-----------------|-------------------|
| One Programming Region         | 32        | 258             | FFC000 - FFFFFFFF |
|                                | ⋮         | ⋮               | ⋮                 |
|                                | 32        | 255             | FF0000 - FF3FFF   |
|                                | 128       | 254             | FE0000 - FEFFFF   |
|                                | ⋮         | ⋮               | ⋮                 |
| Thirty-One Programming Regions | 128       | 248             | F80000 - F8FFFF   |
|                                | 128       | 247             | F70000 - F7FFFF   |
|                                | 128       | 246             | F60000 - F6FFFF   |
|                                | ⋮         | ⋮               | ⋮                 |
|                                | 128       | 1               | 010000 - 01FFFF   |
| 128                            | 0         | 000000 - 00FFFF |                   |

**Table 8: Discrete Bottom Parameter Memory Maps (all packages)**

|                           | Size (KB) | Blk | 64-Mbit         |
|---------------------------|-----------|-----|-----------------|
| Seven Programming Regions | 128       | 66  | 3F0000 - 3FFFFF |
|                           | 128       | 65  | 3E0000 - 3EFFFF |
|                           | ⋮         | ⋮   | ⋮               |
|                           | 128       | 12  | 090000 - 09FFFF |
|                           | 128       | 11  | 080000 - 08FFFF |

|                             | Size (KB) | Blk | 128-Mbit        |
|-----------------------------|-----------|-----|-----------------|
| Fifteen Programming Regions | 128       | 130 | 7F0000 - 7FFFFF |
|                             | 128       | 129 | 7E0000 - 7EFFFF |
|                             | ⋮         | ⋮   | ⋮               |
|                             | 128       | 12  | 090000 - 09FFFF |
|                             | 128       | 11  | 080000 - 08FFFF |

**Table 8: Discrete Bottom Parameter Memory Maps (all packages)**

|                        | Size (KB) | Blk | 64-Mbit         |
|------------------------|-----------|-----|-----------------|
| One Programming Region | 128       | 10  | 070000 - 07FFFF |
|                        | ⋮         | ⋮   | ⋮               |
|                        | 128       | 4   | 010000 - 01FFFF |
|                        | 32        | 3   | 00C000 - 00FFFF |
|                        | ⋮         | ⋮   | ⋮               |
|                        | 32        | 0   | 000000 - 003FFF |

|                        | Size (KB) | Blk | 128-Mbit        |
|------------------------|-----------|-----|-----------------|
| One Programming Region | 128       | 10  | 070000 - 07FFFF |
|                        | ⋮         | ⋮   | ⋮               |
|                        | 128       | 4   | 010000 - 01FFFF |
|                        | 32        | 3   | 00C000 - 00FFFF |
|                        | ⋮         | ⋮   | ⋮               |
|                        | 32        | 0   | 000000 - 003FFF |

|                                | Size (KB) | Blk | 256-Mbit          |
|--------------------------------|-----------|-----|-------------------|
| Thirty-One Programming Regions | 128       | 258 | FF0000 - FFFFFFFF |
|                                | 128       | 257 | FE0000 - FFFFFFFF |
|                                | ⋮         | ⋮   | ⋮                 |
|                                | 128       | 12  | 090000 - 09FFFF   |
|                                | 128       | 11  | 080000 - 08FFFF   |
| One Programming Region         | 128       | 10  | 070000 - 07FFFF   |
|                                | ⋮         | ⋮   | ⋮                 |
|                                | 128       | 4   | 010000 - 01FFFF   |
|                                | 32        | 3   | 00C000 - 00FFFF   |
|                                | ⋮         | ⋮   | ⋮                 |
|                                | 32        | 0   | 000000 - 003FFF   |

Block size is referenced in K-Bytes where a byte=8 bits. Block Address range is referenced in K-Words where a Word is the size of the flash output bus (16 bits).

*Note:* The Dual- Die P30 memory maps are the same for both parameter options because the devices employ virtual chip enable (Refer to [Section 2.1](#)). The parameter option only defines the placement of bottom parameter die.

**Table 9: 512-Mbit Top and Bottom Parameter Memory Map (Easy BGA and QUAD+ SCSP) (Sheet 1 of 2)**

| 512-Mbit Flash (2x256-Mbit w/ 1CE) |           |     |                    |
|------------------------------------|-----------|-----|--------------------|
| Die Stack Config                   | Size (KB) | Blk | Address Range      |
| 256-Mbit<br>Top Parameter Die      | 32        | 517 | 1FFC000 - 1FFFFFFF |
|                                    | ⋮         | ⋮   | ⋮                  |
|                                    | 32        | 514 | 1FF0000 - 1FF3FFF  |
|                                    | 128       | 513 | 1FE0000 - 1FEFFFF  |
|                                    | ⋮         | ⋮   | ⋮                  |
|                                    | 128       | 259 | 1000000 - 100FFFF  |
|                                    | 128       | 258 | FF0000 - FFFFFFFF  |

**Table 9: 512-Mbit Top and Bottom Parameter Memory Map (Easy BGA and QUAD+ SCSP)  
(Sheet 2 of 2)**

| 512-Mbit Flash (2x256-Mbit w/ 1CE) |           |     |                 |
|------------------------------------|-----------|-----|-----------------|
| Die Stack Config                   | Size (KB) | Blk | Address Range   |
| 256-Mbit<br>Bottom Parameter Die   | ∴         | ∴   | ∴               |
|                                    | 128       | 4   | 010000 - 01FFFF |
|                                    | 32        | 3   | 00C000 - 00FFFF |
|                                    | ∴         | ∴   | ∴               |
|                                    | 32        | 0   | 000000 - 003FFF |

**Note:** Refer to the appropriate 256-Mbit Memory Map ([Table 7](#) or [Table 8](#)) for Programming Region information; Block size is referenced in K-Bytes where a byte=8 bits. Block Address range is referenced in K-Words where a Word is the size of the flash output bus (16 bits).



## 5.0 Maximum Ratings and Operating Conditions

### 5.1 Absolute Maximum Ratings

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

**Table 10:**

| Parameter   | Maximum Rating    | Notes |
|---|-------------------|-------|
| Temperature under bias  | -40 °C to +85 °C  |       |
| Storage temperature   | -65 °C to +125 °C |       |
| Voltage on any signal (except $V_{CC}$ , $V_{PP}$ and $V_{CCQ}$ ) | -0.5 V to +4.1 V  | 1     |
| VPP voltage   | -0.2 V to +10 V   | 1,2,3 |
| VCC voltage   | -0.2 V to +2.5 V  | 1     |
| VCCQ voltage  | -0.2 V to +4.1 V  | 1     |
| Output short circuit current                                      | 100 mA            | 4     |

**Notes:**

1. Voltages shown are specified with respect to  $V_{SS}$ . Minimum DC voltage is -0.5 V on input/output signals and -0.2 V on  $V_{CC}$ ,  $V_{CCQ}$ , and  $V_{PP}$ . During transitions, this level may undershoot to -2.0 V for periods less than 20 ns. Maximum DC voltage on  $V_{CC}$  is  $V_{CC} + 0.5$  V, which, during transitions, may overshoot to  $V_{CC} + 2.0$  V for periods less than 20 ns. Maximum DC voltage on input/output signals and  $V_{CCQ}$  is  $V_{CCQ} + 0.5$  V, which, during transitions, may overshoot to  $V_{CCQ} + 2.0$  V for periods less than 20 ns.
2. Maximum DC voltage on  $V_{PP}$  may overshoot to +11.5 V for periods less than 20 ns.
3. Program/erase voltage is typically 1.7 V - 2.0 V. 9.0 V can be applied for 80 hours maximum total, to any blocks for 1000 cycles maximum. 9.0 V program/erase voltage may reduce block cycling capability.
4. Output shorted for no more than one second. No more than one output shorted at a time.

### 5.2 Operating Conditions

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

**Table 11: Operating Conditions**

| Symbol             | Parameter                             |                    | Min     | Max  | Units  | Notes |
|--------------------|---------------------------------------|--------------------|---------|------|--------|-------|
| $T_C$              | Operating Temperature                 |                    | -40     | +85  | °C     | 1     |
| $V_{CC}$           | $V_{CC}$ Supply Voltage               |                    | 1.7     | 2.0  | V      | 3     |
| $V_{CCQ}$          | I/O Supply Voltage                    | CMOS inputs        | 1.7     | 3.6  |        | 2     |
|                    |                                       | TTL inputs         | 2.4     | 3.6  |        |       |
| $V_{PPL}$          | $V_{PP}$ Voltage Supply (Logic Level) |                    | 0.9     | 3.6  | Hours  | 2     |
| $V_{PPH}$          | Factory word programming $V_{PP}$     |                    | 8.5     | 9.5  |        |       |
| $t_{PPH}$          | Maximum $V_{PP}$ Hours                | $V_{PP} = V_{PPH}$ | -       | 80   |        |       |
| Block Erase Cycles | Main and Parameter Blocks             | $V_{PP} = V_{PPL}$ | 100,000 | -    | Cycles | 2     |
|                    | Main Blocks                           | $V_{PP} = V_{PPH}$ | -       | 1000 |        |       |
|                    | Parameter Blocks                      | $V_{PP} = V_{PPH}$ | -       | 2500 |        |       |

**Notes:**

1.  $T_C$  = Case Temperature.
2. In typical operation VPP program voltage is  $V_{PPL}$ .
3. 40Mhz burst operation on the TSOP package has a min  $V_{CC}$  value of 1.85V. Please refer to the latest Specification Update regarding synchronous burst operation with the TSOP package

## 6.0 Electrical Specifications

### 6.1 DC Current Characteristics

Table 12: DC Current Characteristics (Sheet 1 of 2)

| Sym                                       | Parameter  |  | CMOS Inputs<br>( $V_{CCQ} = 1.7\text{ V} - 3.6\text{ V}$ ) |         | TTL Inputs<br>( $V_{CCQ} = 2.4\text{ V} - 3.6\text{ V}$ ) |          | Unit          | Test Conditions  | Notes  |       |
|---|--|--|--|---------|---|----------|---------------|--|--|-------|
|   |  |  | Typ  | Max     | Typ   | Max      |               |  |  |       |
| $I_{LI}$                                  | Input Load Current   |  | -  | $\pm 1$ | -   | $\pm 2$  | $\mu\text{A}$ | $V_{CC} = V_{CCMax}$<br>$V_{CCQ} = V_{CCQMax}$<br>$V_{IN} = V_{CCQ}$ or $V_{SS}$ | 1  |       |
| $I_{LO}$                                  | Output Leakage Current   | DQ[15:0], WAIT   | -  | $\pm 1$ | -   | $\pm 10$ | $\mu\text{A}$ | $V_{CC} = V_{CCMax}$<br>$V_{CCQ} = V_{CCQMax}$<br>$V_{IN} = V_{CCQ}$ or $V_{SS}$ |  |       |
| $I_{CCS}$ ,<br>$I_{CCD}$                  | $V_{CC}$ Standby,<br>Power Down  |  | 64-Mbit  | 20      | 35  | 20       | 35            | $\mu\text{A}$  | $V_{CC} = V_{CCMax}$<br>$V_{CCQ} = V_{CCQMax}$<br>$CE\# = V_{CCQ}$<br>$RST\# = V_{CCQ}$ (for $I_{CCS}$ )<br>$RST\# = V_{SS}$ (for $I_{CCD}$ )<br>$WP\# = V_{IH}$ | 1,2   |
| 128-Mbit                                  |  |  | 30   | 75      | 30  | 75       |               |  |  |       |
| 256-Mbit                                  |  |  | 55   | 115     | 55  | 200      |               |  |  |       |
| 512-Mbit                                  |  |  | 110  | 230     | 110   | 400      |               |  |  |       |
| $I_{CCR}$                                 | Average<br>$V_{CC}$<br>Read<br>Current   | Asynchronous Single-Word<br>$f = 5\text{ MHz}$ (1 CLK) | 14   | 16      | 14  | 16       | mA            | 1-Word Read  | $V_{CC} = V_{CCMax}$<br>$CE\# = V_{IL}$<br>$OE\# = V_{IH}$<br>Inputs: $V_{IL}$ or $V_{IH}$   | 1     |
|   |  | Page-Mode Read<br>$f = 13\text{ MHz}$ (5 CLK)          | 9  | 10      | 9   | 10       | mA            | 4-Word Read  |  |       |
|   |  | Synchronous Burst<br>$f = 40\text{ MHz}$               | 13   | 17      | n/a   | n/a      | mA            | BL = 4W  |  |       |
|   |  |  | 15   | 19      | n/a   | n/a      | mA            | BL = 8W  |  |       |
|   |  |  | 17   | 21      | n/a   | n/a      | mA            | BL = 16W   |  |       |
|   |  |  | 21   | 26      | n/a   | n/a      | mA            | BL = Cont.   |  |       |
|   |  | Synchronous Burst<br>$f = 52\text{ MHz}$               | 16   | 19      | n/a   | n/a      | mA            | BL = 4W  |  |       |
|   |  |  | 19   | 23      | n/a   | n/a      | mA            | BL = 8W  |  |       |
|   |  |  | 22   | 26      | n/a   | n/a      | mA            | BL = 16W   |  |       |
|   |  |  | 23   | 28      | n/a   | n/a      | mA            | BL = Cont.   |  |       |
| $I_{CCW}$ ,<br>$I_{CCE}$                  | $V_{CC}$ Program Current,<br>$V_{CC}$ Erase Current  |  | 36   | 51      | 36  | 51       | mA            | $V_{PP} = V_{PPL}$ , pgm/ers in progress   | 1,3,5  |       |
|   |  |  | 26   | 33      | 26  | 33       |               | $V_{PP} = V_{PPH}$ , pgm/ers in progress   |  |       |
| $I_{CCWS}$ ,<br>$I_{CCES}$                | $V_{CC}$ Program Suspend Current,<br>$V_{CC}$ Erase Suspend Current                              |  | 64-Mbit  | 20      | 35  | 20       | 35            | $\mu\text{A}$  | $CE\# = V_{CCQ}$ ; suspend in progress   | 1,3,4 |
|   |  |  | 128-Mbit   | 30      | 75  | 30       | 75            |  |  |       |
|   |  |  | 256-Mbit   | 55      | 115   | 55       | 200           |  |  |       |
|   |  |  | 512-Mbit   | 110     | 230   | 110      | 400           |  |  |       |
| $I_{PPS}$ ,<br>$I_{PPWS}$ ,<br>$I_{PPES}$ | $V_{PP}$ Standby Current,<br>$V_{PP}$ Program Suspend Current,<br>$V_{PP}$ Erase Suspend Current |  | 0.2  | 5       | 0.2   | 5        | $\mu\text{A}$ | $V_{PP} = V_{PPL}$ , suspend in progress   | 1,3  |       |
| $I_{PPR}$                                 | $V_{PP}$ Read  |  | 2  | 15      | 2   | 15       | $\mu\text{A}$ | $V_{PP} = V_{PPL}$   | 1,3  |       |
| $I_{PPW}$                                 | $V_{PP}$ Program Current   |  | 0.05   | 0.10    | 0.05  | 0.10     | mA            | $V_{PP} = V_{PPL}$ , program in progress   |  |       |
|   |  |  | 8  | 22      | 8   | 22       |               | $V_{PP} = V_{PPH}$ , program in progress   |  |       |

**Table 12: DC Current Characteristics (Sheet 2 of 2)**

| Sym              | Parameter                     | CMOS Inputs<br>( $V_{CCQ} = 1.7\text{ V} - 3.6\text{ V}$ ) |      | TTL Inputs<br>( $V_{CCQ} = 2.4\text{ V} - 3.6\text{ V}$ ) |      | Unit | Test Conditions   | Notes |
|------------------|-------------------------------|--|------|---|------|------|---|-------|
|                  |                               | Typ  | Max  | Typ   | Max  |      |   |       |
| I <sub>PPE</sub> | V <sub>PP</sub> Erase Current | 0.05   | 0.10 | 0.05  | 0.10 | mA   | V <sub>PP</sub> = V <sub>PP,L</sub> , erase in progress |       |
|                  |                               | 8  | 22   | 8   | 22   |      | V <sub>PP</sub> = V <sub>PP,H</sub> , erase in progress |       |

**Notes:**

1. All currents are RMS unless noted. Typical values at typical  $V_{CC}$ ,  $T_C = +25\text{ }^\circ\text{C}$ .
2.  $I_{CCS}$  is the average current measured over any 5 ms time interval 5  $\mu\text{s}$  after CE# is deasserted.
3. Sampled, not 100% tested.
4.  $I_{CCES}$  is specified with the device deselected. If device is read while in erase suspend, current is  $I_{CCES}$  plus  $I_{CCR}$ .
5.  $I_{CCW}$ ,  $I_{CCE}$  measured over typical or max times specified in [Section 7.5, "Program and Erase Characteristics"](#) on page 39.

## 6.2 DC Voltage Characteristics

**Table 13: DC Voltage Characteristics**

| Sym                | Parameter                        | CMOS Inputs<br>( $V_{CCQ} = 1.7\text{ V} - 3.6\text{ V}$ ) |           | TTL Inputs <sup>(1)</sup><br>( $V_{CCQ} = 2.4\text{ V} - 3.6\text{ V}$ ) |           | Unit | Test Condition   | Notes |
|--------------------|----------------------------------|--|-----------|--|-----------|------|--|-------|
|                    |                                  | Min  | Max       | Min  | Max       |      |  |       |
| V <sub>IL</sub>    | Input Low Voltage                | 0  | 0.4       | 0  | 0.6       | V    |  | 2     |
| V <sub>IH</sub>    | Input High Voltage               | $V_{CCQ} - 0.4\text{ V}$                                   | $V_{CCQ}$ | 2.0  | $V_{CCQ}$ | V    |  |       |
| V <sub>OL</sub>    | Output Low Voltage               | -  | 0.1       | -  | 0.1       | V    | $V_{CC} = V_{CCMin}$<br>$V_{CCQ} = V_{CCQMin}$<br>$I_{OL} = 100\text{ }\mu\text{A}$  |       |
| V <sub>OH</sub>    | Output High Voltage              | $V_{CCQ} - 0.1$  | -         | $V_{CCQ} - 0.1$  | -         | V    | $V_{CC} = V_{CCMin}$<br>$V_{CCQ} = V_{CCQMin}$<br>$I_{OH} = -100\text{ }\mu\text{A}$ |       |
| V <sub>PP,LK</sub> | V <sub>PP</sub> Lock-Out Voltage | -  | 0.4       | -  | 0.4       | V    |  | 3     |
| V <sub>LKO</sub>   | V <sub>CC</sub> Lock Voltage     | 1.0  | -         | 1.0  | -         | V    |  |       |
| V <sub>LKOQ</sub>  | V <sub>CCQ</sub> Lock Voltage    | 0.9  | -         | 0.9  | -         | V    |  |       |

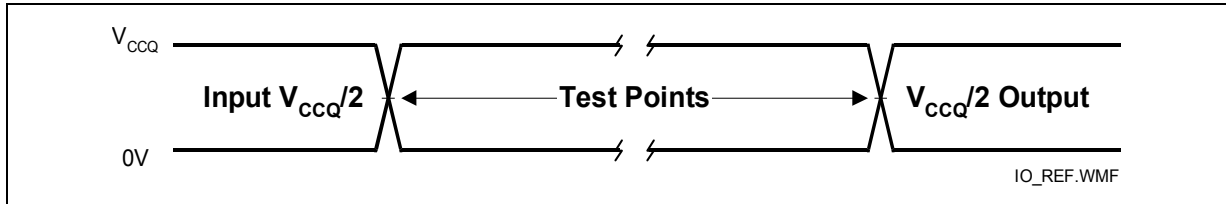
**Notes:**

1. Synchronous read mode is not supported with TTL inputs.
2. V<sub>IL</sub> can undershoot to -0.4 V and V<sub>IH</sub> can overshoot to  $V_{CCQ} + 0.4\text{ V}$  for durations of 20 ns or less.
3.  $V_{PP} \leq V_{PP,LK}$  inhibits erase and program operations. Do not use V<sub>PP,L</sub> and V<sub>PP,H</sub> outside their valid ranges.

## 7.0 AC Characteristics

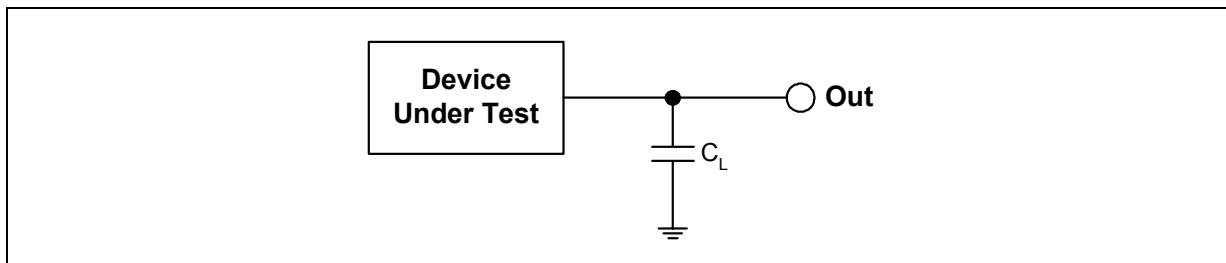
### 7.1 AC Test Conditions

**Figure 11: AC Input/Output Reference Waveform**



**Note:** AC test inputs are driven at  $V_{CCQ}$  for Logic "1" and 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  $V_{CC} = V_{CCMin}$ .

**Figure 12: Transient Equivalent Testing Load Circuit**



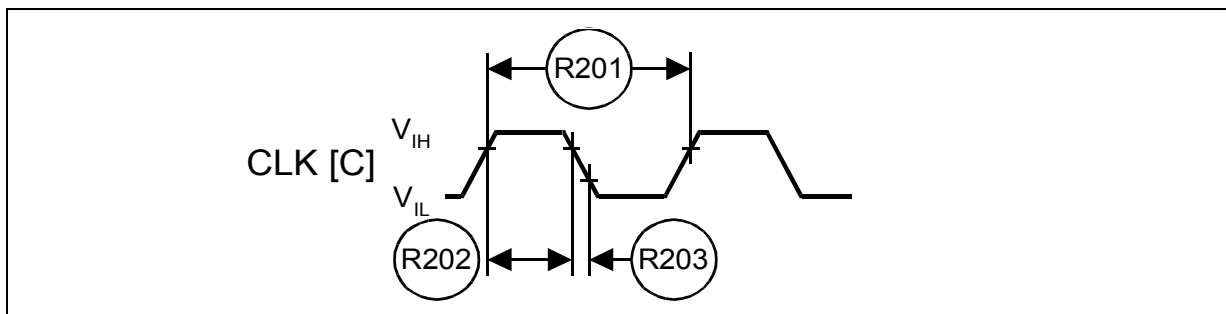
**Notes:**

1. See the following table for component values.
2. Test configuration component value for worst-case speed conditions.
3.  $C_L$  includes jig capacitance.

**Table 14: Test Configuration Component Value For Worst Case Speed Conditions**

| Test Configuration         | $C_L$ (pF) |
|----------------------------|------------|
| $V_{CCQMin}$ Standard Test | 30         |

**Figure 13: Clock Input AC Waveform**



## 7.2 Capacitance

**Table 15: Capacitance**

| Parameter          | Signals  | Min | Typ | Max | Unit | Condition   | Notes |
|--------------------|--|-----|-----|-----|------|---|-------|
| Input Capacitance  | Address, Data, CE#, WE#, OE#, RST#, CLK, ADV#, WP# | 2   | 6   | 7   | pF   | Typ temp = 25 °C,<br>Max temp = 85 °C,<br>V <sub>CC</sub> = (0 V - 2.0 V),<br>V <sub>CCQ</sub> = (0 V - 3.6 V),<br>Discrete silicon die | 1,2,3 |
| Output Capacitance | Data, WAIT   | 2   | 4   | 5   | pF   |   |       |

**Notes:**

1. Capacitance values are for a single die; for 2-die and 4-die stacks, multiply the capacitance values by the number of dies in the stack.
2. Sampled, but not 100% tested.
3. Silicon die capacitance only; add 1 pF for discrete packages.

## 7.3 AC Read Specifications

**Table 16: AC Read Specifications for 64/128- Mbit Densities (Sheet 1 of 2)**

| Num                                | Symbol            | Parameter  | Min | Max | Unit | Notes |
|------------------------------------|-------------------|--|-----|-----|------|-------|
| <b>Asynchronous Specifications</b> |                   |  |     |     |      |       |
| R1                                 | t <sub>AVAV</sub> | Read cycle time  | 85  | -   | ns   |       |
| R2                                 | t <sub>AVQV</sub> | Address to output valid                                      | -   | 85  | ns   |       |
| R3                                 | t <sub>ELQV</sub> | CE# low to output valid                                      | -   | 85  | ns   |       |
| R4                                 | t <sub>GLQV</sub> | OE# low to output valid                                      | -   | 25  | ns   | 1,2   |
| R5                                 | t <sub>PHQV</sub> | RST# high to output valid                                    | -   | 150 | ns   | 1     |
| R6                                 | t <sub>ELQX</sub> | CE# low to output in low-Z                                   | 0   | -   | ns   | 1,3   |
| R7                                 | t <sub>GLQX</sub> | OE# low to output in low-Z                                   | 0   | -   | ns   | 1,2,3 |
| R8                                 | t <sub>EHQZ</sub> | CE# high to output in high-Z                                 | -   | 24  | ns   | 1,3   |
| R9                                 | t <sub>GHQZ</sub> | OE# high to output in high-Z                                 | -   | 24  | ns   |       |
| R10                                | t <sub>OH</sub>   | Output hold from first occurring address, CE#, or OE# change | 0   | -   | ns   |       |
| R11                                | t <sub>EHEL</sub> | CE# pulse width high   | 20  | -   | ns   | 1     |
| R12                                | t <sub>ELTV</sub> | CE# low to WAIT valid  | -   | 17  | ns   |       |
| R13                                | t <sub>EHTZ</sub> | CE# high to WAIT high-Z                                      | -   | 20  | ns   | 1,3   |
| R15                                | t <sub>GLTV</sub> | OE# low to WAIT valid  | -   | 17  | ns   | 1     |
| R16                                | t <sub>GLTX</sub> | OE# low to WAIT in low-Z                                     | 0   | -   | ns   | 1,3   |
| R17                                | t <sub>GHTZ</sub> | OE# high to WAIT in high-Z                                   | -   | 20  | ns   |       |
| <b>Latching Specifications</b>     |                   |  |     |     |      |       |
| R101                               | t <sub>AVVH</sub> | Address setup to ADV# high                                   | 10  | -   | ns   | 1     |
| R102                               | t <sub>ELVH</sub> | CE# low to ADV# high   | 10  | -   | ns   |       |
| R103                               | t <sub>VLQV</sub> | ADV# low to output valid                                     | -   | 85  | ns   |       |
| R104                               | t <sub>VLVH</sub> | ADV# pulse width low   | 10  | -   | ns   |       |
| R105                               | t <sub>VHVL</sub> | ADV# pulse width high  | 10  | -   | ns   |       |
| R106                               | t <sub>VHAX</sub> | Address hold from ADV# high                                  | 9   | -   | ns   |       |

**Table 16: AC Read Specifications for 64/128- Mbit Densities (Sheet 2 of 2)**

| Num   | Symbol                | Parameter               | Min  | Max | Unit | Notes   |
|---|-----------------------|-------------------------|------|-----|------|---------|
| R108  | $t_{APA}$             | Page address access     | -    | 25  | ns   | 1       |
| R111  | $t_{phvh}$            | RST# high to ADV# high  | 30   | -   | ns   |         |
| <b>Clock Specifications</b>                       |                       |                         |      |     |      |         |
| R200  | $f_{CLK}$             | CLK frequency           | -    | 52  | MHz  | 1,3,5,6 |
|   |                       | TSOP                    | -    | 40  | MHz  |         |
| R201  | $t_{CLK}$             | CLK period              | 19.2 | -   | ns   |         |
|   |                       | TSOP                    | 25   | -   | ns   |         |
| R202  | $t_{CH/CL}$           | CLK high/low time       | 5    | -   | ns   |         |
| R203  | $t_{FCLK/RCLK}$       | CLK fall/rise time      | -    | 3   | ns   |         |
| <b>Synchronous Specifications<sup>(5,6)</sup></b> |                       |                         |      |     |      |         |
| R301  | $t_{AVCH/L}$          | Address setup to CLK    | 9    | -   | ns   | 1       |
| R302  | $t_{VLCH/L}$          | ADV# low setup to CLK   | 9    | -   | ns   |         |
| R303  | $t_{ELCH/L}$          | CE# low setup to CLK    | 9    | -   | ns   |         |
| R304  | $t_{CHQV} / t_{CLQV}$ | CLK to output valid     | -    | 17  | ns   |         |
| R305  | $t_{CHQX}$            | Output hold from CLK    | 3    | -   | ns   | 1,7     |
| R306  | $t_{CHAX}$            | Address hold from CLK   | 10   | -   | ns   | 1,4,7   |
| R307  | $t_{CHTV}$            | CLK to WAIT valid       | -    | 17  | ns   | 1,7     |
| R311  | $t_{CHVL}$            | CLK Valid to ADV# Setup | 3    | -   | ns   | 1       |
| R312  | $t_{CHTX}$            | WAIT Hold from CLK      | 3    | -   | ns   | 1,7     |

**Notes:**

- See Figure 11, "AC Input/Output Reference Waveform" on page 28 for timing measurements and max allowable input slew rate.
- OE# may be delayed by up to  $t_{ELQV} - t_{GLQV}$  after CE#'s falling edge without impact to  $t_{ELQV}$ .
- Sampled, not 100% tested.
- Address hold in synchronous burst mode is  $t_{CHAX}$  or  $t_{VHAX}$ , whichever timing specification is satisfied first.
- Please see the latest P30 Spec Update for synchronous burst operation with the TSOP package.
- Synchronous read mode is not supported with TTL level inputs.
- Applies only to subsequent synchronous reads.

**Table 17: AC Read Specifications for 256/512-Mbit Densities (Sheet 1 of 3)**

| Num                                | Symbol     | Parameter               | Speed                    | Min | Max | Unit | Notes |
|------------------------------------|------------|-------------------------|--------------------------|-----|-----|------|-------|
| <b>Asynchronous Specifications</b> |            |                         |                          |     |     |      |       |
| R1                                 | $t_{AVAV}$ | Read cycle time         | $V_{CC} = 1.8V - 2.0$    | 85  | -   | ns   |       |
|                                    |            |                         | $V_{CC} = 1.7V - 2.0$    | 88  | -   |      |       |
|                                    |            |                         | 256/512-Mb TSOP packages | 95  |     |      |       |
| R2                                 | $t_{AVQV}$ | Address to output valid | $V_{CC} = 1.8V - 2.0$    | -   | 85  | ns   |       |
|                                    |            |                         | $V_{CC} = 1.7V - 2.0$    | -   | 88  |      |       |
|                                    |            |                         | 256/512-Mb TSOP packages | -   | 95  |      |       |

**Table 17: AC Read Specifications for 256/512-Mbit Densities (Sheet 2 of 3)**

| Num   | Symbol          | Parameter  | Speed                                  | Min  | Max | Unit | Notes   |
|---|-----------------|--|--|------|-----|------|---------|
| R3  | $t_{ELQV}$      | CE# low to output valid                                      | $V_{CC} = 1.8\text{ V} - 2.0\text{ V}$ | -    | 85  | ns   |         |
|   |                 |  | $V_{CC} = 1.7\text{ V} - 2.0\text{ V}$ | -    | 88  |      |         |
|   |                 |  | 256/512-Mb TSOP packages               | -    | 95  |      |         |
| R4  | $t_{GLQV}$      | OE# low to output valid                                      |  | -    | 25  | ns   | 1,2     |
| R5  | $t_{PHQV}$      | RST# high to output valid                                    |  | -    | 150 | ns   | 1       |
| R6  | $t_{ELQX}$      | CE# low to output in low-Z                                   |  | 0    | -   | ns   | 1,3     |
| R7  | $t_{GLQX}$      | OE# low to output in low-Z                                   |  | 0    | -   | ns   | 1,2,3   |
| R8  | $t_{EHQZ}$      | CE# high to output in high-Z                                 |  | -    | 24  | ns   | 1,3     |
| R9  | $t_{GHQZ}$      | OE# high to output in high-Z                                 |  | -    | 24  | ns   |         |
| R10   | $t_{OH}$        | Output hold from first occurring address, CE#, or OE# change |  | 0    | -   | ns   |         |
| R11   | $t_{EHEL}$      | CE# pulse width high   |  | 20   | -   | ns   | 1       |
| R12   | $t_{ELTV}$      | CE# low to WAIT valid  |  | -    | 17  | ns   |         |
| R13   | $t_{EHTZ}$      | CE# high to WAIT high-Z                                      |  | -    | 20  | ns   | 1,3     |
| R15   | $t_{GLTV}$      | OE# low to WAIT valid  |  | -    | 17  | ns   | 1       |
| R16   | $t_{GLTX}$      | OE# low to WAIT in low-Z                                     |  | 0    | -   | ns   | 1,3     |
| R17   | $t_{GHTZ}$      | OE# high to WAIT in high-Z                                   |  | -    | 20  | ns   |         |
| <b>Latching Specifications</b>                    |                 |  |  |      |     |      |         |
| R101  | $t_{AVVH}$      | Address setup to ADV# high                                   |  | 10   | -   | ns   | 1       |
| R102  | $t_{ELVH}$      | CE# low to ADV# high   |  | 10   | -   | ns   |         |
| R103  | $t_{VLQV}$      | ADV# low to output valid                                     | $V_{CC} = 1.8\text{ V} - 2.0\text{ V}$ | -    | 85  | ns   |         |
|   |                 |  | $V_{CC} = 1.7\text{ V} - 2.0\text{ V}$ | -    | 88  |      |         |
|   |                 |  | 256/512-Mb TSOP packages               | -    | 95  |      |         |
| R104  | $t_{VLVH}$      | ADV# pulse width low   |  | 10   | -   | ns   |         |
| R105  | $t_{VHVL}$      | ADV# pulse width high  |  | 10   | -   | ns   |         |
| R106  | $t_{VHAX}$      | Address hold from ADV# high                                  |  | 9    | -   | ns   | 1,4     |
| R108  | $t_{APA}$       | Page address access  |  | -    | 25  | ns   | 1       |
| R111  | $t_{phvh}$      | RST# high to ADV# high                                       |  | 30   | -   | ns   |         |
| <b>Clock Specifications</b>                       |                 |  |  |      |     |      |         |
| R200  | $f_{CLK}$       | CLK frequency  |  | -    | 52  | MHz  | 1,3,5,6 |
|   |                 |  | TSOP Package                           | -    | 40  | MHz  |         |
| R201  | $t_{CLK}$       | CLK period   |  | 19.2 | -   | ns   |         |
|   |                 |  | TSOP Package                           | 25   | -   | ns   |         |
| R202  | $t_{CH/CL}$     | CLK high/low time  |  | 5    | -   | ns   |         |
| R203  | $t_{FCLK/RCLK}$ | CLK fall/rise time   |  | -    | 3   | ns   |         |
| <b>Synchronous Specifications<sup>(5,6)</sup></b> |                 |  |  |      |     |      |         |

**Table 17: AC Read Specifications for 256/512-Mbit Densities (Sheet 3 of 3)**

| Num  | Symbol                | Parameter               | Speed | Min | Max | Unit | Notes |
|------|-----------------------|-------------------------|-------|-----|-----|------|-------|
| R301 | $t_{AVCH/L}$          | Address setup to CLK    |       | 9   | -   | ns   | 1     |
| R302 | $t_{VLCH/L}$          | ADV# low setup to CLK   |       | 9   | -   | ns   |       |
| R303 | $t_{ELCH/L}$          | CE# low setup to CLK    |       | 9   | -   | ns   |       |
| R304 | $t_{CHQV} / t_{CLQV}$ | CLK to output valid     |       | -   | 17  | ns   |       |
| R305 | $t_{CHQX}$            | Output hold from CLK    |       | 3   | -   | ns   | 1,7   |
| R306 | $t_{CHAX}$            | Address hold from CLK   |       | 10  | -   | ns   | 1,4,7 |
| R307 | $t_{CHTV}$            | CLK to WAIT valid       |       | -   | 17  | ns   | 1,7   |
| R311 | $t_{CHVL}$            | CLK Valid to ADV# Setup |       | 3   | -   | ns   | 1     |
| R312 | $t_{CHTX}$            | WAIT Hold from CLK      |       | 3   | -   | ns   | 1,7   |

**Notes:**

1. See [Figure 11, "AC Input/Output Reference Waveform" on page 28](#) for timing measurements and max 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. Address hold in synchronous burst mode is  $t_{CHAX}$  or  $t_{VHAX}$ , whichever timing specification is satisfied first.
5. Please see the latest P30 Spec Update for synchronous burst operation with the TSOP package.
6. Synchronous read mode is not supported with TTL level inputs.
7. Applies only to subsequent synchronous reads.

**Table 18: AC Read Specification differences for 65nm**

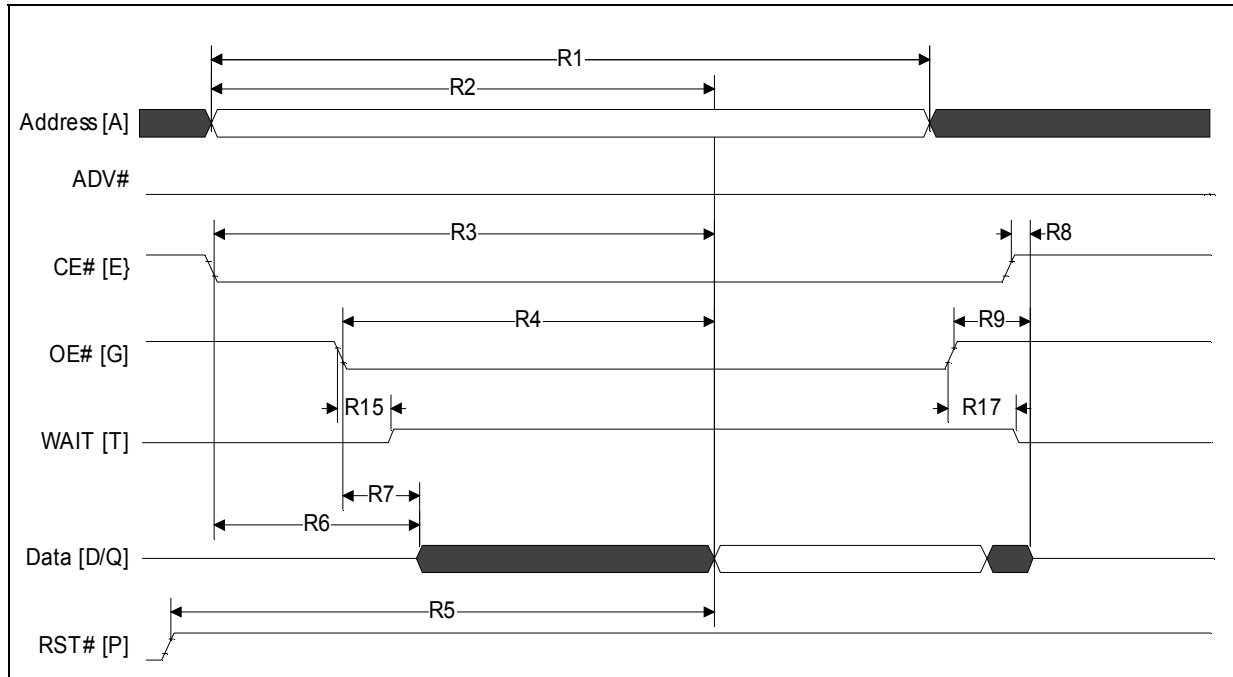
| Num                                | Symbol     | Parameter                              | Min  | Max | Unit | Notes |     |
|------------------------------------|------------|--|------|-----|------|-------|-----|
| <b>Asynchronous Specifications</b> |            |  |      |     |      |       |     |
| R1                                 | $t_{AVAV}$ | Read cycle time                        |      | 100 | -    | ns    | 2   |
|                                    |            |  | TSOP | 110 |      | ns    | 2   |
| R2                                 | $t_{AVQV}$ | Address to output valid                |      | -   | 100  | ns    | 2   |
|                                    |            |  | TSOP |     | 110  | ns    | 2   |
| R3                                 | $t_{ELQV}$ | CE# low to output valid                |      | -   | 100  | ns    | 2   |
|                                    |            |  | TSOP |     | 110  | ns    | 2   |
| R103                               |            | $t_{VLQV}$<br>ADV# low to output valid |      | -   | 100  | ns    | 1,2 |
|                                    |            |  | TSOP |     | 110  | ns    | 2   |

**Notes:**

1. See [Figure 11, "AC Input/Output Reference Waveform" on page 28](#) for timing measurements and max allowable input slew rate.
2. This is the recommended specification for all new designs supporting both 130nm and 65nm lithos, or for new designs that will use the 65nm lithography.

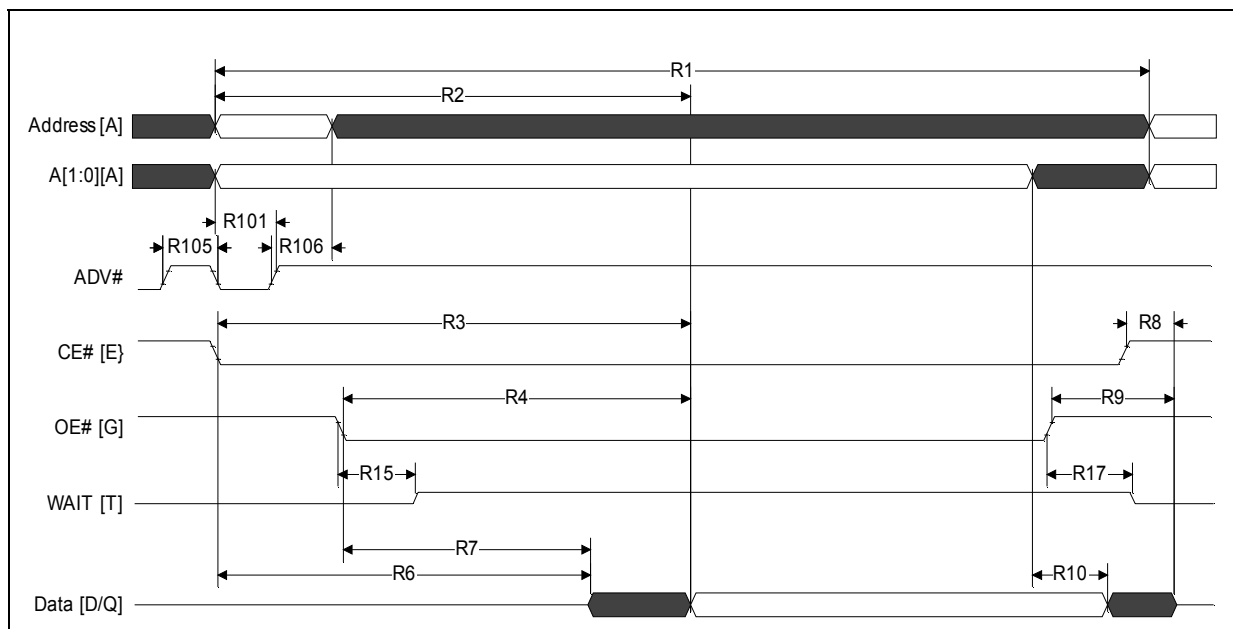


**Figure 14: Asynchronous Single-Word Read (ADV# Low)**



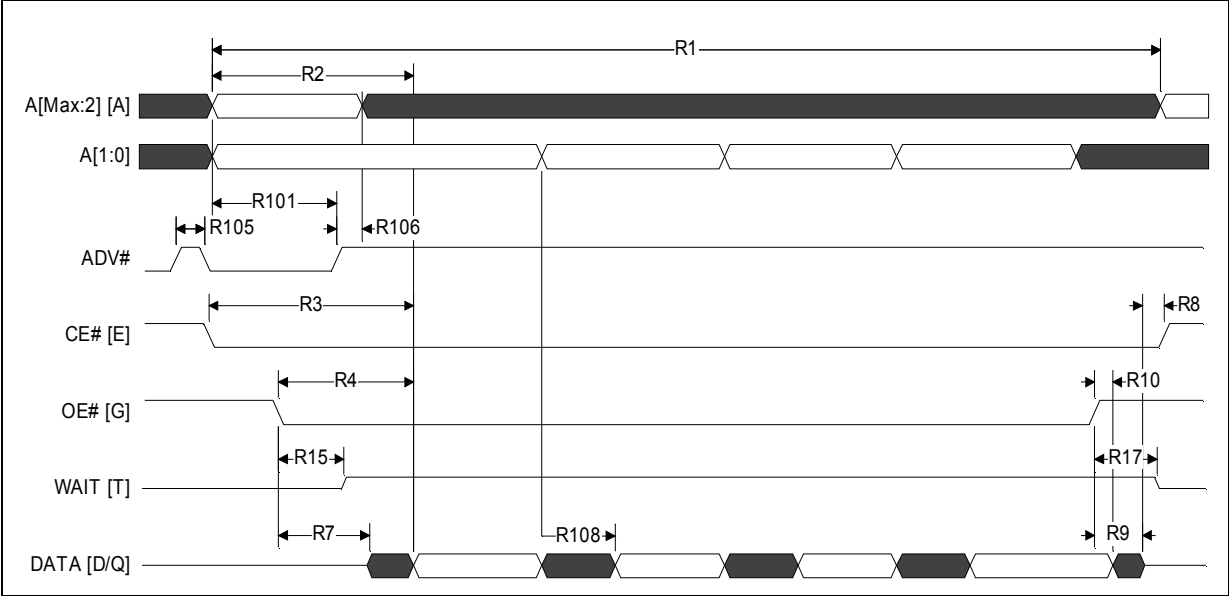
**Note:** WAIT shown deasserted during asynchronous read mode (RCR[10]=0, Wait asserted low).

**Figure 15: Asynchronous Single-Word Read (ADV# Latch)**



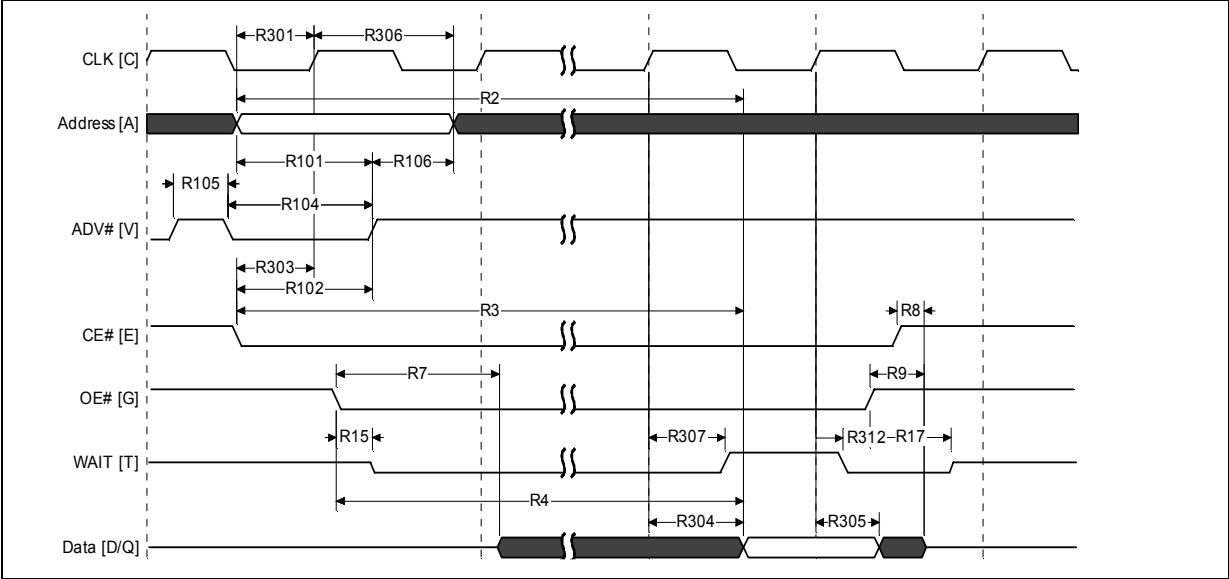
**Note:** WAIT shown deasserted during asynchronous read mode (RCR[10]=0, Wait asserted low).

Figure 16: Asynchronous Page-Mode Read Timing



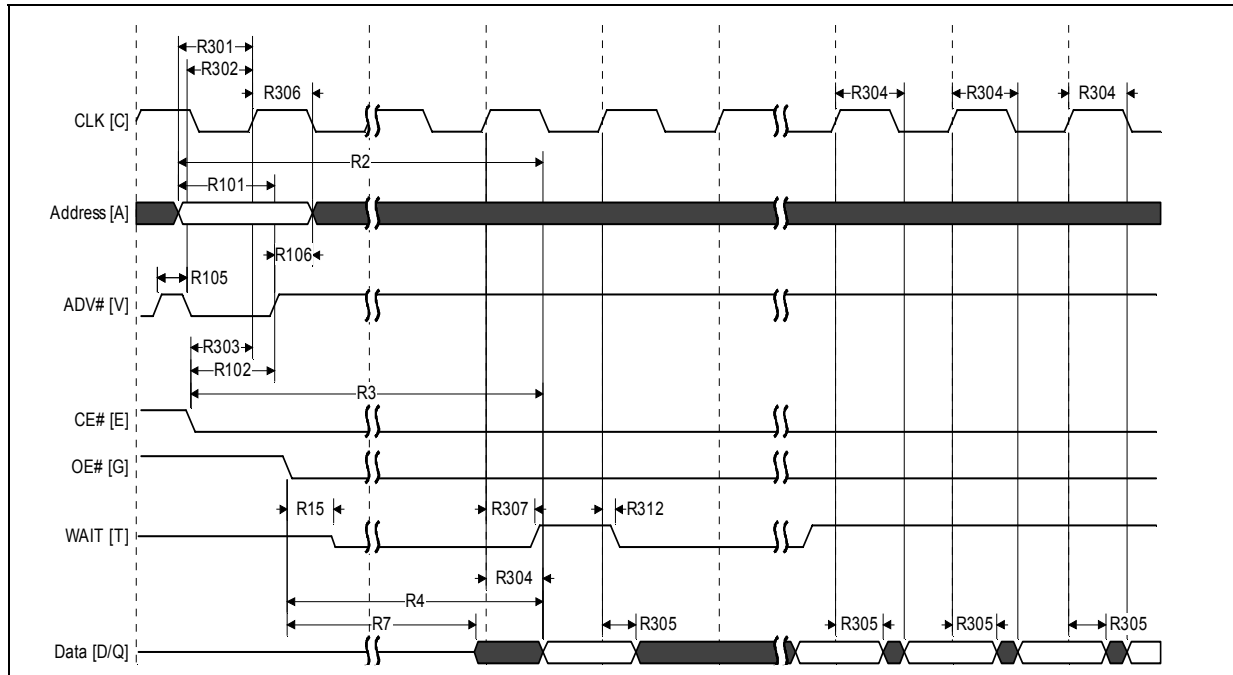
Note: WAIT shown deasserted during asynchronous read mode (RCR[10]=0, Wait asserted low).

Figure 17: Synchronous Single-Word Array or Non-array Read Timing



1. WAIT is driven per OE# assertion during synchronous array or non-array read, and can be configured to assert either during or one data cycle before valid data.
2. This diagram illustrates the case in which an n-word burst is initiated to the flash memory array and it is terminated by CE# deassertion after the first word in the burst.

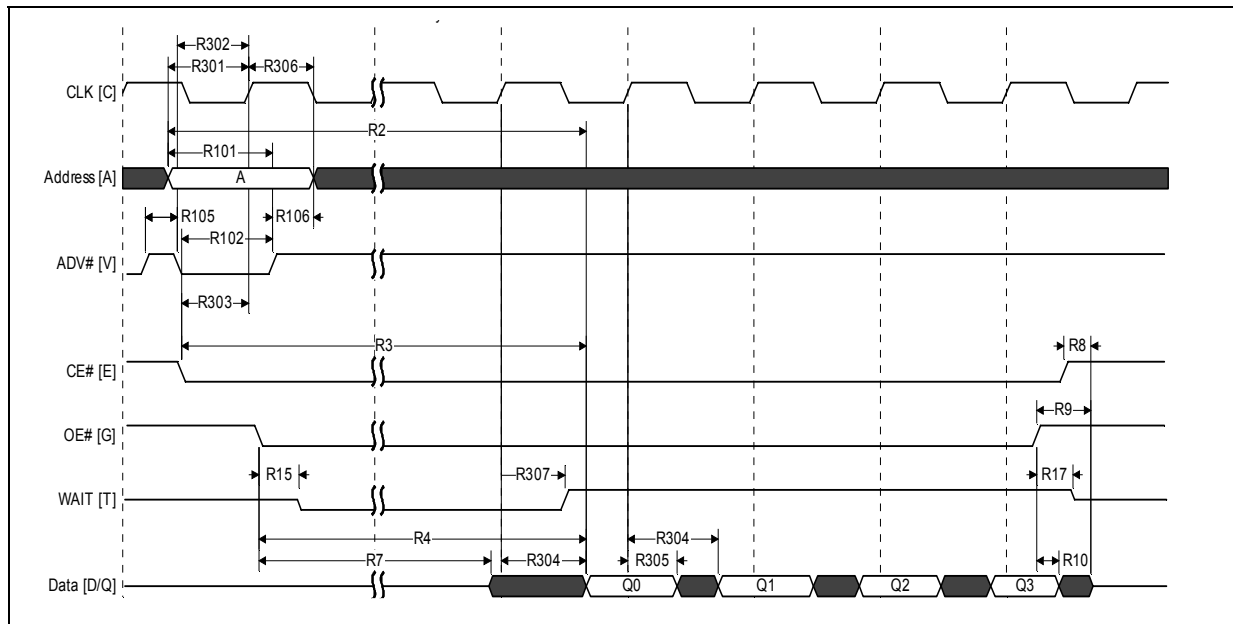
**Figure 18: Continuous Burst Read, Showing An Output Delay Timing**



**Notes:**

1. WAIT is driven per OE# assertion during synchronous array or non-array read, and can be configured to assert either during or one data cycle before valid data.
2. At the end of Word Line; the delay incurred when a burst access crosses a 16-word boundary and the starting address is not 4-word boundary aligned.

**Figure 19: Synchronous Burst-Mode Four-Word Read Timing**



**Note:** WAIT is driven per OE# assertion during synchronous array or non-array read. WAIT asserted during initial latency and deasserted during valid data (RCR[10] = 0, Wait asserted low).

## 7.4 AC Write Specifications

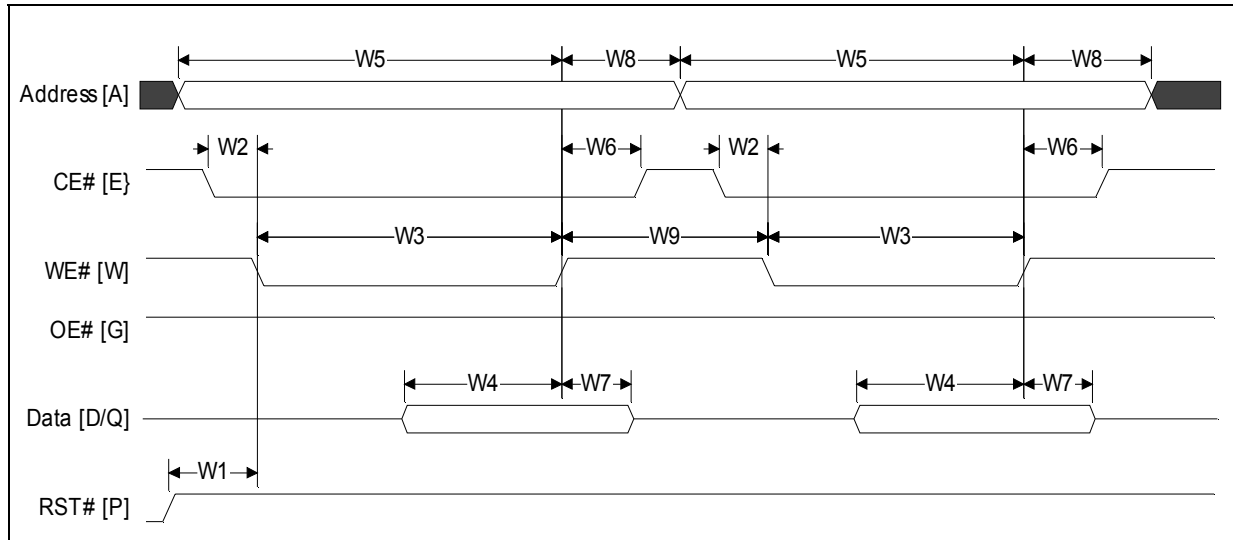
**Table 19: AC Write Specifications**

| Num                                       | Symbol       | Parameter                      | Min             | Max | Unit | Notes      |
|---|--------------|--------------------------------|-----------------|-----|------|------------|
| W1  | $t_{PHWL}$   | RST# high recovery to WE# low  | 150             | -   | ns   | 1,2,3      |
| W2  | $t_{ELWL}$   | CE# setup to WE# low           | 0               | -   | ns   | 1,2,3      |
| W3  | $t_{WLWH}$   | WE# write pulse width low      | 50              | -   | ns   | 1,2,4      |
| W4  | $t_{DVVWH}$  | Data setup to WE# high         | 50              | -   | ns   | 1,2        |
| W5  | $t_{AVVWH}$  | Address setup to WE# high      | 50              | -   | ns   |            |
| W6  | $t_{WHEH}$   | CE# hold from WE# high         | 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   | 1,2,5      |
| W10                                       | $t_{VPWH}$   | $V_{PP}$ setup to WE# high     | 200             | -   | ns   | 1,2,3,7    |
| W11                                       | $t_{QVVL}$   | $V_{PP}$ hold from Status read | 0               | -   | ns   |            |
| W12                                       | $t_{QVBL}$   | WP# hold from Status read      | 0               | -   | ns   | 1,2,3,7    |
| W13                                       | $t_{BHWL}$   | WP# setup to WE# high          | 200             | -   | ns   |            |
| W14                                       | $t_{WHGL}$   | WE# high to OE# low            | 0               | -   | ns   | 1,2,9      |
| W16                                       | $t_{WHQV}$   | WE# high to read valid         | $t_{AVQV} + 35$ | -   | ns   | 1,2,3,6,10 |
| Write to Asynchronous Read Specifications |              |                                |                 |     |      |            |
| W18                                       | $t_{WHAV}$   | WE# high to Address valid      | 0               | -   | ns   | 1,2,3,6,8  |
| Write to Synchronous Read Specifications  |              |                                |                 |     |      |            |
| W19                                       | $t_{WHCH/L}$ | WE# high to Clock valid        | 19              | -   | ns   | 1,2,3,6,10 |
| W20                                       | $t_{WHVH}$   | WE# high to ADV# high          | 19              | -   | ns   |            |
| Write Specifications with Clock Active    |              |                                |                 |     |      |            |
| W21                                       | $t_{VHWL}$   | ADV# high to WE# low           | -               | 20  | ns   | 1,2,3,11   |
| W22                                       | $t_{CHWL}$   | Clock high to WE# low          | -               | 20  | ns   |            |

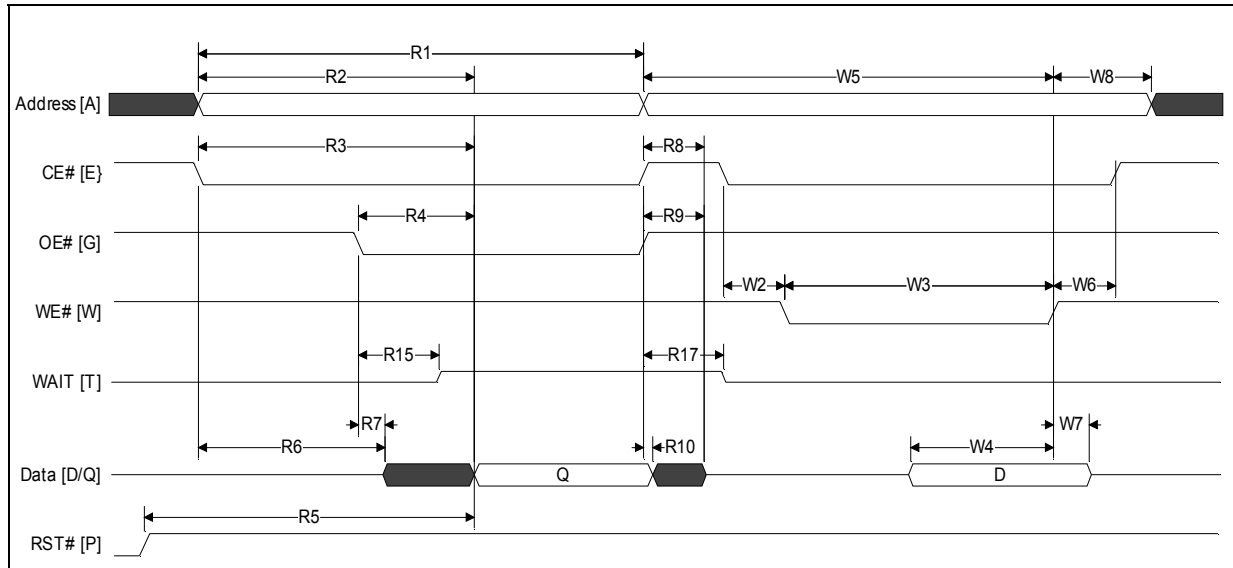
**Notes:**

- Write timing characteristics during erase suspend are the same as write-only operations.
- A write operation can be terminated with either CE# or WE#.
- Sampled, not 100% tested.
- 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,  $t_{WLWH} = t_{ELEH} = t_{WLEH} = t_{ELWH}$ .
- Write pulse width high ( $t_{WHWL}$  or  $t_{EHEL}$ ) is defined from CE# or WE# high (whichever occurs first) to CE# or WE# low (whichever occurs last). Hence,  $t_{WHWL} = t_{EHEL} = t_{WHEL} = t_{EHWL}$ .
- $t_{WHVH}$  or  $t_{WHCH/L}$  must be met when transitioning from a write cycle to a synchronous burst read.
- $V_{PP}$  and WP# should be at a valid level until erase or program success is determined.
- This specification is only applicable when transitioning from a write cycle to an asynchronous read. See spec W19 and W20 for synchronous read.
- When doing a Read Status operation following any command that alters the Status Register, W14 is 20 ns.
- Add 10 ns if the write operation results in a RCR or block lock status change, for the subsequent read operation to reflect this change.
- These specs are required only when the device is in a synchronous mode and clock is active during address setup phase.

**Figure 20: Write-to-Write Timing**



**Figure 21: Asynchronous Read-to-Write Timing**



**Note:** WAIT deasserted during asynchronous read and during write. WAIT High-Z during write per OE# deasserted.

Figure 22: Write-to-Asynchronous Read Timing

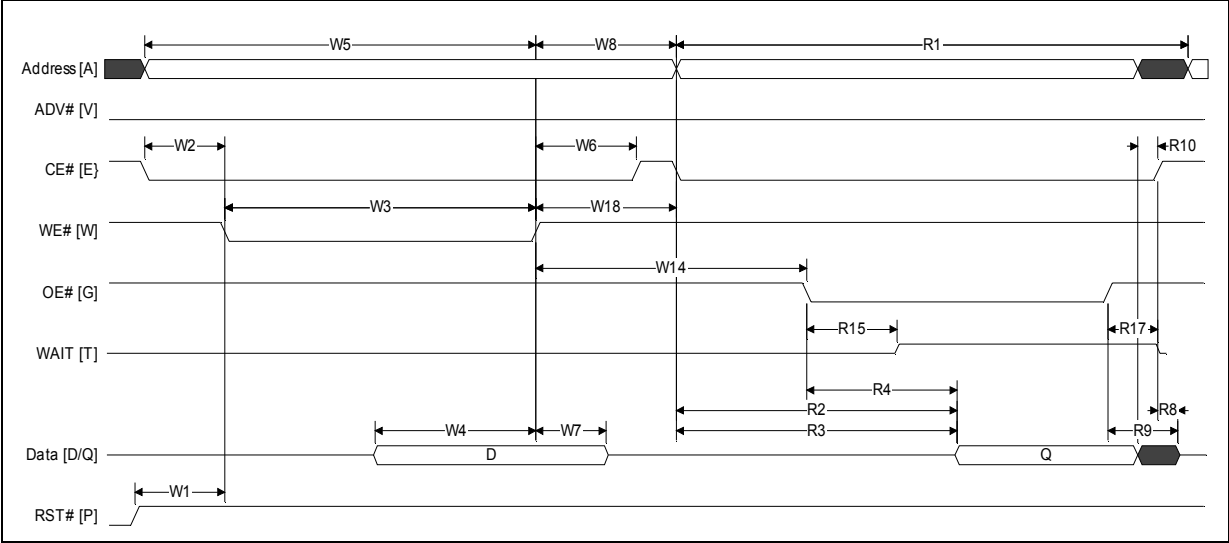
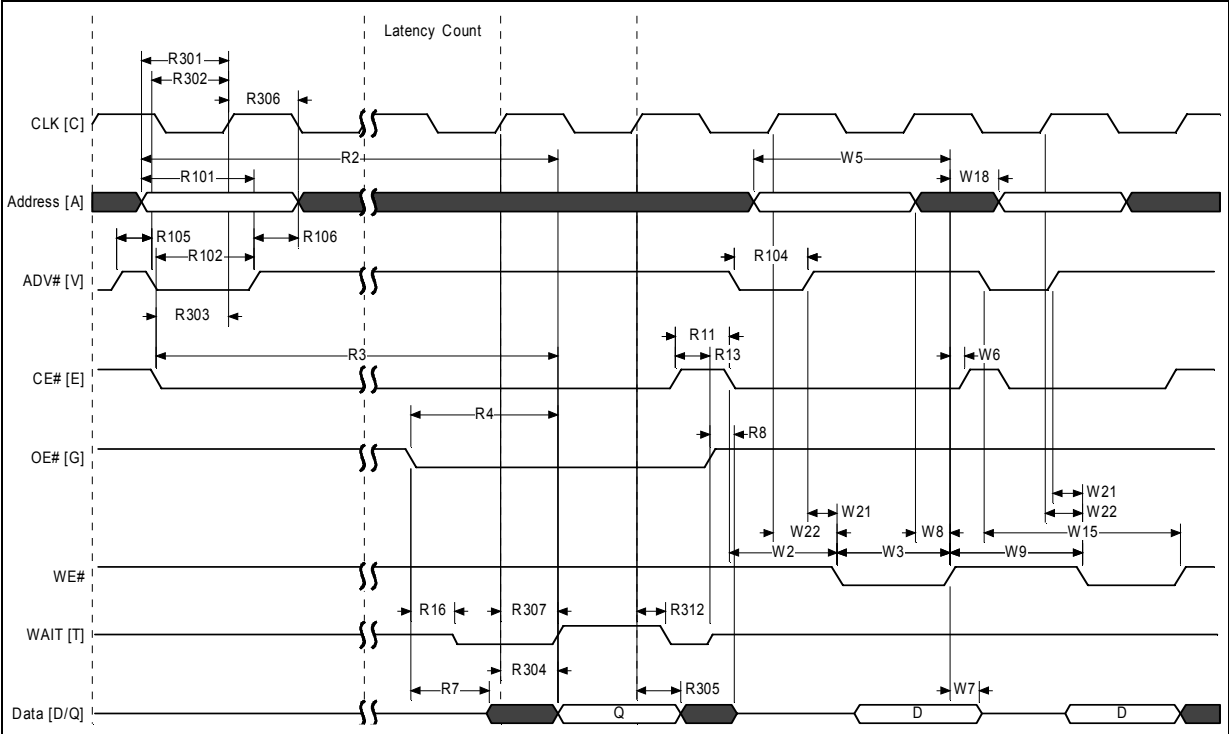
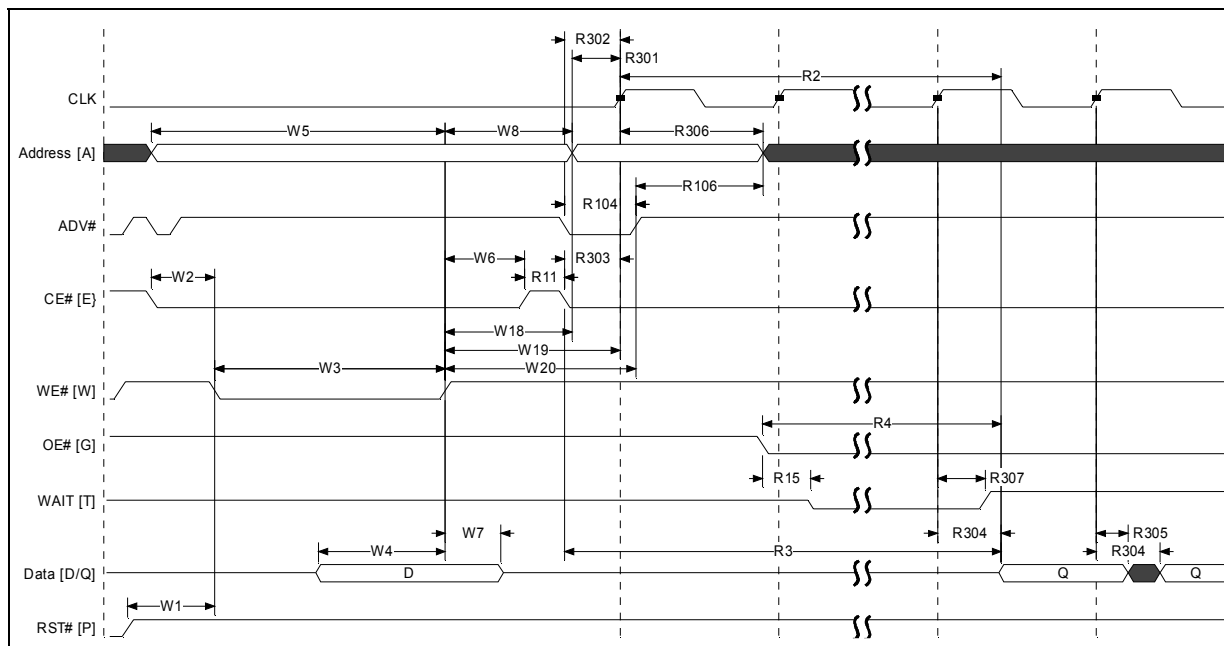


Figure 23: Synchronous Read-to-Write Timing



**Note:** WAIT shown deasserted and High-Z per OE# deassertion during write operation (RCR[10]=0, Wait asserted low). Clock is ignored during write operation.

**Figure 24: Write-to-Synchronous Read Timing**



**Note:** WAIT shown deasserted and High-Z per OE# deassertion during write operation (RCR[10]=0, Wait asserted low).

## 7.5 Program and Erase Characteristics

**Table 20:**

| Num                                   | Symbol                  | Parameter    | V <sub>PPL</sub>    |     |     | V <sub>PPH</sub> |     |     | Units | Notes |     |
|---------------------------------------|-------------------------|--------------|---------------------|-----|-----|------------------|-----|-----|-------|-------|-----|
|                                       |                         |              | Min                 | Typ | Max | Min              | Typ | Max |       |       |     |
| Conventional Word Programming         |                         |              |                     |     |     |                  |     |     |       |       |     |
| W200                                  | t <sub>PROG/W</sub>     | Program Time | Single word - 130nm | -   | 90  | 200              | -   | 85  | 190   | μs    | 1   |
|                                       |                         |              | Single word - 65nm  | -   | 125 | 150              | -   | 125 | 150   |       |     |
|                                       |                         |              | Single cell         | -   | 30  | 60               | -   | 30  | 60    |       |     |
| Buffered Programming                  |                         |              |                     |     |     |                  |     |     |       |       |     |
| W200                                  | t <sub>PROG/W</sub>     | Program Time | Single word         | -   | 90  | 200              | -   | 85  | 190   | μs    | 1   |
| W251                                  | t <sub>BUFF</sub>       |              | 32-word buffer      | -   | 440 | 880              | -   | 340 | 680   |       |     |
| Buffered Enhanced Factory Programming |                         |              |                     |     |     |                  |     |     |       |       |     |
| W451                                  | t <sub>BEFP/W</sub>     | Program      | Single word         | n/a | n/a | n/a              | -   | 10  | -     | μs    | 1,2 |
| W452                                  | t <sub>BEFP/Setup</sub> |              | BEFP Setup          | n/a | n/a | n/a              | 5   | -   | -     |       | 1   |
| Erasing and Suspending                |                         |              |                     |     |     |                  |     |     |       |       |     |

Table 20:

| Num  | Symbol                | Parameter       |                    | V <sub>PPL</sub> |     |     | V <sub>PPH</sub> |     |     | Units | Notes |
|------|-----------------------|-----------------|--------------------|------------------|-----|-----|------------------|-----|-----|-------|-------|
|      |                       |                 |                    | Min              | Typ | Max | Min              | Typ | Max |       |       |
| W500 | t <sub>ERS/PB</sub>   | Erase Time      | 32-KByte Parameter | -                | 0.4 | 2.5 | -                | 0.4 | 2.5 | s     | 1     |
| W501 | t <sub>ERS/MB</sub>   |                 | 128-KByte Main     | -                | 1.2 | 4.0 | -                | 1.0 | 4.0 |       |       |
| W600 | t <sub>SUSP/P</sub>   | Suspend Latency | Program suspend    | -                | 20  | 25  | -                | 20  | 25  | μs    |       |
| W601 | t <sub>SUSP/E</sub>   |                 | Erase suspend      | -                | 20  | 25  | -                | 20  | 25  |       |       |
| W602 | t <sub>ERS/SUSP</sub> |                 | Erase to Suspend   | -                | 500 | -   | -                | 500 | -   |       | 1,3   |

**Notes:**

1. Typical values measured at T<sub>C</sub> = +25 °C and nominal voltages. Performance numbers are valid for all speed versions. Excludes system overhead. Sampled, but not 100% tested.
2. Averaged over entire device.
3. W602 is the typical 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.



## 8.0 Power and Reset Specifications

### 8.1 Power-Up and Power-Down

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

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

### 8.2 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.

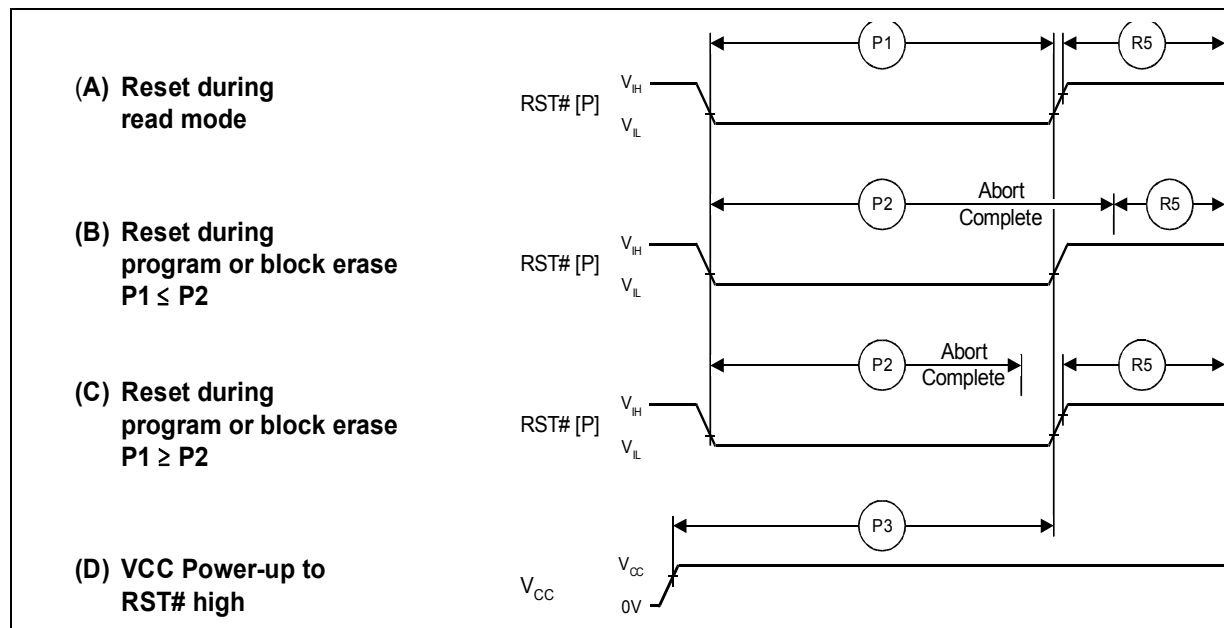
**Table 21:**

| Num | Symbol             | Parameter   | Min | Max | Unit | Notes   |
|-----|--------------------|---|-----|-----|------|---------|
| P1  | t <sub>PLPH</sub>  | RST# pulse width low                              | 100 | -   | ns   | 1,2,3,4 |
| P2  | t <sub>PLRH</sub>  | RST# low to device reset during erase             | -   | 25  | μs   | 1,3,4,7 |
|     |                    | RST# low to device reset during program           | -   | 25  |      | 1,3,4,7 |
| P3  | t <sub>VCCPH</sub> | VCC Power valid to RST# de-assertion (high) 130nm | 60  | -   |      | 1,4,5,6 |
|     |                    | VCC Power valid to RST# de-assertion (high) 65nm  | 300 | -   |      | 1,4,5,6 |

**Notes:**

1. These specifications are valid for all device versions (packages and speeds).
2. The device may reset if t<sub>PLPH</sub> is < t<sub>PLPH MIN</sub>, 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 VCC supply, device will not be ready until t<sub>VCCPH</sub> after V<sub>CC</sub> ≥ V<sub>CCMIN</sub>.
6. When RST# is tied to the VCCQ supply, device will not be ready until t<sub>VCCPH</sub> after V<sub>CC</sub> ≥ V<sub>CCMIN</sub>.
7. Reset completes within t<sub>PLPH</sub> if RST# is asserted while no erase or program operation is executing.

Figure 25: Reset Operation Waveforms



### 8.3 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.

Because Numonyx Multi-Level Cell (MLC) flash memory devices draw their power from VCC, VPP, and VCCQ, each power connection should have a 0.1  $\mu\text{F}$  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  $\mu\text{F}$  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.

## 9.0 Device Operations

This section provides an overview of device operations. 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.

### 9.1 Bus Operations

CE# low and RST# high enable device read operations. The device internally decodes upper address inputs to determine the accessed block. ADV# low opens the internal address latches. OE# low activates the outputs and gates selected data onto the I/O bus.

In asynchronous mode, the address is latched when ADV# goes high or continuously flows through if ADV# is held low. In synchronous mode, the address is latched by the first of either the rising ADV# edge or the next valid CLK edge with ADV# low (WE# and RST# must be  $V_{IH}$ ; CE# must be  $V_{IL}$ ).

Bus cycles to/from the P30 device conform to standard microprocessor bus operations. [Table 22](#) summarizes the bus operations and the logic levels that must be applied to the device control signal inputs.

**Table 22: Bus Operations Summary**

| Bus Operation  |              | RST#     | CLK     | ADV# | CE# | OE# | WE# | WAIT       | DQ[15:0]<br>] | Notes |
|----------------|--------------|----------|---------|------|-----|-----|-----|------------|---------------|-------|
| Read           | Asynchronous | $V_{IH}$ | X       | L    | L   | L   | H   | Deasserted | Output        |       |
|                | Synchronous  | $V_{IH}$ | Running | L    | L   | L   | H   | Driven     | Output        |       |
| Write          |              | $V_{IH}$ | X       | L    | L   | H   | L   | High-Z     | Input         | 1     |
| Output Disable |              | $V_{IH}$ | X       | X    | L   | H   | H   | High-Z     | High-Z        | 2     |
| Standby        |              | $V_{IH}$ | X       | X    | H   | X   | X   | High-Z     | High-Z        | 2     |
| Reset          |              | $V_{IL}$ | X       | X    | X   | X   | X   | High-Z     | High-Z        | 2,3   |

**Notes:**

1. Refer to the [Table 23, "Command Bus Cycles" on page 45](#) for valid DQ[15:0] during a write operation.
2. X = Don't Care (H or L).
3. RST# must be at  $V_{SS} \pm 0.2$  V to meet the maximum specified power-down current.

#### 9.1.1 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. See [Section 10.0, "Read Operations" on page 48](#) for details on the available read modes, and see [Section 14.0, "Special Read States" on page 69](#) for details regarding the available read states.

### 9.1.2 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 23, "Command Bus Cycles" on page 45](#) shows the bus cycle sequence for each of the supported device commands, while [Table 24, "Command Codes and Definitions" on page 46](#) describes each command. See [Section 7.0, "AC Characteristics" on page 28](#) for signal-timing details.

*Note:* Write operations with invalid V<sub>CC</sub> and/or V<sub>PP</sub> voltages can produce spurious results and should not be attempted.

### 9.1.3 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.

### 9.1.4 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<sub>CCS</sub>, 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.

### 9.1.5 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 7.0, "AC Characteristics" on page 28](#) for details about signal-timing.

## 9.2 Device Commands

Device operations are initiated by writing specific device commands to the Command User Interface (CUI). See [Table 23, "Command Bus Cycles" on page 45](#). 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 23: Command Bus Cycles**

| Mode                    | Command   | Bus Cycles | First Bus Cycle |                     |                     | Second Bus Cycle |                     |                     |
|-------------------------|---|------------|-----------------|---------------------|---------------------|------------------|---------------------|---------------------|
|                         |   |            | Oper            | Addr <sup>(1)</sup> | Data <sup>(2)</sup> | Oper             | Addr <sup>(1)</sup> | Data <sup>(2)</sup> |
| Read                    | Read Array  | 1          | Write           | DnA                 | 0xFF                | -                | -                   | -                   |
|                         | Read Device Identifier                                  | ≥ 2        | Write           | DnA                 | 0x90                | Read             | DBA + IA            | ID                  |
|                         | CFI Query   | ≥ 2        | Write           | DnA                 | 0x98                | Read             | DBA + QA            | QD                  |
|                         | Read Status Register                                    | 2          | Write           | DnA                 | 0x70                | Read             | DnA                 | SRD                 |
|                         | Clear Status Register                                   | 1          | Write           | DnA                 | 0x50                | -                | -                   | -                   |
| Program                 | Word Program  | 2          | Write           | WA                  | 0x40/<br>0x10       | Write            | WA                  | WD                  |
|                         | Buffered Program <sup>(3)</sup>                         | > 2        | Write           | WA                  | 0xE8                | Write            | WA                  | N - 1               |
|                         | Buffered Enhanced Factory Program (BEFP) <sup>(4)</sup> | > 2        | Write           | WA                  | 0x80                | Write            | WA                  | 0xD0                |
| Erase                   | Block Erase   | 2          | Write           | BA                  | 0x20                | Write            | BA                  | 0xD0                |
| Suspend                 | Program/Erase Suspend                                   | 1          | Write           | DnA                 | 0xB0                | -                | -                   | -                   |
|                         | Program/Erase Resume                                    | 1          | Write           | DnA                 | 0xD0                | -                | -                   | -                   |
| Block Locking/Unlocking | Lock Block  | 2          | Write           | BA                  | 0x60                | Write            | BA                  | 0x01                |
|                         | Unlock Block  | 2          | Write           | BA                  | 0x60                | Write            | BA                  | 0xD0                |
|                         | Lock-down Block   | 2          | Write           | BA                  | 0x60                | Write            | BA                  | 0x2F                |
| Protection              | Program Protection Register                             | 2          | Write           | PRA                 | 0xC0                | Write            | PRA                 | PD                  |
|                         | Program Lock Register                                   | 2          | Write           | LRA                 | 0xC0                | Write            | LRA                 | LRD                 |
| Configuration           | Program Read Configuration Register                     | 2          | Write           | RCD                 | 0x60                | Write            | RCD                 | 0x03                |

**Notes:**

1. First command cycle address should be the same as the operation's target address.  
 DBA = Device Base Address (NOTE: needed for dual-die 512 Mb device)  
 DnA = Address within the device.  
 IA = Identification code address offset.  
 QA = CFI Query address offset.  
 WA = Word address of memory location to be written.  
 BA = Address within the block.  
 PRA = Protection Register address.  
 LRA = Lock Register address.  
 RCD = Read Configuration Register data on QUAD+ A[15:0] or EASY BGA A[16:1].
2. ID = Identifier data.  
 QD = Query data on DQ[15:0].  
 SRD = Status Register data.  
 WD = Word data.  
 N = Word count of data to be loaded into the write buffer.  
 PD = Protection Register data.  
 LRD = Lock Register data.
3. The second cycle of the Buffered Program Command is the word count of the data to be loaded into the write buffer. This is followed by up to 32 words of data. Then the confirm command (0xD0) is issued, triggering the array programming operation.
4. The confirm command (0xD0) is followed by the buffer data.

## 9.3 Command Definitions

Valid device command codes and descriptions are shown in [Table 24](#).

**Table 24: Command Codes and Definitions (Sheet 1 of 2)**

| Mode    | Code | Device Mode                              | Description   |
|---------|------|--|---|
| Read    | 0xFF | Read Array                               | Places the device in Read Array mode. Array data is output on DQ[15:0].   |
|         | 0x70 | Read Status Register                     | Places the device in Read Status Register mode. The device enters this mode after a program or erase command is issued. Status Register data is output on DQ[7:0].  |
|         | 0x90 | Read Device ID or Configuration Register | Places device in Read Device Identifier mode. Subsequent reads output manufacturer/device codes, Configuration Register data, Block Lock status, or Protection Register data on DQ[15:0].   |
|         | 0x98 | Read Query                               | Places the device in Read Query mode. Subsequent reads output Common Flash Interface information on DQ[7:0].  |
|         | 0x50 | Clear Status Register                    | The WSM can only set Status Register error bits. The Clear Status Register command is used to clear the SR error bits.  |
| Write   | 0x40 | Word Program Setup                       | First cycle of a 2-cycle programming command; prepares the CUI for a write operation. On the next write cycle, the address and data are latched and the WSM executes the programming algorithm at the addressed location. During program operations, the device responds only to Read Status Register and Program Suspend commands. CE# or OE# must be toggled to update the Status Register in asynchronous read. CE# or ADV# must be toggled to update the Status Register Data for synchronous Non-array reads. The Read Array command must be issued to read array data after programming has finished. |
|         | 0x10 | Alternate Word Program Setup             | Equivalent to the Word Program Setup command, 0x40.   |
|         | 0xE8 | Buffered Program                         | This command loads a variable number of words up to the buffer size of 32 words onto the program buffer.  |
|         | 0xD0 | Buffered Program Confirm                 | The confirm command is Issued after the data streaming for writing into the buffer is done. This instructs the WSM to perform the Buffered Program algorithm, writing the data from the buffer to the flash memory array.   |
|         | 0x80 | BEFP Setup                               | First cycle of a 2-cycle command; initiates Buffered Enhanced Factory Program mode (BEFP). The CUI then waits for the BEFP Confirm command, 0xD0, that initiates the BEFP algorithm. All other commands are ignored when BEFP mode begins.  |
| Erase   | 0xD0 | BEFP Confirm                             | If the previous command was BEFP Setup (0x80), the CUI latches the address and data, and prepares the device for BEFP mode.   |
|         | 0x20 | Block Erase Setup                        | First cycle of a 2-cycle command; prepares the CUI for a block-erase operation. The WSM performs the erase algorithm on the block addressed by the Erase Confirm command. If the next command is <i>not</i> the Erase Confirm (0xD0) command, the CUI sets Status Register bits SR[4] and SR[5], and places the device in read status register mode.  |
| Suspend | 0xD0 | Block Erase Confirm                      | If the first command was Block Erase Setup (0x20), the CUI latches the address and data, and the WSM erases the addressed block. During block-erase operations, the device responds only to Read Status Register and Erase Suspend commands. CE# or OE# must be toggled to update the Status Register in asynchronous read. CE# or ADV# must be toggled to update the Status Register Data for synchronous Non-array reads  |
|         | 0xB0 | Program or Erase Suspend                 | This command issued to any device address initiates a suspend of the currently-executing program or block erase operation. The Status Register indicates successful suspend operation by setting either SR[2] (program suspended) or SR[6] (erase suspended), along with SR[7] (ready). The Write State Machine remains in the suspend mode regardless of control signal states (except for RST# asserted).   |
|         | 0xD0 | Suspend Resume                           | This command issued to any device address resumes the suspended program or block-erase operation.   |

**Table 24: Command Codes and Definitions (Sheet 2 of 2)**

| Mode                        | Code | Device Mode                       | Description  |
|-----------------------------|------|-----------------------------------|--|
| Block Locking/<br>Unlocking | 0x60 | Lock Block Setup                  | First cycle of a 2-cycle command; prepares the CUI for block lock configuration changes. If the next command is not Block Lock (0x01), Block Unlock (0xD0), or Block Lock-Down (0x2F), the CUI sets Status Register bits SR[4] and SR[5], indicating a command sequence error. |
|                             | 0x01 | Lock Block                        | If the previous command was Block Lock Setup (0x60), the addressed block is locked.  |
|                             | 0xD0 | Unlock Block                      | If the previous command was Block Lock Setup (0x60), the addressed block is unlocked. If the addressed block is in a lock-down state, the operation has no effect.   |
|                             | 0x2F | Lock-Down Block                   | If the previous command was Block Lock Setup (0x60), the addressed block is locked down.   |
| Protection                  | 0xC0 | Program Protection Register Setup | First cycle of a 2-cycle command; prepares the device for a Protection Register or Lock Register program operation. The second cycle latches the register address and data, and starts the programming algorithm   |
| Configuration               | 0x60 | Read Configuration Register Setup | First cycle of a 2-cycle command; prepares the CUI for device read configuration. If the Set Read Configuration Register command (0x03) is not the next command, the CUI sets Status Register bits SR[4] and SR[5], indicating a command sequence error.                       |
|                             | 0x03 | Read Configuration Register       | If the previous command was Read Configuration Register Setup (0x60), the CUI latches the address and writes A[15:0] to the Read Configuration Register. Following a Configure Read Configuration Register command, subsequent read operations access array data.              |

## 10.0 Read Operations

The device supports two read modes: asynchronous page mode and synchronous burst mode. Asynchronous page mode is the default read mode after device power-up or a reset. The Read Configuration Register must be configured to enable synchronous burst reads of the flash memory array (see [Section 10.3, "Read Configuration Register" on page 49](#)).

The device can be in any of four read states: Read Array, Read Identifier, Read Status or Read Query. Upon power-up, or after a reset, the device defaults to Read Array. To change the read state, the appropriate read command must be written to the device (see [Section 9.2, "Device Commands" on page 45](#)). See [Section 14.0, "Special Read States" on page 69](#) for details regarding Read Status, Read ID, and CFI Query modes. The following sections describe read-mode operations in detail.

### 10.1 Asynchronous Page-Mode Read

Following a device power-up or reset, asynchronous page mode is the default read mode and the device is set to Read Array. However, to perform array reads after any other device operation (e.g. write operation), the Read Array command must be issued in order to read from the flash memory array.

*Note:* Asynchronous page-mode reads can only be performed when Read Configuration Register bit RCR[15] is set (see [Section 10.3, "Read Configuration Register" on page 49](#)).

To perform an asynchronous page-mode read, an address is driven onto the Address bus, and CE# and ADV# are asserted. WE# and RST# must already have been deasserted. WAIT is deasserted during asynchronous page mode. ADV# can be driven high to latch the address, or it must be held low throughout the read cycle. CLK is not used for asynchronous page-mode reads, and is ignored. If only asynchronous reads are to be performed, CLK should be tied to a valid  $V_{IH}$  level, WAIT signal can be floated and ADV# must be tied to ground. Array data is driven onto DQ[15:0] after an initial access time  $t_{AVQV}$  delay. (see [Section 7.0, "AC Characteristics" on page 28](#)).

In asynchronous page mode, four data words are "sensed" simultaneously from the flash memory array and loaded into an internal page buffer. The buffer word corresponding to the initial address on the Address bus is driven onto DQ[15:0] after the initial access delay. The lowest two address bits determine which word of the 4-word page is output from the data buffer at any given time.

### 10.2 Synchronous Burst-Mode Read

To perform a synchronous burst-read, an initial address is driven onto the Address bus, and CE# and ADV# are asserted. WE# and RST# must already have been deasserted. ADV# is asserted, and then deasserted to latch the address. Alternately, ADV# can remain asserted throughout the burst access, in which case the address is latched on the next valid CLK edge while ADV# is asserted.

During synchronous array and non-array read modes, the first word is output from the data buffer on the next valid CLK edge after the initial access latency delay (see [Section 10.3.2, "Latency Count" on page 50](#)). Subsequent data is output on valid CLK edges following a minimum delay. However, for a synchronous non-array read, the same word of data will be output on successive clock edges until the burst length requirements are satisfied. Refer to the following waveforms for more detailed information:

- [Figure 17, "Synchronous Single-Word Array or Non-array Read Timing" on page 34](#)
- [Figure 18, "Continuous Burst Read, Showing An Output Delay Timing" on page 35](#)
- [Figure 19, "Synchronous Burst-Mode Four-Word Read Timing" on page 35](#)



## 10.3 Read Configuration Register

The Read Configuration Register (RCR) is used to select the read mode (synchronous or asynchronous), and it defines the synchronous burst characteristics of the device. To modify RCR settings, use the Configure Read Configuration Register command (see [Section 9.2, "Device Commands" on page 45](#)).

RCR contents can be examined using the Read Device Identifier command, and then reading from offset 0x05 (see [Section 14.2, "Read Device Identifier" on page 70](#)).

The RCR is shown in [Table 25](#). The following sections describe each RCR bit.

**Table 25: Read Configuration Register Description**

| Read Configuration Register (RCR) |                         |               |    |    |   |           |            |           |          |     |     |            |              |   |   |
|-----------------------------------|-------------------------|---------------|----|----|---|-----------|------------|-----------|----------|-----|-----|------------|--------------|---|---|
| Read Mode                         | RES                     | Latency Count |    |    | WAIT Polarity   | Data Hold | WAIT Delay | Burst Seq | CLK Edge | RES | RES | Burst Wrap | Burst Length |   |   |
| RM                                | R                       | LC[2:0]       |    |    | WP  | DH        | WD         | BS        | CE       | R   | R   | BW         | BL[2:0]      |   |   |
| 15                                | 14                      | 13            | 12 | 11 | 10  | 9         | 8          | 7         | 6        | 5   | 4   | 3          | 2            | 1 | 0 |
| Bit                               | Name                    |               |    |    | Description   |           |            |           |          |     |     |            |              |   |   |
| 15                                | Read Mode (RM)          |               |    |    | 0 = Synchronous burst-mode read<br>1 = Asynchronous page-mode read (default)  |           |            |           |          |     |     |            |              |   |   |
| 14                                | Reserved (R)            |               |    |    | Reserved bits should be cleared (0)   |           |            |           |          |     |     |            |              |   |   |
| 13:11                             | Latency Count (LC[2:0]) |               |    |    | 010 =Code 2<br>011 =Code 3<br>100 =Code 4<br>101 =Code 5<br>110 =Code 6<br>111 =Code 7 (default)<br>(Other bit settings are reserved)     |           |            |           |          |     |     |            |              |   |   |
| 10                                | Wait Polarity (WP)      |               |    |    | 0 =WAIT signal is active low<br>1 =WAIT signal is active high (default)   |           |            |           |          |     |     |            |              |   |   |
| 9                                 | Data Hold (DH)          |               |    |    | 0 =Data held for a 1-clock data cycle<br>1 =Data held for a 2-clock data cycle (default)  |           |            |           |          |     |     |            |              |   |   |
| 8                                 | Wait Delay (WD)         |               |    |    | 0 =WAIT deasserted with valid data<br>1 =WAIT deasserted one data cycle before valid data (default)                                       |           |            |           |          |     |     |            |              |   |   |
| 7                                 | Burst Sequence (BS)     |               |    |    | 0 =Reserved<br>1 =Linear (default)  |           |            |           |          |     |     |            |              |   |   |
| 6                                 | Clock Edge (CE)         |               |    |    | 0 = Falling edge<br>1 = Rising edge (default)   |           |            |           |          |     |     |            |              |   |   |
| 5:4                               | Reserved (R)            |               |    |    | Reserved bits should be cleared (0)   |           |            |           |          |     |     |            |              |   |   |
| 3                                 | Burst Wrap (BW)         |               |    |    | 0 =Wrap; Burst accesses wrap within burst length set by BL[2:0]<br>1 =No Wrap; Burst accesses do not wrap within burst length (default)   |           |            |           |          |     |     |            |              |   |   |
| 2:0                               | Burst Length (BL[2:0])  |               |    |    | 001 =4-word burst<br>010 =8-word burst<br>011 =16-word burst<br>111 =Continuous-word burst (default)<br>(Other bit settings are reserved) |           |            |           |          |     |     |            |              |   |   |

**Note:** Latency Code 2, Data Hold for a 2-clock data cycle (DH = 1) WAIT must be deasserted with valid data (WD = 0). Latency Code 2, Data Hold for a 2-clock data cycle (DH=1) WAIT deasserted one data cycle before valid data (WD = 1) combination is not supported. [Table 25, "Read Configuration Register Description" on page 49](#) is shown using the QUAD+ package. For EASY BGA and TSOP packages, the table reference should be adjusted using address bits A[16:1].

### 10.3.1 Read Mode

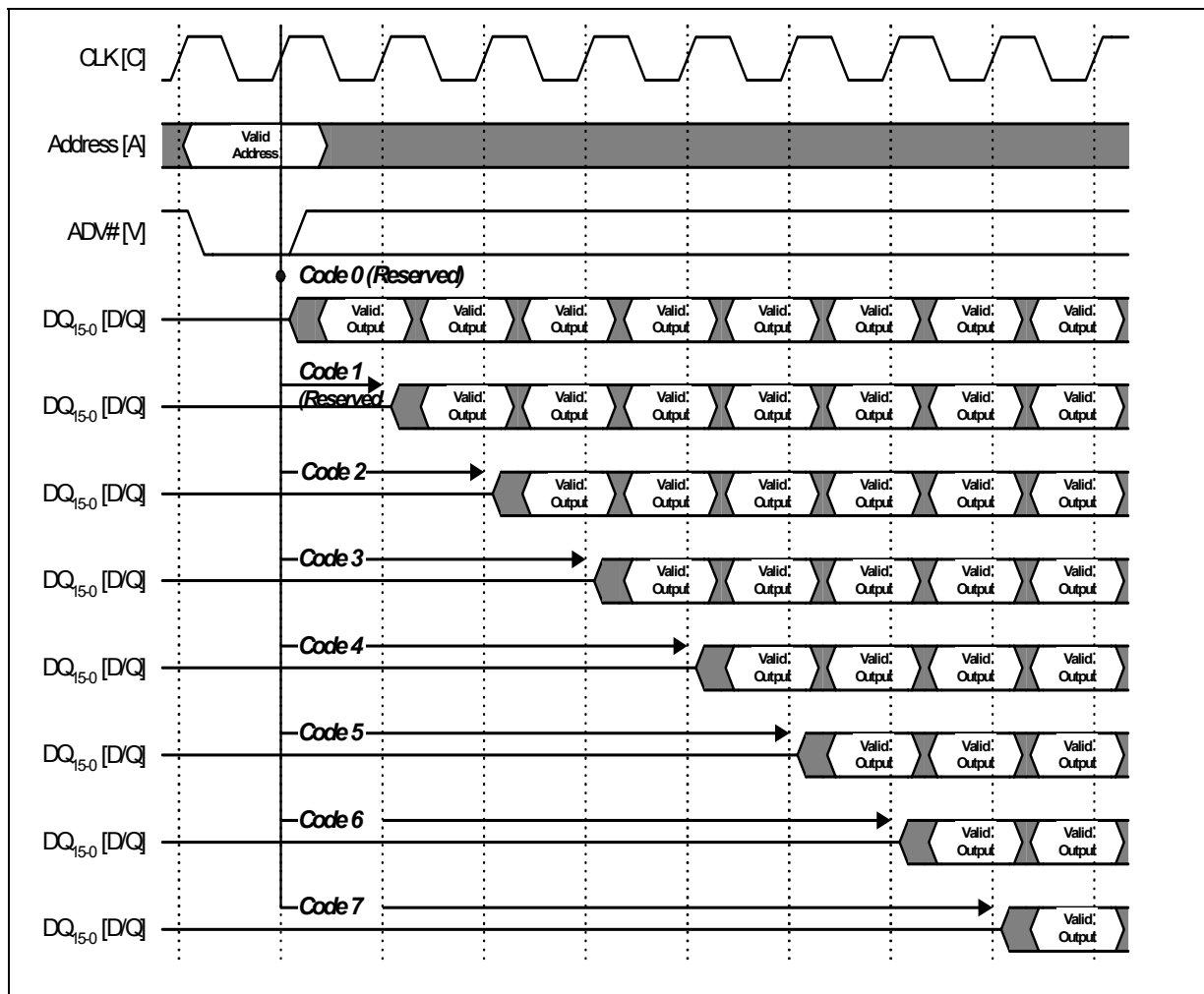
The Read Mode (RM) bit selects synchronous burst-mode or asynchronous page-mode operation for the device. When the RM bit is set, asynchronous page mode is selected (default). When RM is cleared, synchronous burst mode is selected.

### 10.3.2 Latency Count

The Latency Count (LC) bits tell the device how many clock cycles must elapse from the rising edge of ADV# (or from the first valid clock edge after ADV# is asserted) until the first valid data word is to be driven onto DQ[15:0]. The input clock frequency is used to determine this value and [Figure 26](#) shows the data output latency for the different settings of LC. The maximum Latency Count for P30 would be Code 4 based on the Max Clock frequency specification of 52 mhz, and there will be zero WAIT States when bursting within the word line. Please also refer to ["End of Word Line \(EOWL\) Considerations"](#) on [page 55](#) for more information on EOWL.

Refer to [Table 26, "Latency Count \(LC\) and Frequency Support"](#) on [page 51](#) for Latency Code Settings.

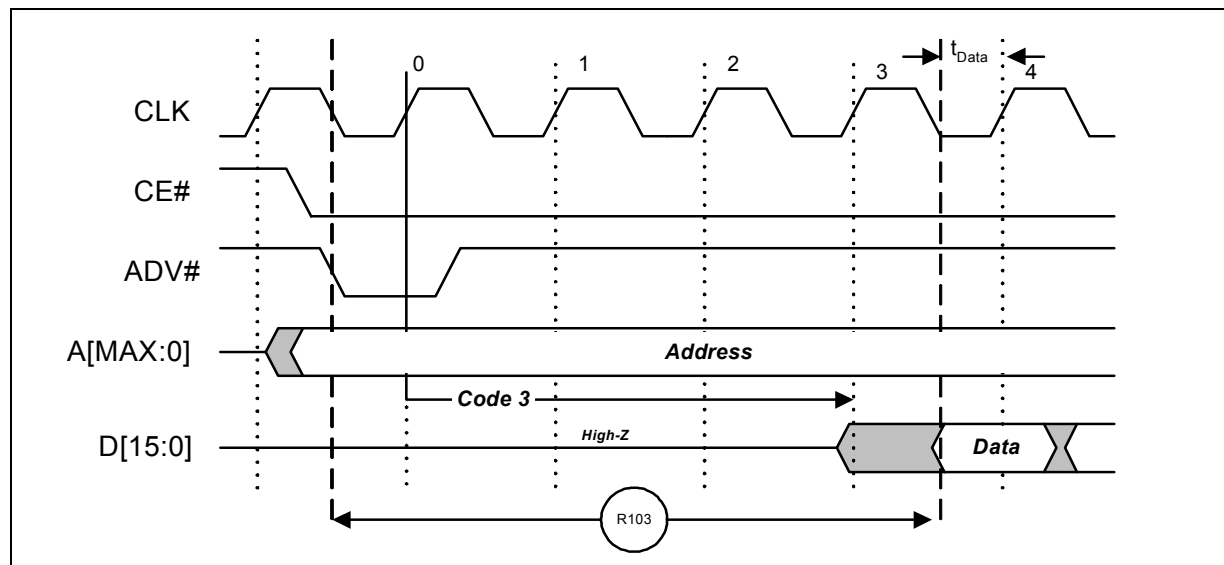
**Figure 26: First-Access Latency Count**



**Table 26: Latency Count (LC) and Frequency Support**

| Latency Count Settings | Frequency Support (MHz) |
|------------------------|-------------------------|
| 2                      | £ 27                    |
| 3                      | £ 40                    |
| 4                      | £ 52                    |

**Note:** Synchronous burst read operation is currently not supported for the TSOP package.

**Figure 27: Example Latency Count Setting using Code 3**

### 10.3.3 WAIT Polarity

The WAIT Polarity bit (WP), RCR[10] determines the asserted level ( $V_{OH}$  or  $V_{OL}$ ) of WAIT. When WP is set, WAIT is asserted high (default). When WP is cleared, WAIT is asserted low. WAIT changes state on valid clock edges during active bus cycles (CE# asserted, OE# asserted, RST# deasserted).

#### 10.3.3.1 WAIT Signal Function

The WAIT signal indicates data valid when the device is operating in synchronous mode (RCR[15]=0). The WAIT signal is only "deasserted" when data is valid on the bus.

When the device is operating in synchronous non-array read mode, such as read status, read ID, or read query. The WAIT signal is also "deasserted" when data is valid on the bus.

WAIT behavior during synchronous non-array reads at the end of word line works correctly only on the first data access.

When the device is operating in asynchronous page mode, asynchronous single word read mode, and all write operations, WAIT is set to a deasserted state as determined by RCR[10]. See [Figure 15, "Asynchronous Single-Word Read \(ADV# Latch\)" on page 33](#), and [Figure 16, "Asynchronous Page-Mode Read Timing" on page 34](#).

**Table 27: WAIT Functionality Table (Sheet 1 of 2)**

| Condition                                    | WAIT   | Notes |
|--|--------|-------|
| CE# = '1', OE# = 'X' or CE# = '0', OE# = '1' | High-Z | 1     |
| CE# = '0', OE# = '0'                         | Active | 1     |
| Synchronous Array Reads                      | Active | 1     |
| Synchronous Non-Array Reads                  | Active | 1     |

**Table 27: WAIT Functionality Table (Sheet 2 of 2)**

| Condition              | WAIT       | Notes |
|------------------------|------------|-------|
| All Asynchronous Reads | Deasserted | 1     |
| All Writes             | High-Z     | 1,2   |

**Notes:**

- Active:** WAIT is asserted until data becomes valid, then deasserts
- When  $OE\# = V_{IH}$  during writes, WAIT = High-Z

**10.3.4 Data Hold**

For burst read operations, the Data Hold (DH) bit determines whether the data output remains valid on DQ[15:0] for one or two clock cycles. This period of time is called the “**data cycle**”. When DH is set, output data is held for two clocks (default). When DH is cleared, output data is held for one clock (see [Figure 28](#)). The processor’s data setup time and the flash memory’s clock-to-data output delay should be considered when determining whether to hold output data for one or two clocks. A method for determining the Data Hold configuration is shown below:

To set the device at one clock data hold for subsequent reads, the following condition must be satisfied:

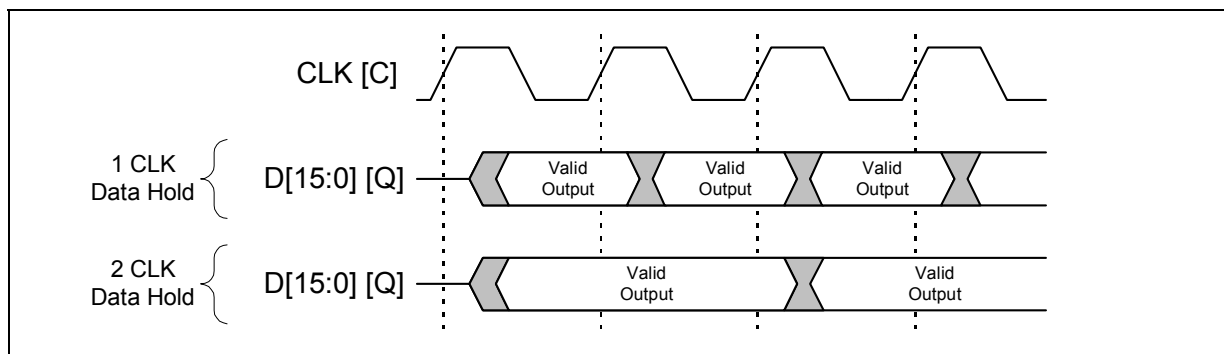
$$t_{CHQV} \text{ (ns)} + t_{DATA} \text{ (ns)} \leq \text{One CLK Period (ns)}$$

$$t_{DATA} = \text{Data set up to Clock (defined by CPU)}$$

For example, with a clock frequency of 40 MHz, the clock period is 25 ns. Assuming  $t_{CHQV} = 20$  ns and  $t_{DATA} = 4$  ns. Applying these values to the formula above:

$$20 \text{ ns} + 4 \text{ ns} \leq 25 \text{ ns}$$

The equation is satisfied and data will be available at every clock period with data hold setting at one clock. If  $t_{CHQV} \text{ (ns)} + t_{DATA} \text{ (ns)} > \text{One CLK Period (ns)}$ , data hold setting of 2 clock periods must be used.

**Figure 28: Data Hold Timing****10.3.5 WAIT Delay**

The WAIT Delay (WD) bit controls the WAIT assertion-delay behavior during synchronous burst reads. WAIT can be asserted either during or one data cycle before valid data is output on DQ[15:0]. When WD is set, WAIT is deasserted one data cycle before valid data (default). When WD is cleared, WAIT is deasserted during valid data.

### 10.3.6 Burst Sequence

The Burst Sequence (BS) bit selects linear-burst sequence (default). Only linear-burst sequence is supported. Table 28 shows the synchronous burst sequence for all burst lengths, as well as the effect of the Burst Wrap (BW) setting.

**Table 28: Burst Sequence Word Ordering**

| Start Addr. (DEC) | Burst Wrap (RCR[3]) | Burst Addressing Sequence (DEC) |                                |                                 |                                    |
|-------------------|---------------------|---------------------------------|--------------------------------|---------------------------------|------------------------------------|
|                   |                     | 4-Word Burst (BL[2:0] = 0b001)  | 8-Word Burst (BL[2:0] = 0b010) | 16-Word Burst (BL[2:0] = 0b011) | Continuous Burst (BL[2:0] = 0b111) |
| 0                 | 0                   | 0-1-2-3                         | 0-1-2-3-4-5-6-7                | 0-1-2-3-4...14-15               | 0-1-2-3-4-5-6-...                  |
| 1                 | 0                   | 1-2-3-0                         | 1-2-3-4-5-6-7-0                | 1-2-3-4-5...15-0                | 1-2-3-4-5-6-7-...                  |
| 2                 | 0                   | 2-3-0-1                         | 2-3-4-5-6-7-0-1                | 2-3-4-5-6...15-0-1              | 2-3-4-5-6-7-8-...                  |
| 3                 | 0                   | 3-0-1-2                         | 3-4-5-6-7-0-1-2                | 3-4-5-6-7...15-0-1-2            | 3-4-5-6-7-8-9-...                  |
| 4                 | 0                   |                                 | 4-5-6-7-0-1-2-3                | 4-5-6-7-8...15-0-1-2-3          | 4-5-6-7-8-9-10-...                 |
| 5                 | 0                   |                                 | 5-6-7-0-1-2-3-4                | 5-6-7-8-9...15-0-1-2-3-4        | 5-6-7-8-9-10-11-...                |
| 6                 | 0                   |                                 | 6-7-0-1-2-3-4-5                | 6-7-8-9-10...15-0-1-2-3-4-5     | 6-7-8-9-10-11-12-...               |
| 7                 | 0                   |                                 | 7-0-1-2-3-4-5-6                | 7-8-9-10...15-0-1-2-3-4-5-6     | 7-8-9-10-11-12-13-...              |
| :                 | :                   | :                               | :                              | :                               | :                                  |
| 14                | 0                   |                                 |                                | 14-15-0-1-2...12-13             | 14-15-16-17-18-19-20-...           |
| 15                | 0                   |                                 |                                | 15-0-1-2-3...13-14              | 15-16-17-18-19-20-21-...           |
| :                 | :                   | :                               | :                              | :                               | :                                  |
| 0                 | 1                   | 0-1-2-3                         | 0-1-2-3-4-5-6-7                | 0-1-2-3-4...14-15               | 0-1-2-3-4-5-6-...                  |
| 1                 | 1                   | 1-2-3-4                         | 1-2-3-4-5-6-7-8                | 1-2-3-4-5...15-16               | 1-2-3-4-5-6-7-...                  |
| 2                 | 1                   | 2-3-4-5                         | 2-3-4-5-6-7-8-9                | 2-3-4-5-6...16-17               | 2-3-4-5-6-7-8-...                  |
| 3                 | 1                   | 3-4-5-6                         | 3-4-5-6-7-8-9-10               | 3-4-5-6-7...17-18               | 3-4-5-6-7-8-9-...                  |
| 4                 | 1                   |                                 | 4-5-6-7-8-9-10-11              | 4-5-6-7-8...18-19               | 4-5-6-7-8-9-10-...                 |
| 5                 | 1                   |                                 | 5-6-7-8-9-10-11-12             | 5-6-7-8-9...19-20               | 5-6-7-8-9-10-11-...                |
| 6                 | 1                   |                                 | 6-7-8-9-10-11-12-13            | 6-7-8-9-10...20-21              | 6-7-8-9-10-11-12-...               |
| 7                 | 1                   |                                 | 7-8-9-10-11-12-13-14           | 7-8-9-10-11...21-22             | 7-8-9-10-11-12-13-...              |
| :                 | :                   | :                               | :                              | :                               | :                                  |
| 14                | 1                   |                                 |                                | 14-15-16-17-18...28-29          | 14-15-16-17-18-19-20-...           |
| 15                | 1                   |                                 |                                | 15-16-17-18-19...29-30          | 15-16-17-18-19-20-21-...           |

### 10.3.7 Clock Edge

The Clock Edge (CE) bit selects either a rising (default) or falling clock edge for CLK. This clock edge is used at the start of a burst cycle, to output synchronous data, and to assert/deassert WAIT.

### 10.3.8 Burst Wrap

The Burst Wrap (BW) bit determines whether 4-word, 8-word, or 16-word burst length accesses wrap within the selected word-length boundaries or cross word-length boundaries. When BW is set, burst wrapping does not occur (default). When BW is cleared, burst wrapping occurs.

When performing synchronous burst reads with BW set (no wrap), an output delay may occur when the burst sequence crosses its first device-row (16-word) boundary. If the burst sequence's start address is 4-word aligned, then no delay occurs. If the start address is at the end of a 4-word boundary, the worst case output delay is one clock cycle less than the first access Latency Count. This delay can take place only once, and doesn't occur if the burst sequence does not cross a device-row boundary. WAIT informs the system of this delay when it occurs.

### 10.3.9 Burst Length

The Burst Length bit (BL[2:0]) selects the linear burst length for all synchronous burst reads of the flash memory array. The burst lengths are 4-word, 8-word, 16-word, and continuous word.

Continuous-burst accesses are linear only, and do not wrap within any word length boundaries (see [Table 28, "Burst Sequence Word Ordering" on page 54](#)). When a burst cycle begins, the device outputs synchronous burst data until it reaches the end of the "burstable" address space.

### 10.3.10 End of Word Line (EOWL) Considerations

When performing synchronous burst reads with BW set (no wrap) and DH reset (1 clock cycle), an output "delay" requiring additional clock Wait States may occur when the burst sequence crosses its first device-row (16-word) boundary. The delay would take place only once, and will not occur if the burst sequence does not cross a device-row boundary. The WAIT signal informs the system of this delay when it occurs. If the burst sequence's start address is 4-word aligned (i.e. 0x00h, 0x04h, 0x08h, 0x0Ch) then no delay occurs. If the start address is at the end of a 4-word boundary (i.e. 0x03h, 0x07h, 0x0Bh, 0x0Fh), the worst case delay (number of Wait States required) will be one clock cycle less than the first access Latency Count (LC-1) when crossing the first device-row boundary (i.e. 0x0Fh to 0x10h). Other address misalignments may require wait states depending upon the LC setting and the starting address alignment. For example, an LC setting of 3 with a starting address of 0xFD requires 0 wait states, but the same LC setting of 3 with a starting address of 0xFE would require 1 wait state when crossing the first device row boundary.

## 11.0 Programming Operations

The device supports three programming methods: Word Programming (40h/10h), Buffered Programming (E8h, D0h), and Buffered Enhanced Factory Programming (80h, D0h). See

[Section 9.0, "Device Operations" on page 43](#) for details on the various programming commands issued to the device. The following sections describe device programming in detail.

Successful programming requires the addressed block to be unlocked. If the block is locked down, WP# must be deasserted and the block must be unlocked before attempting to program the block. Attempting to program a locked block causes a program error (SR[4] and SR[1] set) and termination of the operation. See [Section 13.0, "Security Modes" on page 64](#) for details on locking and unlocking blocks.

The Product Name is segmented into multiple 8-Mbit Programming Regions. See [Section 4.4, "Memory Maps" on page 22](#) for complete addressing. Execute in Place (XIP) applications must partition the memory such that code and data are in separate programming regions. XIP is executing code directly from flash memory. Each Programming Region should contain only code or data but not both. The following terms define the difference between code and data. System designs must use these definitions when partitioning their code and data for the P30 device.

- Code: Execution code ran out of the flash device on a continuous basis in the system.
- Data: Information periodically programmed into the flash device and read back (e.g. execution code shadowed and executed in RAM, pictures, log files, etc.).

### 11.1 Word Programming

Word programming operations are initiated by writing the Word Program Setup command to the device (see [Section 9.0, "Device Operations" on page 43](#)). This is followed by a second write to the device with the address and data to be programmed. The device outputs Status Register data when read. See [Figure 38, "Word Program Flowchart" on page 79](#).  $V_{pp}$  must be above  $V_{ppLK}$ , and within the specified  $V_{ppl}$  min/max values.

During programming, the Write State Machine (WSM) executes a sequence of internally-timed events that program the desired data bits at the addressed location, and verifies that the bits are sufficiently programmed. Programming the flash memory array changes "ones" to "zeros". Memory array bits that are zeros can be changed to ones only by erasing the block (see [Section 12.0, "Erase Operations" on page 62](#)).

The Status Register can be examined for programming progress and errors by reading at any address. The device remains in the Read Status Register state until another command is written to the device.

Status Register bit SR[7] indicates the programming status while the sequence executes. Commands that can be issued to the device during programming are Program Suspend, Read Status Register, Read Device Identifier, CFI Query, and Read Array (this returns unknown data).

When programming has finished, Status Register bit SR[4] (when set) indicates a programming failure. If SR[3] is set, the WSM could not perform the word programming operation because  $V_{pp}$  was outside of its acceptable limits. If SR[1] is set, the word programming operation attempted to program a locked block, causing the operation to abort.



Before issuing a new command, the Status Register contents should be examined and then cleared using the Clear Status Register command. Any valid command can follow, when word programming has completed.

### 11.1.1 Factory Word Programming

Factory word programming is similar to word programming in that it uses the same commands and programming algorithms. However, factory word programming enhances the programming performance with  $V_{PP} = V_{PPH}$ . This can enable faster programming times during OEM manufacturing processes. Factory word programming is not intended for extended use. See [Section 5.2, "Operating Conditions" on page 25](#) for limitations when  $V_{PP} = V_{PPH}$ .

*Note:* When  $V_{PP} = V_{PPL}$ , the device draws programming current from the  $V_{CC}$  supply. If  $V_{PP}$  is driven by a logic signal,  $V_{PPL}$  must remain above  $V_{PPL\ MIN}$  to program the device. When  $V_{PP} = V_{PPH}$ , the device draws programming current from the  $V_{PP}$  supply. [Figure 29, "Example VPP Supply Connections" on page 61](#) shows examples of device power supply configurations.

## 11.2 Buffered Programming

The device features a 32-word buffer to enable optimum programming performance. For Buffered Programming, data is first written to an on-chip write buffer. Then the buffer data is programmed into the flash memory array in buffer-size increments. This can improve system programming performance significantly over non-buffered programming.

When the Buffered Programming Setup command is issued (see [Section 9.2, "Device Commands" on page 45](#)), Status Register information is updated and reflects the availability of the buffer. SR[7] indicates buffer availability: if set, the buffer is available; if cleared, the buffer is not available. To retry, issue the Buffered Programming Setup command again, and re-check SR[7]. When SR[7] is set, the buffer is ready for loading. (see [Figure 40, "Buffer Program Flowchart" on page 81](#)).

On the next write, a word count is written to the device at the buffer address. This tells the device how many data words will be written to the buffer, up to the maximum size of the buffer.

On the next write, a device start address is given along with the first data to be written to the flash memory array. Subsequent writes provide additional device addresses and data. All data addresses must lie within the start address plus the word count. Optimum programming performance and lower power usage are obtained by aligning the starting address at the beginning of a 32-word boundary ( $A[4:0] = 0x00$ ). Crossing a 32-word boundary during programming will double the total programming time.

After the last data is written to the buffer, the Buffered Programming Confirm command must be issued to the original block address. The WSM begins to program buffer contents to the flash memory array. If a command other than the Buffered Programming Confirm command is written to the device, a command sequence error occurs and Status Register bits SR[7,5,4] are set. If an error occurs while writing to the array, the device stops programming, and Status Register bits SR[7,4] are set, indicating a programming failure.

When Buffered Programming has completed, additional buffer writes can be initiated by issuing another Buffered Programming Setup command and repeating the buffered program sequence. Buffered programming may be performed with  $V_{PP} = V_{PPL}$  or  $V_{PPH}$  (see [Section 5.2, "Operating Conditions" on page 25](#) for limitations when operating the device with  $V_{PP} = V_{PPH}$ ).

If an attempt is made to program past an erase-block boundary using the Buffered Program command, the device aborts the operation. This generates a command sequence error, and Status Register bits SR[5,4] are set.

If Buffered programming is attempted while  $V_{PP}$  is below  $V_{PPLK}$ , Status Register bits SR[4,3] are set. If any errors are detected that have set Status Register bits, the Status Register should be cleared using the Clear Status Register command.

## 11.3 Buffered Enhanced Factory Programming

Buffered Enhanced Factory Programming (BEFP) speeds up Multi-Level Cell (MLC) flash programming. The enhanced programming algorithm used in BEFP eliminates traditional programming elements that drive up overhead in device programmer systems.

BEFP consists of three phases: Setup, Program/Verify, and Exit (see [Figure 41, "BEFP Flowchart" on page 82](#)). It uses a write buffer to spread MLC program performance across 32 data words. Verification occurs in the same phase as programming to accurately program the flash memory cell to the correct bit state.

A single two-cycle command sequence programs the entire block of data. This enhancement eliminates three write cycles per buffer: two commands and the word count for each set of 32 data words. Host programmer bus cycles fill the device's write buffer followed by a status check. SR[0] indicates when data from the buffer has been programmed into sequential flash memory array locations.

Following the buffer-to-flash array programming sequence, the Write State Machine (WSM) increments internal addressing to automatically select the next 32-word array boundary. This aspect of BEFP saves host programming equipment the address-bus setup overhead.

With adequate continuity testing, programming equipment can rely on the WSM's internal verification to ensure that the device has programmed properly. This eliminates the external post-program verification and its associated overhead.

### 11.3.1 BEFP Requirements and Considerations

**Table 29: BEFP Requirements**

| Parameter/Issue   | Requirement   | Notes |
|-------------------|---|-------|
| Case Temperature  | $T_C = 25\text{ }^\circ\text{C} \pm 5\text{ }^\circ\text{C}$  |       |
| $V_{CC}$          | Within operating range  |       |
| $V_{PP}$          | Driven to $V_{PPH}$   |       |
| Setup and Confirm | Target block unlocked before issuing the BEFP Setup and Confirm commands  |       |
| Programming       | The first-word address (WA0) of the block to be programmed must be held constant from the setup phase through all data streaming into the target block, until transition to the exit phase is desired |       |
| Buffer Alignment  | WA0 must align with the start of an array buffer boundary   | 1     |

**Note:**

1. Word buffer boundaries in the array are determined by A[4:0] (0x00 through 0x1F). The alignment start point is A[4:0] = 0x00.

**Table 30: BEFP Considerations**

| Parameter/Issue                    | Requirement  | Notes |
|------------------------------------|--|-------|
| Cycling                            | For optimum performance, cycling must be limited below 100 erase cycles per block. | 1     |
| Programming blocks                 | BEFP programs one block at a time; all buffer data must fall within a single block | 2     |
| Suspend                            | BEFP cannot be suspended   |       |
| Programming the flash memory array | Programming to the flash memory array can occur only when the buffer is full.      | 3     |

**Note:**

1. Some degradation in performance may occur if this limit is exceeded, but the internal algorithm continues to work properly.
2. If the internal address counter increments beyond the block's maximum address, addressing wraps around to the beginning of the block.
3. If the number of words is less than 32, remaining locations must be filled with 0xFFFF.

**11.3.2 BEFP Setup Phase**

After receiving the BEFP Setup and Confirm command sequence, Status Register bit SR[7] (Ready) is cleared, indicating that the WSM is busy with BEFP algorithm startup. A delay before checking SR[7] is required to allow the WSM enough time to perform all of its setups and checks (Block-Lock status,  $V_{pp}$  level, etc.). If an error is detected, SR[4] is set and BEFP operation terminates. If the block was found to be locked, SR[1] is also set. SR[3] is set if the error occurred due to an incorrect  $V_{pp}$  level.

**Note:** Reading from the device after the BEFP Setup and Confirm command sequence outputs Status Register data. Do not issue the Read Status Register command; it will be interpreted as data to be loaded into the buffer.

**11.3.3 BEFP Program/Verify Phase**

After the BEFP Setup Phase has completed, the host programming system must check SR[7,0] to determine the availability of the write buffer for data streaming. SR[7] cleared indicates the device is busy and the BEFP program/verify phase is activated. SR[0] indicates the write buffer is available.

Two basic sequences repeat in this phase: loading of the write buffer, followed by buffer data programming to the array. For BEFP, the count value for buffer loading is always the maximum buffer size of 32 words. During the buffer-loading sequence, data is stored to sequential buffer locations starting at address 0x00. Programming of the buffer contents to the flash memory array starts as soon as the buffer is full. If the number of words is less than 32, the remaining buffer locations must be filled with 0xFFFF.

**Caution:** **The buffer must be completely filled for programming to occur. Supplying an address outside of the current block's range during a buffer-fill sequence causes the algorithm to exit immediately. Any data previously loaded into the buffer during the fill cycle is not programmed into the array.**

The starting address for data entry must be buffer size aligned, if not the BEFP algorithm will be aborted and the program fails and (SR[4]) flag will be set.

Data words from the write buffer are directed to sequential memory locations in the flash memory array; programming continues from where the previous buffer sequence ended. The host programming system must poll SR[0] to determine when the buffer program sequence completes. SR[0] cleared indicates that all buffer data has been transferred to the flash array; SR[0] set indicates that the buffer is not available yet for the next fill cycle. The host system may check full status for errors at any time, but it is

only necessary on a block basis after BEFP exit. After the buffer fill cycle, no write cycles should be issued to the device until SR[0] = 0 and the device is ready for the next buffer fill.

*Note:* Any spurious writes are ignored after a buffer fill operation and when internal program is proceeding.

The host programming system continues the BEFP algorithm by providing the next group of data words to be written to the buffer. Alternatively, it can terminate this phase by changing the block address to one outside of the current block's range.

The Program/Verify phase concludes when the programmer writes to a different block address; data supplied must be 0xFFFF. Upon Program/Verify phase completion, the device enters the BEFP Exit phase.

#### 11.3.4 BEFP Exit Phase

When SR[7] is set, the device has returned to normal operating conditions. A full status check should be performed at this time to ensure the entire block programmed successfully. When exiting the BEFP algorithm with a block address change, the read mode will not change. After BEFP exit, any valid command can be issued to the device.

### 11.4 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 (see [Figure 39, "Program Suspend/Resume Flowchart" on page 80](#)).

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. Suspend latency is specified in [Section 7.5, "Program and Erase Characteristics" on page 39](#).

To read data from the device, the Read Array command must be issued. Read Array, Read Status Register, Read Device Identifier, CFI Query, 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.

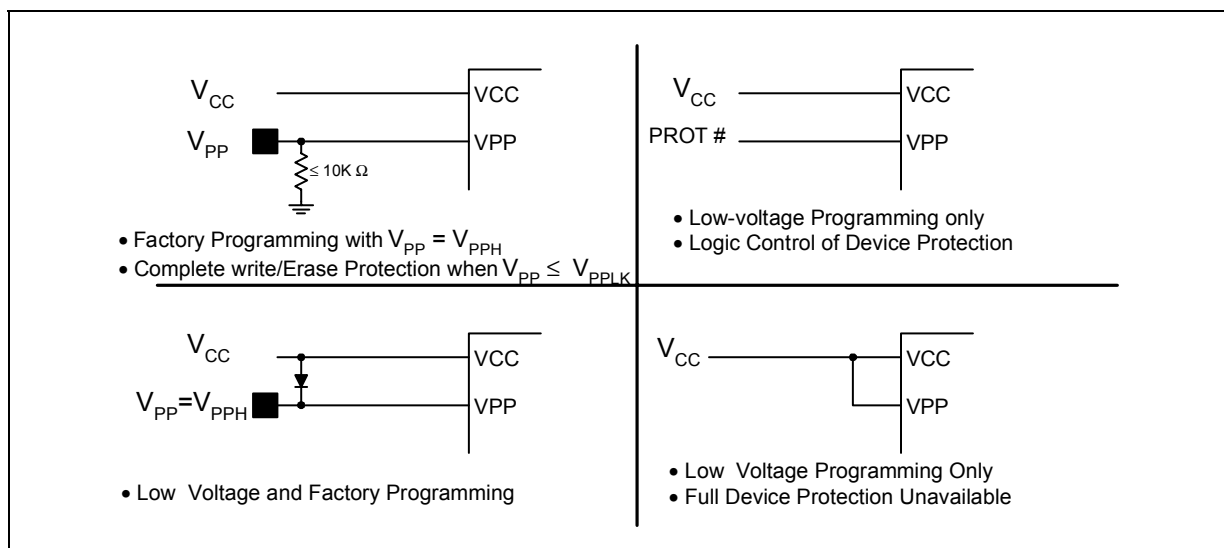
### 11.5 Program Resume

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 (see [Figure 39, "Program Suspend/Resume Flowchart" on page 80](#)).

## 11.6 Program Protection

When  $V_{PP} = V_{IL}$ , absolute hardware write protection is provided for all device blocks. If  $V_{PP}$  is at or below  $V_{PPLK}$ , programming operations halt and SR[3] is set indicating a  $V_{PP}$ -level error. Block lock registers are not affected by the voltage level on  $V_{PP}$ ; they may still be programmed and read, even if  $V_{PP}$  is less than  $V_{PPLK}$ .

**Figure 29: Example VPP Supply Connections**



## 12.0 Erase Operations

Flash erasing is performed on a block basis. An entire block is erased each time an erase command sequence is issued, and only one block is erased at a time. When a block is erased, all bits within that block read as logical ones. The following sections describe block erase operations in detail.

### 12.1 Block Erase

Block erase operations are initiated by writing the Block Erase Setup command to the address of the block to be erased (see [Section 9.2, "Device Commands" on page 45](#)). Next, the Block Erase Confirm command is written to the address of the block to be erased. If the device is placed in standby (CE# deasserted) during an erase operation, the device completes the erase operation before entering standby.  $V_{PP}$  must be above  $V_{PPLK}$  and the block must be unlocked (see [Figure 42, "Block Erase Flowchart" on page 83](#)).

During a block erase, the Write State Machine (WSM) executes a sequence of internally-timed events that conditions, erases, and verifies all bits within the block. Erasing the flash memory array changes "zeros" to "ones". Memory array bits that are ones can be changed to zeros only by programming the block (see [Section 11.0, "Programming Operations" on page 56](#)).

The Status Register can be examined for block erase progress and errors by reading any address. The device remains in the Read Status Register state until another command is written. SR[0] indicates whether the addressed block is erasing. Status Register bit SR[7] is set upon erase completion.

Status Register bit SR[7] indicates block erase status while the sequence executes. When the erase operation has finished, Status Register bit SR[5] indicates an erase failure if set. SR[3] set would indicate that the WSM could not perform the erase operation because  $V_{PP}$  was outside of its acceptable limits. SR[1] set indicates that the erase operation attempted to erase a locked block, causing the operation to abort.

Before issuing a new command, the Status Register contents should be examined and then cleared using the Clear Status Register command. Any valid command can follow once the block erase operation has completed.

### 12.2 Erase Suspend

Issuing the Erase Suspend command while erasing suspends the block erase operation. This allows data to be accessed from memory locations other than the one being erased. The Erase Suspend command can be issued to any device address. A block erase operation can be suspended to perform a word or buffer program operation, or a read operation within any block except the block that is erase suspended (see [Figure 39, "Program Suspend/Resume Flowchart" on page 80](#)).

When a block erase operation is executing, issuing the Erase Suspend command requests the WSM to suspend the erase algorithm at predetermined points. The device continues to output Status Register data after the Erase Suspend command is issued. Block erase is suspended when Status Register bits SR[7,6] are set. Suspend latency is specified in [Section 7.5, "Program and Erase Characteristics" on page 39](#).

To read data from the device (other than an erase-suspended block), the Read Array command must be issued. During Erase Suspend, a Program command can be issued to any block other than the erase-suspended block. Block erase cannot resume until program operations initiated during erase suspend complete. Read Array, Read Status Register, Read Device Identifier, CFI Query, and Erase Resume are valid commands

during Erase Suspend. Additionally, Clear Status Register, Program, Program Suspend, Block Lock, Block Unlock, and Block Lock-Down are valid commands during Erase Suspend.

During an erase suspend, deasserting CE# places the device in standby, reducing active current.  $V_{pp}$  must remain at a valid level, and WP# must remain unchanged while in erase suspend. If RST# is asserted, the device is reset.

### 12.3 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 39, "Program Suspend/Resume Flowchart" on page 80](#)).

### 12.4 Erase Protection

When  $V_{pp} = V_{IL}$ , absolute hardware erase protection is provided for all device blocks. If  $V_{pp}$  is below  $V_{PPLK}$ , erase operations halt and SR[3] is set indicating a  $V_{pp}$ -level error.

## 13.0 Security Modes

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.

### 13.1 Block Locking

Individual instant block locking is used to protect user code and/or data within the flash memory array. All blocks power up in a locked state to protect array data from being altered during power transitions. Any block can be locked or unlocked with no latency. Locked blocks cannot be programmed or erased; they can only be read.

Software-controlled security is implemented using the Block Lock and Block Unlock commands. Hardware-controlled security can be implemented using the Block Lock-Down command along with asserting WP#. Also,  $V_{PP}$  data security can be used to inhibit program and erase operations (see [Section 11.6, "Program Protection" on page 61](#) and [Section 12.4, "Erase Protection" on page 63](#)).

The P30 device also offers four pre-defined areas in the main array that can be configured as One-Time Programmable (OTP) for the highest level of security. These include the four 32 KB parameter blocks together as one and the three adjacent 128 KB main blocks. This is available for top or bottom parameter devices.

#### 13.1.1 Lock Block

To lock a block, issue the Lock Block Setup command. The next command must be the Lock Block command issued to the desired block's address (see [Section 9.2, "Device Commands" on page 45](#) and [Figure 44, "Block Lock Operations Flowchart" on page 85](#)). If the Set Read Configuration Register command is issued after the Block Lock Setup command, the device configures the RCR instead.

Block lock and unlock operations are not affected by the voltage level on  $V_{PP}$ . The block lock bits may be modified and/or read even if  $V_{PP}$  is at or below  $V_{PPLK}$ .

#### 13.1.2 Unlock Block

The Unlock Block command is used to unlock blocks (see [Section 9.2, "Device Commands" on page 45](#)). Unlocked blocks can be read, programmed, and erased. Unlocked blocks return to a locked state when the device is reset or powered down. If a block is in a lock-down state, WP# must be deasserted before it can be unlocked (see [Figure 30, "Block Locking State Diagram" on page 65](#)).

#### 13.1.3 Lock-Down Block

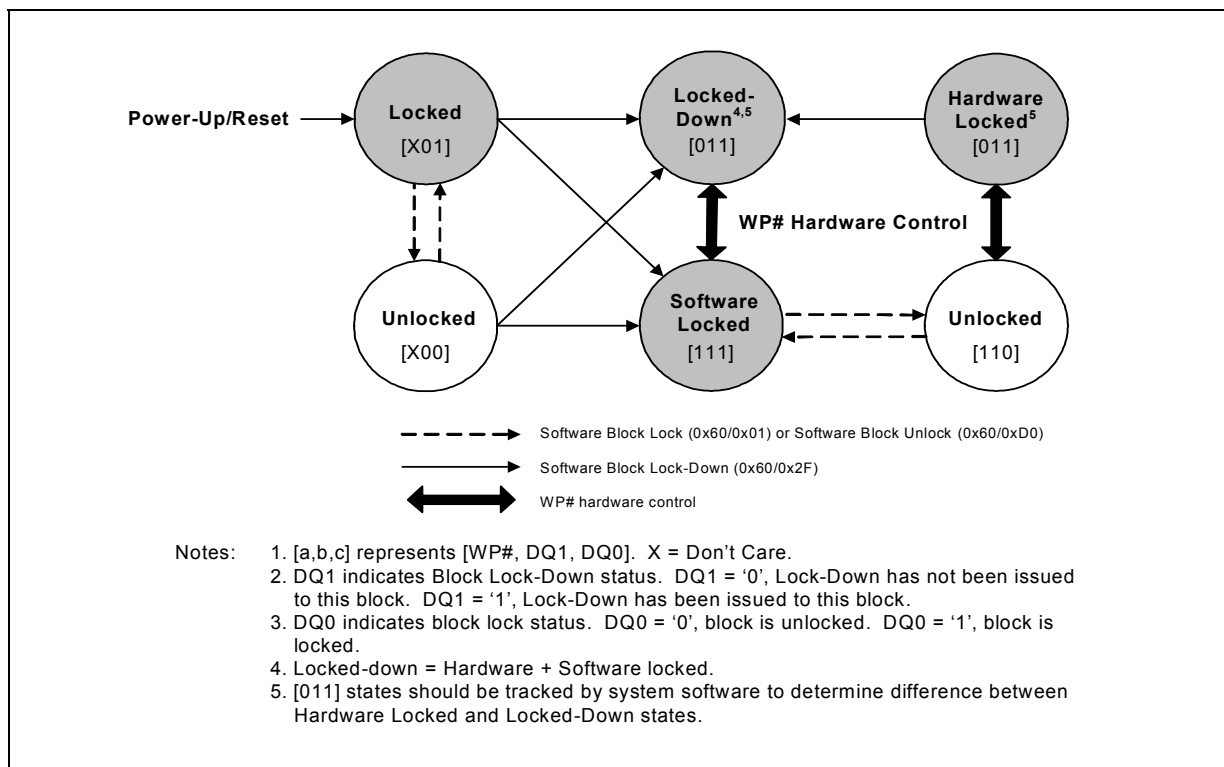
A locked or unlocked block can be locked-down by writing the Lock-Down Block command sequence (see [Section 9.2, "Device Commands" on page 45](#)). Blocks in a lock-down state cannot be programmed or erased; they can only be read. However, unlike locked blocks, their locked state cannot be changed by software commands alone. A locked-down block can only be unlocked by issuing the Unlock Block command with WP# deasserted. To return an unlocked block to locked-down state, a Lock-Down command must be issued prior to changing WP# to  $V_{IL}$ . Locked-down blocks revert to the locked state upon reset or power up the device (see [Figure 30, "Block Locking State Diagram" on page 65](#)).



### 13.1.4 Block Lock Status

The Read Device Identifier command is used to determine a block's lock status (see Section 14.2, "Read Device Identifier" on page 70). Data bits DQ[1:0] display the addressed block's lock status; DQ0 is the addressed block's lock bit, while DQ1 is the addressed block's lock-down bit.

Figure 30: Block Locking State Diagram



### 13.1.5 Block Locking During Suspend

Block lock and unlock changes can be performed during an erase suspend. To change block locking during an erase operation, first issue the Erase Suspend command. Monitor the Status Register until SR[7] and SR[6] are set, indicating the device is suspended and ready to accept another command.

Next, write the desired lock command sequence to a block, which changes the lock state of that block. After completing block lock or unlock operations, resume the erase operation using the Erase Resume command.

**Note:** A Lock Block Setup command followed by any command other than Lock Block, Unlock Block, or Lock-Down Block produces a command sequence error and set Status Register bits SR[4] and SR[5]. If a command sequence error occurs during an erase suspend, SR[4] and SR[5] remains set, even after the erase operation is resumed. Unless the Status Register is cleared using the Clear Status Register command before resuming the erase operation, possible erase errors may be masked by the command sequence error.

If a block is locked or locked-down during an erase suspend of the *same* block, the lock status bits change immediately. However, the erase operation completes when it is resumed. Block lock operations cannot occur during a program suspend. See [Appendix A, "Write State Machine" on page 72](#), which shows valid commands during an erase suspend.

## 13.2 Selectable One-Time Programmable Blocks

Any of four pre-defined areas from the main array (the four 32-KB parameter blocks together as one and three adjacent 128 KB main blocks) can be configured as OTP so further program and erase operations are not allowed. This option is available for top or bottom parameter devices.

**Table 31: Selectable OTP Block Mapping**

| Density  | Top Parameter Configuration | Bottom Parameter Configuration |
|----------|-----------------------------|--------------------------------|
| 256-Mbit | blocks 258:255 (parameters) | blocks 3:0 (parameters)        |
|          | block 254 (main)            | block 4 (main)                 |
|          | block 253 (main)            | block 5 (main)                 |
|          | block 252 (main)            | block 6 (main)                 |
| 128-Mbit | blocks 130:127 (parameters) | blocks 3:0 (parameters)        |
|          | block 126 (main)            | block 4 (main)                 |
|          | block 125 (main)            | block 5 (main)                 |
|          | block 124 (main)            | block 6 (main)                 |
| 64-Mbit  | blocks 66:63 (parameters)   | blocks 3:0 (parameters)        |
|          | block 62 (main)             | block 4 (main)                 |
|          | block 61 (main)             | block 5 (main)                 |
|          | block 60 (main)             | block 6 (main)                 |

**Notes:**

1. The 512-Mbit devices will have multiple die and selectable OTP areas depending on the placement of the parameter blocks.
2. When programming the OTP bits for a **Top Parameter Device**, the following upper address bits must also be driven properly: A[Max:17] driven high ( $V_{IH}$ ) for TSOP and Easy BGA packages, and A[Max:16] driven high ( $V_{IH}$ ) for QUAD+SCSP.

**Note:** Please see your local Numonyx representative for details about the Selectable OTP implementation.

## 13.3 Protection Registers

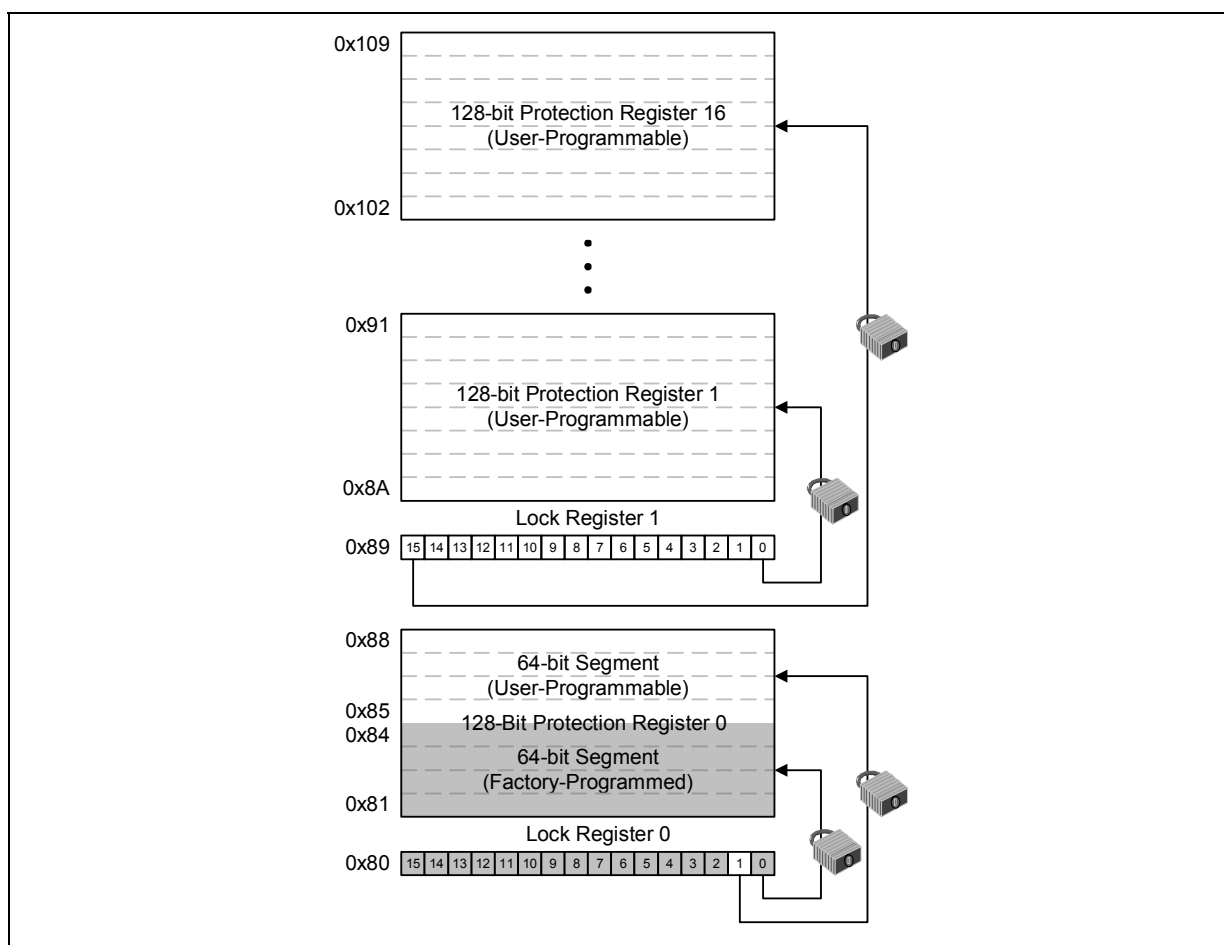
The device contains 17 Protection Registers (PRs) that can be used to implement system security measures and/or device identification. Each Protection Register can be individually locked.

The first 128-bit Protection Register is comprised of two 64-bit (8-word) segments. The lower 64-bit segment is pre-programmed at the Numonyx factory with a unique 64-bit number. The other 64-bit segment, as well as the other sixteen 128-bit Protection Registers, are blank. Users can program these registers as needed. When programmed, users can then lock the Protection Register(s) to prevent additional bit programming (see [Figure 31, "Protection Register Map" on page 67](#)).

The user-programmable Protection Registers contain one-time programmable (OTP) bits; when programmed, register bits cannot be erased. Each Protection Register can be accessed multiple times to program individual bits, as long as the register remains unlocked.

Each Protection Register has an associated Lock Register bit. When a Lock Register bit is programmed, the associated Protection Register can only be read; it can no longer be programmed. Additionally, because the Lock Register bits themselves are OTP, when programmed, Lock Register bits cannot be erased. Therefore, when a Protection Register is locked, it cannot be unlocked.

**Figure 31: Protection Register Map**



### 13.3.1 Reading the Protection Registers

The Protection Registers can be read from any address. To read the Protection Register, first issue the Read Device Identifier command at any address to place the device in the Read Device Identifier state (see Section 9.2, "Device Commands" on page 45). Next, perform a read operation using the address offset corresponding to the register to be read. Table 33, "Device Identifier Information" on page 70 shows the address offsets of the Protection Registers and Lock Registers. Register data is read 16 bits at a time.

### 13.3.2 Programming the Protection Registers

To program any of the Protection Registers, first issue the Program Protection Register command at the parameter's base address plus the offset to the desired Protection Register (see [Section 9.2, "Device Commands" on page 45](#)). Next, write the desired Protection Register data to the same Protection Register address (see [Figure 31, "Protection Register Map" on page 67](#)).

The device programs the 64-bit and 128-bit user-programmable Protection Register data 16 bits at a time (see [Figure 45, "Protection Register Programming Flowchart" on page 86](#)). Issuing the Program Protection Register command outside of the Protection Register's address space causes a program error (SR[4] set). Attempting to program a locked Protection Register causes a program error (SR[4] set) and a lock error (SR[1] set).

**Note:** When programming the OTP bits for a **Top Parameter Device**, the following upper address bits must also be driven properly: A[Max:17] driven high ( $V_{IH}$ ) for TSOP and Easy BGA packages, and A[Max:16] driven high ( $V_{IH}$ ) for QUAD+ SCSP.

### 13.3.3 Locking the Protection Registers

Each Protection Register can be locked by programming its respective lock bit in the Lock Register. To lock a Protection Register, program the corresponding bit in the Lock Register by issuing the Program Lock Register command, followed by the desired Lock Register data (see [Section 9.2, "Device Commands" on page 45](#)). The physical addresses of the Lock Registers are 0x80 for register 0 and 0x89 for register 1. These addresses are used when programming the lock registers (see [Table 33, "Device Identifier Information" on page 70](#)).

Bit 0 of Lock Register 0 is already programmed at the factory, locking the lower, pre-programmed 64-bit region of the first 128-bit Protection Register containing the unique identification number of the device. Bit 1 of Lock Register 0 can be programmed by the user to lock the user-programmable, 64-bit region of the first 128-bit Protection Register. When programming Bit 1 of Lock Register 0, all other bits need to be left as '1' such that the data programmed is 0xFFFD.

Lock Register 1 controls the locking of the upper sixteen 128-bit Protection Registers. Each of the 16 bits of Lock Register 1 correspond to each of the upper sixteen 128-bit Protection Registers. Programming a bit in Lock Register 1 locks the corresponding 128-bit Protection Register.

**Caution:** **After being locked, the Protection Registers cannot be unlocked.**

## 14.0 Special Read States

The following sections describe non-array read states. Non-array reads can be performed in asynchronous read or synchronous burst mode. A non-array read operation occurs as asynchronous single-word mode. When non-array reads are performed in asynchronous page mode only the first data is valid and all subsequent data are undefined. When a non-array read operation occurs as synchronous burst mode, the same word of data requested will be output on successive clock edges until the burst length requirements are satisfied.

Refer to the following waveforms for more detailed information:

- [Figure 14, "Asynchronous Single-Word Read \(ADV# Low\)" on page 33](#)
- [Figure 15, "Asynchronous Single-Word Read \(ADV# Latch\)" on page 33](#)
- [Figure 17, "Synchronous Single-Word Array or Non-array Read Timing" on page 34](#)

### 14.1 Read Status Register

To read the Status Register, issue the Read Status Register command at any address. Status Register information is available to which the Read Status Register, Word Program, or Block Erase command was issued. Status Register data is automatically made available following a Word Program, Block Erase, or Block Lock command sequence. Reads from the device after any of these command sequences outputs the device's status until another valid command is written (e.g. Read Array command).

The Status Register is read using single asynchronous-mode or synchronous burst mode reads. Status Register data is output on DQ[7:0], while 0x00 is output on DQ[15:8]. In asynchronous mode the falling edge of OE#, or CE# (whichever occurs first) updates and latches the Status Register contents. However, reading the Status Register in synchronous burst mode, CE# or ADV# must be toggled to update status data.

The Device Write Status bit (SR[7]) provides overall status of the device. Status register bits SR[6:1] present status and error information about the program, erase, suspend, V<sub>pp</sub>, and block-locked operations.

**Table 32: Status Register Description (Sheet 1 of 2)**

| Status Register (SR) |                            |              |   |                        |                        |                     | Default Value = 0x80 |
|----------------------|----------------------------|--------------|---|------------------------|------------------------|---------------------|----------------------|
| Device Write Status  | Erase Suspend Status       | Erase Status | Program Status  | V <sub>pp</sub> Status | Program Suspend Status | Block-Locked Status | BEFP Status          |
| DWS                  | ESS                        | ES           | PS  | VPPS                   | PSS                    | BLS                 | BWS                  |
| 7                    | 6                          | 5            | 4   | 3                      | 2                      | 1                   | 0                    |
| Bit                  | Name                       |              | Description   |                        |                        |                     |                      |
| 7                    | Device Write Status (DWS)  |              | 0 = Device is busy; program or erase cycle in progress; SR[0] valid.<br>1 = Device is ready; SR[6:1] are valid. |                        |                        |                     |                      |
| 6                    | Erase Suspend Status (ESS) |              | 0 = Erase suspend not in effect.<br>1 = Erase suspend in effect.  |                        |                        |                     |                      |
| 5                    | Erase Status (ES)          |              | 0 = Erase successful.<br>1 = Erase fail or program sequence error when set with SR[4,7].                        |                        |                        |                     |                      |
| 4                    | Program Status (PS)        |              | 0 = Program successful.<br>1 = Program fail or program sequence error when set with SR[5,7]                     |                        |                        |                     |                      |

**Table 32: Status Register Description (Sheet 2 of 2)**

| Status Register (SR) |                               | Default Value = 0x80   |
|----------------------|-------------------------------|--|
| 3                    | V <sub>pp</sub> Status (VPPS) | 0 = VPP within acceptable limits during program or erase operation.<br>1 = VPP < VPP <sub>PLK</sub> during program or erase operation. |
| 2                    | Program Suspend Status (PSS)  | 0 = Program suspend not in effect.<br>1 = Program suspend in effect.   |
| 1                    | Block-Locked Status (BLS)     | 0 = Block not locked during program or erase.<br>1 = Block locked during program or erase; operation aborted.                          |
| 0                    | BEFP Status (BWS)             | After Buffered Enhanced Factory Programming (BEFP) data is loaded into the buffer:<br>0 = BEFP complete.<br>1 = BEFP in-progress.      |

**Note:** Always clear the Status Register prior to resuming erase operations. It avoids Status Register ambiguity when issuing commands during Erase Suspend. If a command sequence error occurs during an erase-suspend state, the Status Register contains the command sequence error status (SR[7,5,4] set). When the erase operation resumes and finishes, possible errors during the erase operation cannot be detected via the Status Register because it contains the previous error status.

### 14.1.1 Clear Status Register

The Clear Status Register command clears the status register. It functions independent of V<sub>pp</sub>. The Write State Machine (WSM) sets and clears SR[7,6,2], but it sets bits SR[5:3,1] without clearing them. The Status Register should be cleared before starting a command sequence to avoid any ambiguity. A device reset also clears the Status Register.

## 14.2 Read Device Identifier

The Read Device Identifier command instructs the device to output manufacturer code, device identifier code, block-lock status, protection register data, or configuration register data (see [Section 9.2, "Device Commands" on page 45](#) for details on issuing the Read Device Identifier command). [Table 33, "Device Identifier Information" on page 70](#) and [Table 34, "Device ID codes" on page 71](#) show the address offsets and data values for this device.

**Table 33: Device Identifier Information (Sheet 1 of 2)**

| Item  | Address <sup>(1)</sup> | Data  |
|---|------------------------|---|
| Manufacturer Code   | 0x00                   | 0089h   |
| Device ID Code  | 0x01                   | ID (see <a href="#">Table 34</a> )  |
| Block Lock Configuration:<br>• Block Is Unlocked<br>• Block Is Locked<br>• Block Is not Locked-Down<br>• Block Is Locked-Down | BBA + 0x02             | Lock Bit:<br>DQ <sub>0</sub> = 0b0<br>DQ <sub>0</sub> = 0b1<br>DQ <sub>1</sub> = 0b0<br>DQ <sub>1</sub> = 0b1 |
| Read Configuration Register   | 0x05                   | RCR Contents  |
| Lock Register 0   | 0x80                   | PR-LK0  |
| 64-bit Factory-Programmed Protection Register   | 0x81-0x84              | Factory Protection Register Data  |
| 64-bit User-Programmable Protection Register  | 0x85-0x88              | User Protection Register Data   |

**Table 33: Device Identifier Information (Sheet 2 of 2)**

| Item   | Address <sup>(1)</sup> | Data                     |
|--|------------------------|--------------------------|
| Lock Register 1                                | 0x89                   | PR-LK1                   |
| 128-bit User-Programmable Protection Registers | 0x8A-0x109             | Protection Register Data |

**Notes:**

1. BBA = Block Base Address.

**Table 34: Device ID codes**

| ID Code Type | Device Density | Device Identifier Codes |                          |
|--------------|----------------|-------------------------|--------------------------|
|              |                | -T<br>(Top Parameter)   | -B<br>(Bottom Parameter) |
| Device Code  | 64-Mbit        | 8817                    | 881A                     |
|              | 128-Mbit       | 8818                    | 881B                     |
|              | 256-Mbit       | 8919                    | 891C                     |

**Note:** The 512-Mbit devices do not have a Device ID associated with them. Each die within the stack can be identified by either of the 256-Mbit Device ID codes depending on its parameter option.

## 14.3 CFI Query

The CFI Query command instructs the device to output Common Flash Interface (CFI) data when read. See [Section 9.2, "Device Commands" on page 45](#) for details on issuing the CFI Query command. [Appendix C, "Common Flash Interface" on page 87](#) shows CFI information and address offsets within the CFI database.

## Appendix A Write State Machine

Figure 32 through Figure 37 show the command state transitions (Next State Table) based on incoming commands. Only one partition can be actively programming or erasing at a time. Each partition stays in its last read state (Read Array, Read Device ID, CFI Query or Read Status Register) until a new command changes it. The next WSM state does not depend on the partition's output state.

**Figure 32: Write State Machine—Next State Table (Sheet 1 of 6)**

| Command Input to Chip and resulting Chip Next State |                                    |   |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
|---|------------------------------------|---|-------------------------------------|---------------------------------------|--|--|-----------------------------------|----------------------|---|-----------------------------|---|--|
| Current Chip State <sup>(7)</sup>                   | Read Array <sup>(2)</sup><br>(FFH) | Word Program <sup>(3,4)</sup><br>(10H/40H)                              | Buffered Program (BP)<br>(E8H)      | Erase Setup <sup>(3,4)</sup><br>(20H) | Buffered Enhanced Factory Pgm Setup <sup>(3, 4)</sup><br>(80H) | BE Confirm, P/E Resume, ULB, Confirm <sup>(8)</sup><br>(D0H) | BP / Prg / Erase Suspend<br>(B0H) | Read Status<br>(70H) | Clear Status Register <sup>(5)</sup><br>(50H) | Read ID/Query<br>(90H, 98H) | Lock, Unlock, Lock-down, CR setup <sup>(4)</sup><br>(60H) |  |
| Ready   | Ready                              | Program Setup   | BP Setup                            | Erase Setup                           | BEFP Setup   | Ready  |                                   |                      |   | Lock/CR Setup               |   |  |
| Lock/CR Setup                                       | Ready (Lock Error)                 |   |                                     |                                       |  | Ready (Unlock Block)   | Ready (Lock Error)                |                      |   |                             |   |  |
| OTP   | Setup                              | OTP Busy  |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
|   | Busy                               | OTP Busy  |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
| Word Program  | Setup                              | Word Program Busy   |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
|   | Busy                               | Program Busy  |                                     |                                       |  |  | Word Program Suspend              | Word Program Busy    |   |                             |   |  |
|   | Suspend                            | Word Program Suspend  |                                     |                                       |  |  | Word Program Busy                 | Word Program Suspend |   |                             |   |  |
| BP  | Setup                              | BP Load 1   |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
|   | BP Load 1                          | BP Load 2   |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
|   | BP Load 2                          | BP Confirm if Data load into Program Buffer is complete; Else BP Load 2 |                                     |                                       |  |  |                                   |                      |   |                             |   |  |
|   | BP Confirm                         | Ready (Error)   |                                     |                                       |  |  | BP Busy                           | Ready (Error)        |   |                             |   |  |
|   | BP Busy                            | BP Busy   |                                     |                                       |  |  | BP Suspend                        | BP Busy              |   |                             |   |  |
|   | BP Suspend                         | BP Suspend  |                                     |                                       |  |  | BP Busy                           | BP Suspend           |   |                             |   |  |
| Erase   | Setup                              | Ready (Error)   |                                     |                                       |  |  | Erase Busy                        | Ready (Error)        |   |                             |   |  |
|   | 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                            |  |



Figure 33: Write State Machine—Next State Table (Sheet 2 of 6)

| Current Chip State <sup>(7)</sup>      |                            | Command Input to Chip and resulting <i>Chip</i> Next State  |  |                                |                                       |   |  |   |                      |   |                             |
|--|----------------------------|---|--|--------------------------------|---------------------------------------|---|--|---|----------------------|---|-----------------------------|
|  |                            | Read Array <sup>(2)</sup><br>(FFH)  | Word Program <sup>(3,4)</sup><br>(10H/40H) | Buffered Program (BP)<br>(E8H) | Erase Setup <sup>(3,4)</sup><br>(20H) | Buffered Enhanced Factory Pgm Setup <sup>(3,4)</sup><br>(80H) | BE Confirm, P/E Resume, ULB, Confirm <sup>(6)</sup><br>(D0H) | BP / Prg / Erase Suspend<br>(B0H)       | Read Status<br>(70H) | Clear Status Register <sup>(5)</sup><br>(50H) | Read ID/Query<br>(90H, 98H) |
| 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 Program Busy in Erase Suspend                            | Word Program Suspend in Erase Suspend                        |   |                      |   |                             |
| BP in Erase Suspend                    | Setup                      | BP Load 1   |  |                                |                                       |   |  |   |                      |   |                             |
|  | BP Load 1                  | BP Load 2   |  |                                |                                       |   |  |   |                      |   |                             |
|  | BP Load 2                  | BP Confirm if Data load into Program Buffer is complete; Else BP Load 2   |  |                                |                                       |   |  |   |                      |   |                             |
|  | BP Confirm                 | Erase Suspend (Error)   |  |                                |                                       | BP Busy in Erase Suspend                                      | Ready (Error 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) |   |  |                                | Erase Suspend (Unlock Block)          | Erase Suspend (Lock Error [Botch])                            |  |   |                      |   |                             |
| Buffered Enhanced Factory Program Mode | Setup                      | Ready (Error)   |  |                                |                                       | BEFP Loading Data (X=32)                                      | Ready (Error)  |   |                      |   |                             |
|  | BEFP Busy                  | BEFP Program and Verify Busy (if Block Address given matches address given on BEFP Setup command). Commands treated as data. <sup>(7)</sup> |  |                                |                                       |   |  |   |                      |   |                             |

Figure 34: Write State Machine—Next State Table (Sheet 3 of 6)

| Current Chip State <sup>(7)</sup> |               | Command Input to Chip and resulting <i>Chip</i> Next State              |  |   |   |   |   |                         |
|-----------------------------------|---------------|---|--|---|---|---|---|-------------------------|
|                                   |               | OTP Setup <sup>(4)</sup><br>(C0H)                                       | Lock Block Confirm <sup>(8)</sup><br>(01H) | Lock-Down Block Confirm <sup>(8)</sup><br>(2FH) | Write RCR Confirm <sup>(8)</sup><br>(03H)         | Block Address (?WA0) <sup>9</sup><br>(XXXXH)                            | Illegal Cmds or BEFP Data <sup>(1)</sup><br>(all other codes) | WSM Operation Completes |
| Ready                             | OTP Setup     | Ready   |  |   |   |   | N/A   |                         |
|                                   | Lock/CR Setup | Ready (Lock Error)  | Ready (Lock Block)                         | Ready (Lock Down Blk)                           | Ready (Set CR)                                    | Ready (Lock Error)  |   |                         |
| OTP                               | Setup         | OTP Busy  |  |   |   |   | Ready   |                         |
|                                   | Busy          |   |  |   |   |   |   |                         |
| Word Program                      | Setup         | Word Program Busy   |  |   |   |   | N/A   |                         |
|                                   | Busy          | Word Program Busy   |  |   |   |   |   |                         |
|                                   | Suspend       | Word Program Suspend  |  |   |   |   |   |                         |
| BP                                | Setup         | BP Load 1   |  |   |   |   | N/A   |                         |
|                                   | BP Load 1     | BP Load 2   |  |   | Ready (BP Load 2)                                 | BP Load 2   |   |                         |
|                                   | BP Load 2     | BP Confirm if Data load into Program Buffer is complete; ELSE BP load 2 |  |   | Ready   | BP Confirm if Data load into Program Buffer is complete; ELSE BP Load 2 |   |                         |
|                                   | BP Confirm    | Ready (Error)   |  |   | Ready (Error) (Proceed if unlocked or lock error) | Ready (Error)   |   |                         |
|                                   | BP Busy       | BP Busy   |  |   |   |   |   | Ready                   |
|                                   | BP Suspend    | BP Suspend  |  |   |   |   |   |                         |
| Erase                             | Setup         | Ready (Error)   |  |   |   |   | N/A   |                         |
|                                   | Busy          | Erase Busy  |  |   |   |   |   |                         |
|                                   | Suspend       | Erase Suspend   |  |   |   |   |   |                         |

Figure 35: Write State Machine—Next State Table (Sheet 4 of 6)

| Current Chip State <sup>(7)</sup>      |                            | Command Input to Chip and resulting <i>Chip</i> Next State  |  |   |   |   |   |                         |
|--|----------------------------|---|--|---|---|---|---|-------------------------|
|  |                            | OTP Setup <sup>(4)</sup><br>(C0H)   | Lock Block Confirm <sup>(8)</sup><br>(01H) | Lock-Down Block Confirm <sup>(8)</sup><br>(2FH) | Write RCR Confirm <sup>(8)</sup><br>(03H)         | Block Address (?WAO) <sup>9</sup><br>(XXXXH)                            | Illegal Cmds or BEFP Data <sup>(1)</sup><br>(all other codes) | WSM Operation Completes |
| Word Program in Erase Suspend          | Setup                      | Word Program Busy in Erase Suspend  |  |   |   |   | NA  |                         |
|  | Busy                       | Word Program Busy in Erase Suspend Busy   |  |   |   |   | Erase Suspend   |                         |
|  | Suspend                    | Word Program Suspend in Erase Suspend   |  |   |   |   | N/A   |                         |
| BP in Erase Suspend                    | Setup                      | BP Load 1   |  |   |   |   |   |                         |
|  | BP Load 1                  | BP Load 2   |  |   | Ready (BP Load 2)                                 | BP Load 2   |   |                         |
|  | BP Load 2                  | BP Confirm if Data load into Program Buffer is complete; Else BP Load 2   |  |   | Ready   | BP Confirm if Data load into Program Buffer is complete; Else BP Load 2 |   |                         |
|  | BP Confirm                 | Ready (Error in Erase Suspend)  |  |   | Ready (Error) (Proceed if unlocked or lock error) | Ready (Error)   |   |                         |
|  | BP Busy                    | BP Busy in Erase Suspend  |  |   |   |   | Erase Suspend   |                         |
|  | BP Suspend                 | BP Suspend in Erase Suspend   |  |   |   |   | N/A   |                         |
| Lock/CR Setup in Erase Suspend         | Erase Suspend (Lock Error) | Erase Suspend (Lock Block)  | Erase Suspend (Lock Down Block)            | Erase Suspend (Set CR)                          | Erase Suspend (Lock Error)                        |   |   | N/A                     |
| Buffered Enhanced Factory Program Mode | Setup                      | Ready (Error)   |  |   | Ready (BEFP Loading Data)                         | Ready (Error)   |   |                         |
|  | BEFP Busy                  | BEFP Program and Verify Busy (if Block Address given matches address given on BEFP Setup command). Commands treated as data. <sup>(7)</sup> |  |   | Ready   | BEFP Busy   |   | Ready                   |

Figure 36: Write State Machine—Next State Table (Sheet 5 of 6)

| Output Next State Table  |  |                          |             |                              |   |  |                        |             |                                      |               |  |
|--|--|--------------------------|-------------|------------------------------|---|--|------------------------|-------------|--------------------------------------|---------------|--|
| Current chip state   | Command Input to Chip and resulting <i>Output</i> Mux Next State |                          |             |                              |   |  |                        |             |                                      |               |  |
|  | Read Array <sup>(2)</sup>  | Word Program Setup (3,4) | BP Setup    | Erase Setup <sup>(3,4)</sup> | Buffered Enhanced Factory Pgm Setup <sup>(3, 4)</sup> | BE Confirm, P/E Resume, ULB Confirm <sup>(8)</sup> | Program/ Erase Suspend | Read Status | Clear Status Register <sup>(5)</sup> | Read ID/Query | Lock, Unlock, Lock-down, CR setup <sup>(4)</sup> |
|  | (FFH)  | (10H/40H)                | (E8H)       | (20H)                        | (30H)   | (D0H)  | (B0H)                  | (70H)       | (50H)                                | (90H, 98H)    | (60H)  |
| BEFP Setup, BEFP Pgm & Verify Busy, Erase Setup, OTP Setup, BP: Setup, Load 1, Load 2, Confirm, Word Pgm Setup, Word Pgm Setup in Erase Susp, BP Setup, Load 1, Load 2, Confirm in Erase Suspend | Status Read  |                          |             |                              |   |  |                        |             |                                      |               |  |
| Lock/CR Setup, Lock/CR Setup in Erase Susp   | Status Read  |                          |             |                              |   |  |                        |             |                                      |               |  |
| OTP Busy   | Read Array   | Status Read              | Status Read | Output does not change.      | Status Read   | Output mux does not change.                        | ID Read                | Status Read | Status Read                          |               |  |
| 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      |  |                          |             |                              |   |  |                        |             |                                      |               |  |

Figure 37: Write State Machine—Next State Table (Sheet 6 of 6)

| Output Next State Table   |  |                                   |  |                                 |                         |  |                         |
|---|--|-----------------------------------|--|---------------------------------|-------------------------|--|-------------------------|
| Current chip state  | Command Input to Chip and resulting <i>Output Mux</i> Next State |                                   |  |                                 |                         |  |                         |
|   | OTP Setup <sup>(4)</sup>   | Lock Block Confirm <sup>(8)</sup> | Lock-Down Block Confirm <sup>(8)</sup> | Write CR Confirm <sup>(8)</sup> | Block Address (?WA0)    | Illegal Cmds or BEFP Data <sup>(1)</sup> | WSM Operation Completes |
|   | (C0H)  | (01H)                             | (2FH)                                  | (03H)                           | (FFFFH)                 | (all other codes)                        |                         |
| BEFP Setup, BEFP Pgm & Verify Busy, Erase Setup, OTP Setup, BP: Setup, Load 1, Load 2, Confirm, Word Pgm Setup, Word Pgm Setup in Erase Susp, BP Setup, Load1, Load 2, Confirm in Erase Suspend | Status Read  |                                   |  |                                 |                         |  | Output does not change. |
| Lock/CR Setup, Lock/CR Setup in Erase Susp  | Status Read  |                                   | Array Read                             | Status Read                     |                         |  |                         |
| OTP Busy  | Status Read  | Output does not change.           |  | Array Read                      | Output does not change. |  |                         |
| 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     |  | Output does not change.           |  | Array Read                      | Output does not change. |  |                         |

**Notes:**

- "Illegal commands" include commands outside of the allowed command set (allowed commands: 40H [pgm], 20H [erase], etc.)
- If a "Read Array" is attempted from a busy partition, the result will be invalid data. The ID and Query data are located at different locations in the address map.
- 1st and 2nd cycles of "2 cycles write commands" must be given to the same partition address, or unexpected results will occur.
- To protect memory contents against erroneous command sequences, there are specific instances in a multi-cycle command sequence in which the second cycle will be ignored. For example, when the device is program suspended and an erase setup command (0x20) is given followed by a confirm/resume command (0xD0), the second command will be ignored because it is unclear whether the user intends to erase the block or resume the program operation.
- The Clear Status command only clears the error bits in the status register if the device is not in the following modes: WSM running (Pgm Busy, Erase Busy, Pgm Busy In Erase Suspend, OTP Busy, BEFP modes).
- BEFP writes are only allowed when the status register bit #0 = 0, or else the data is ignored.

7. The "current state" is that of the "chip" and not of the "partition"; Each partition "remembers" which output (Array, ID/CFI or Status) it was last pointed to on the last instruction to the "chip", but the next state of the chip does not depend on where the partition's output mux is presently pointing to.
8. Confirm commands (Lock Block, Unlock Block, Lock-Down Block, Configuration Register) perform the operation and then move to the Ready State.
9. WAO refers to the block address latched during the first write cycle of the current operation.

## Appendix B Flowcharts

Figure 38: Word Program Flowchart

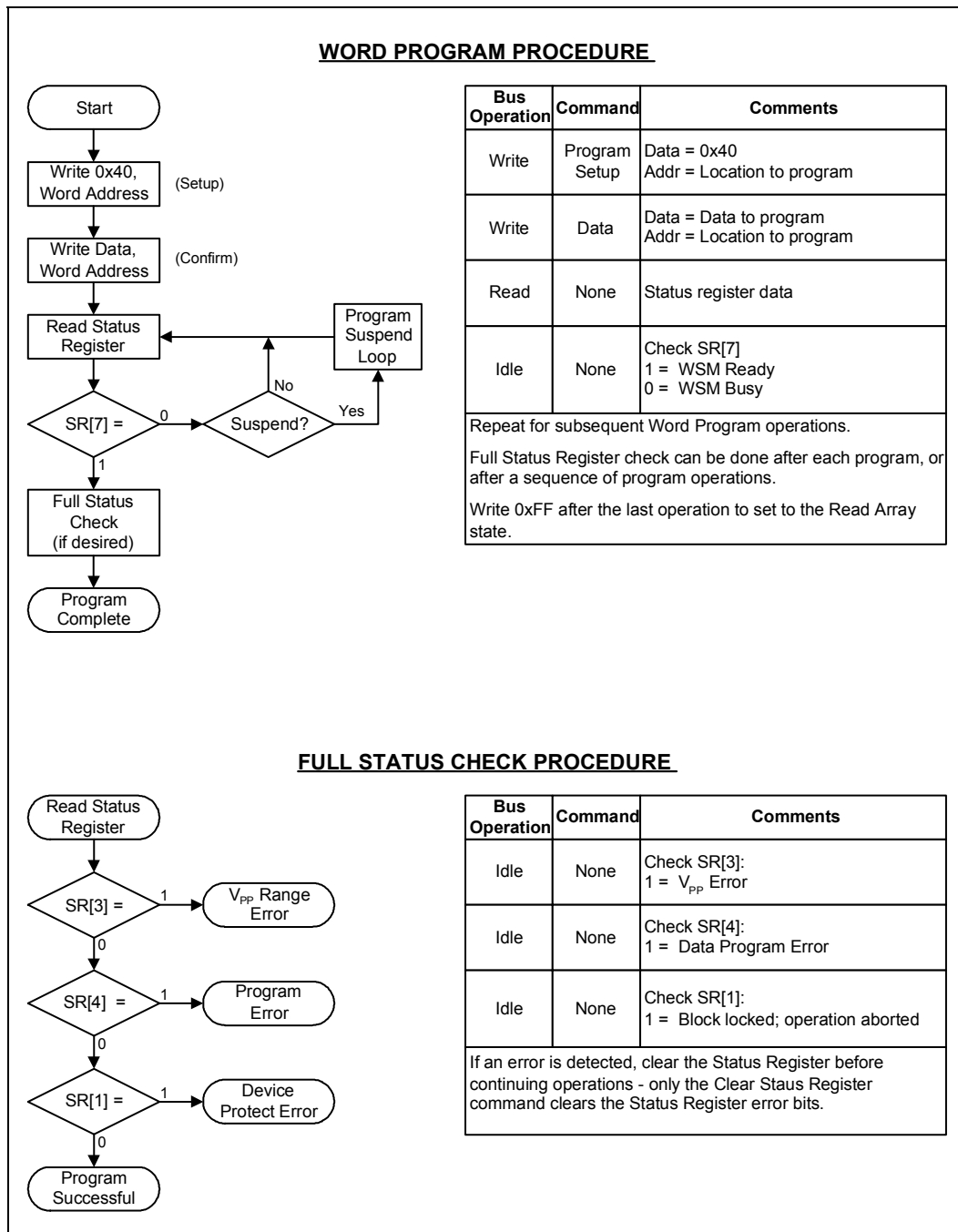


Figure 39: Program Suspend/Resume Flowchart

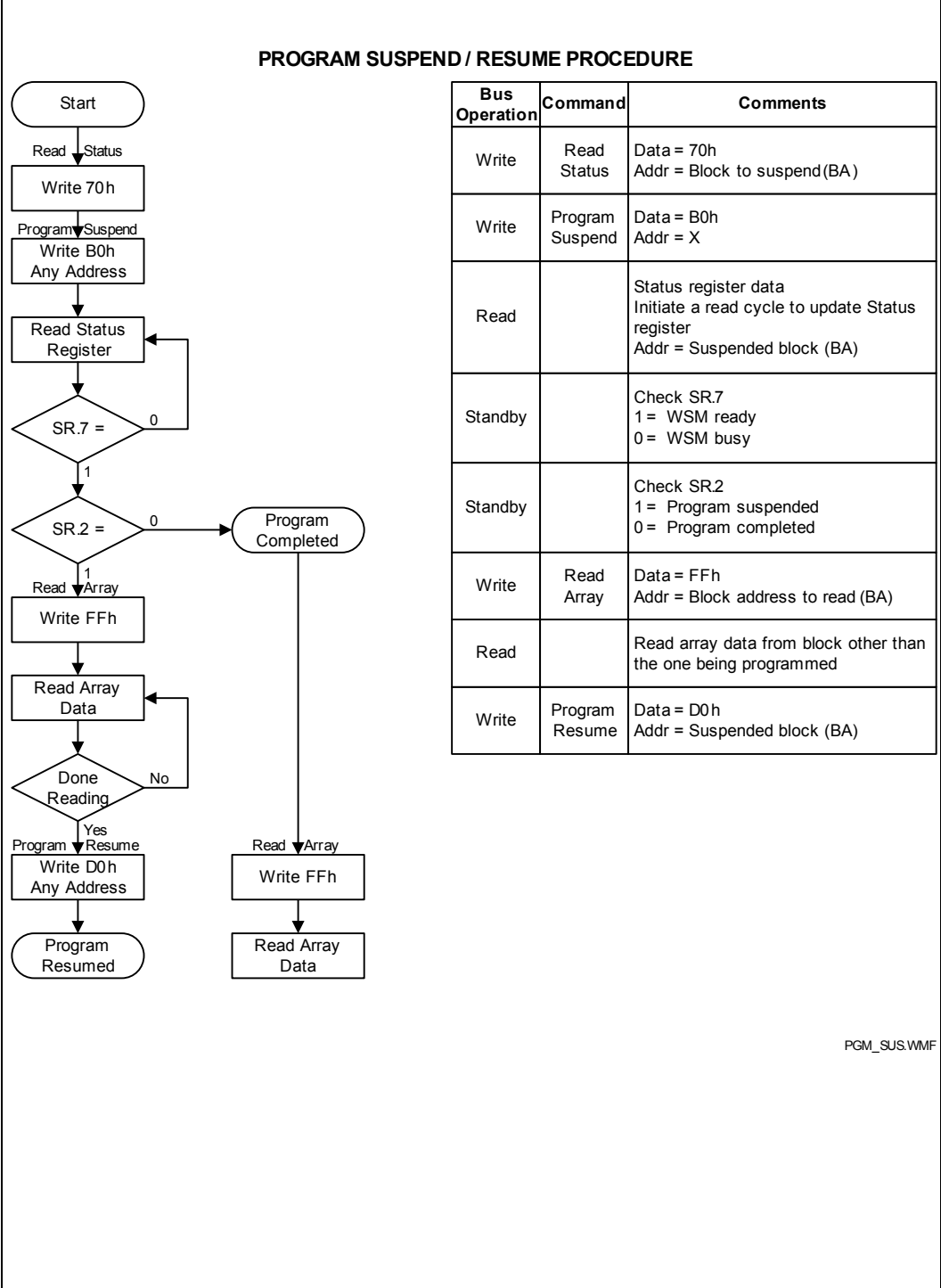
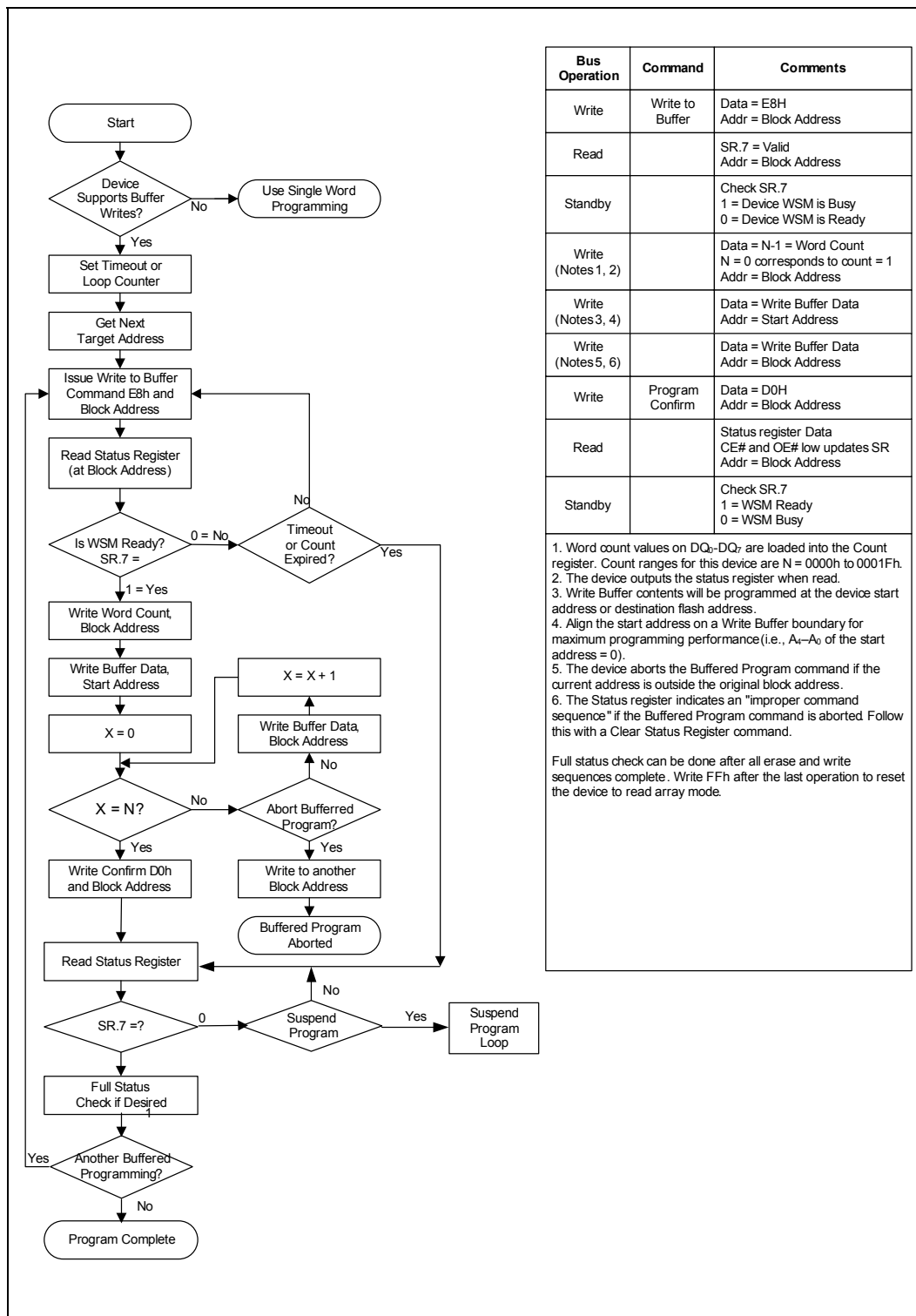




Figure 40: Buffer Program Flowchart



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

1. Word count values on DQ<sub>6</sub>-DQ<sub>7</sub> 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 programmed at the device start address or destination flash address.
4. Align the start address on a Write Buffer boundary for maximum programming performance (i.e., A<sub>4</sub>-A<sub>0</sub> 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 41: BEFP Flowchart

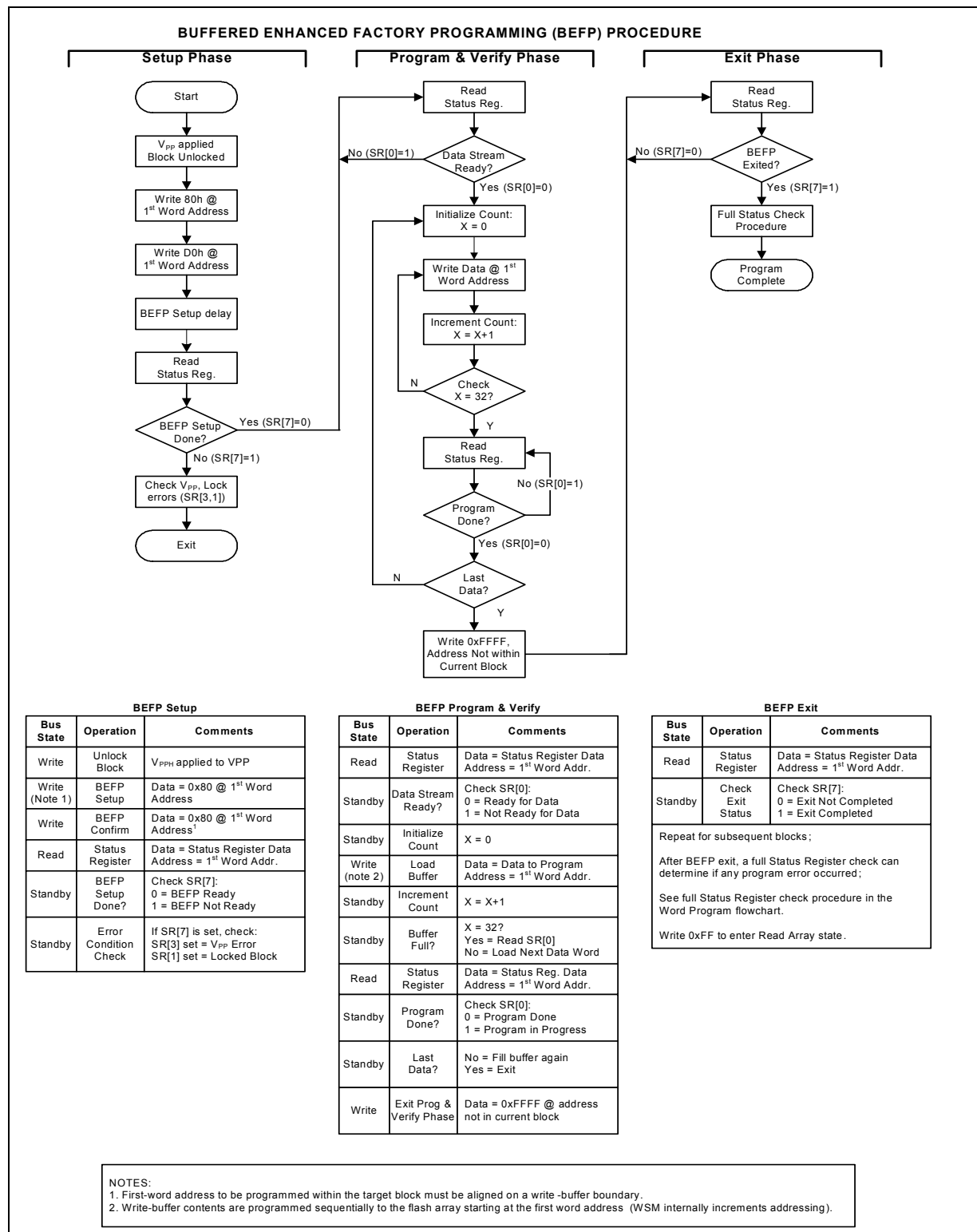


Figure 42: Block Erase Flowchart

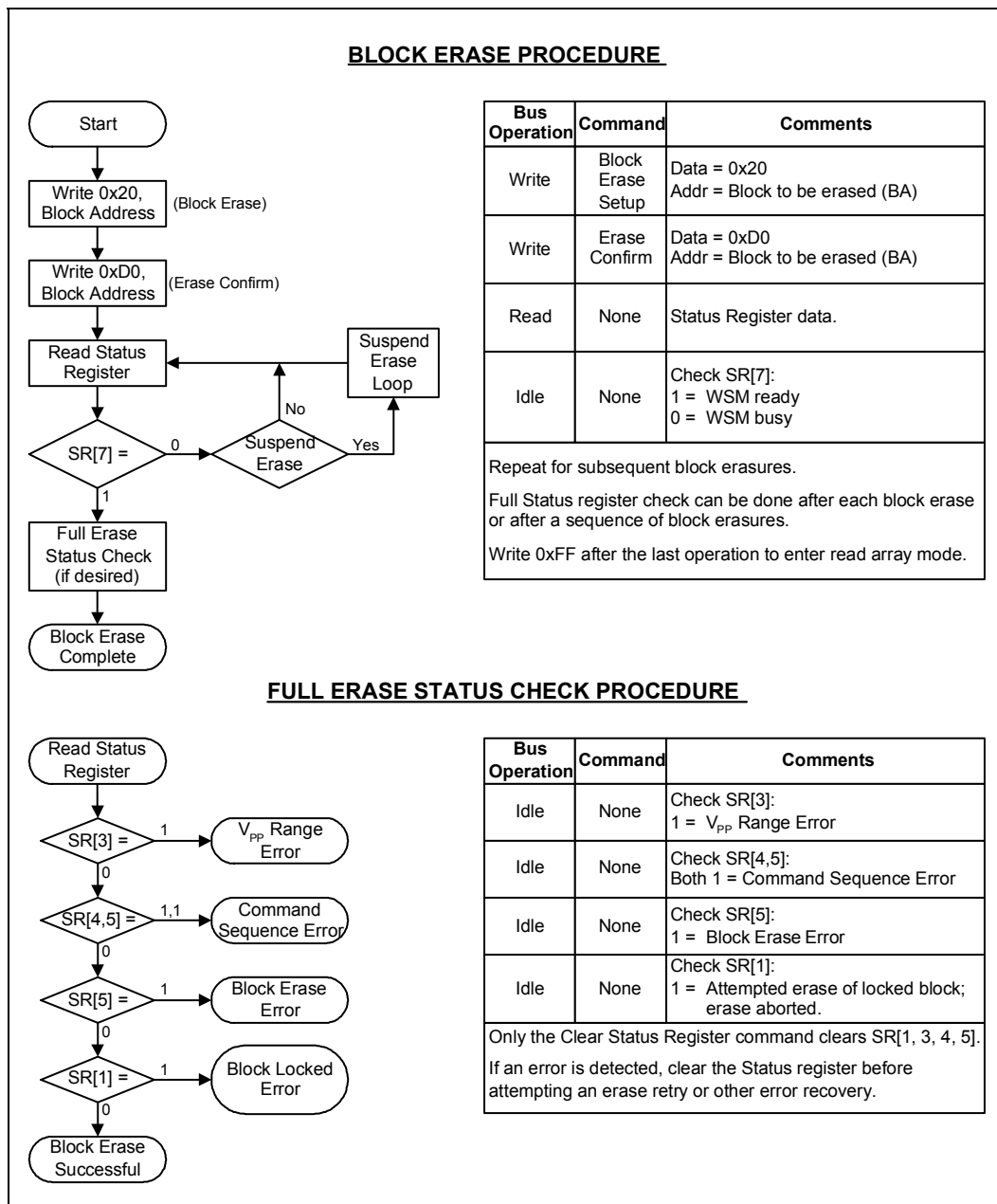
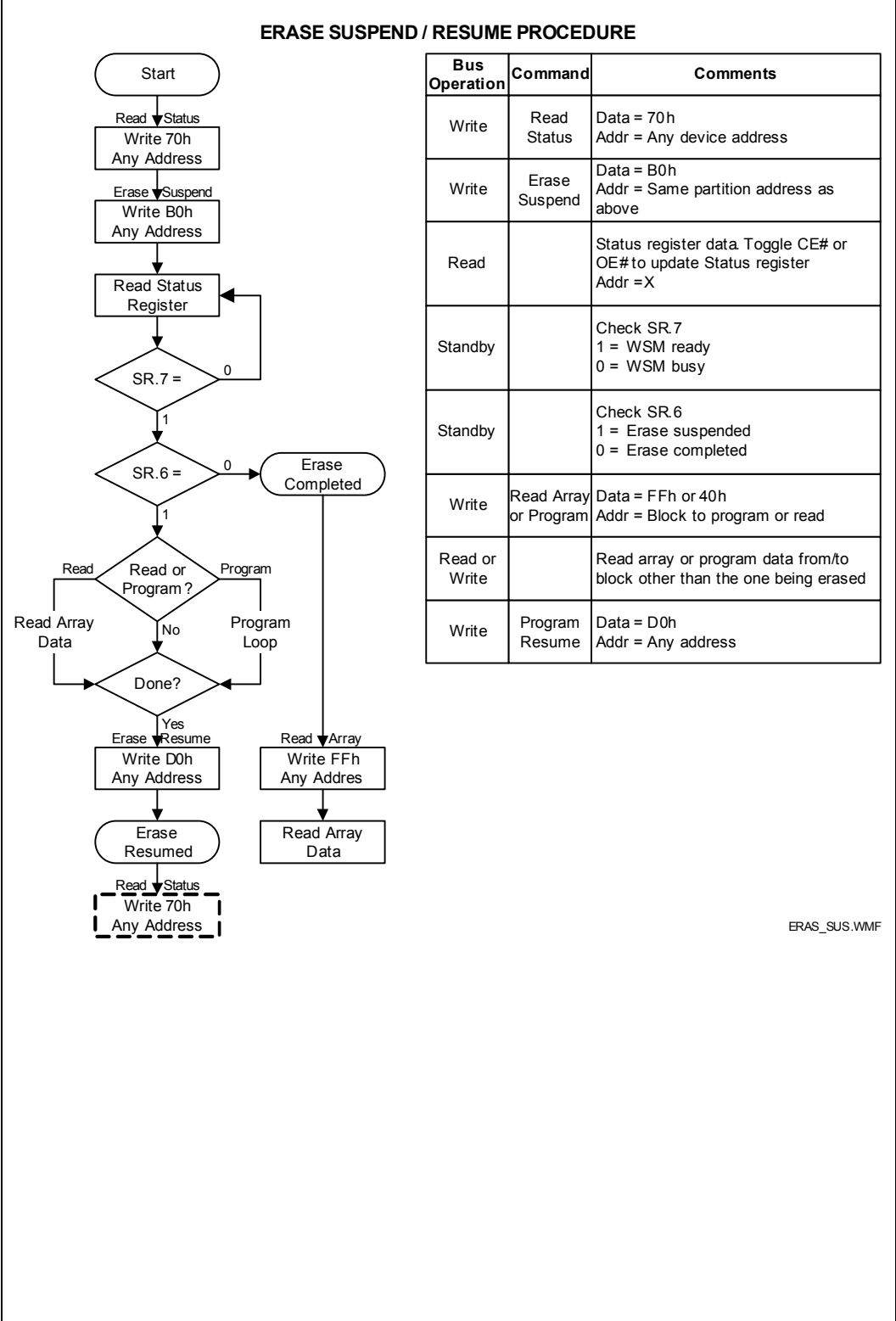
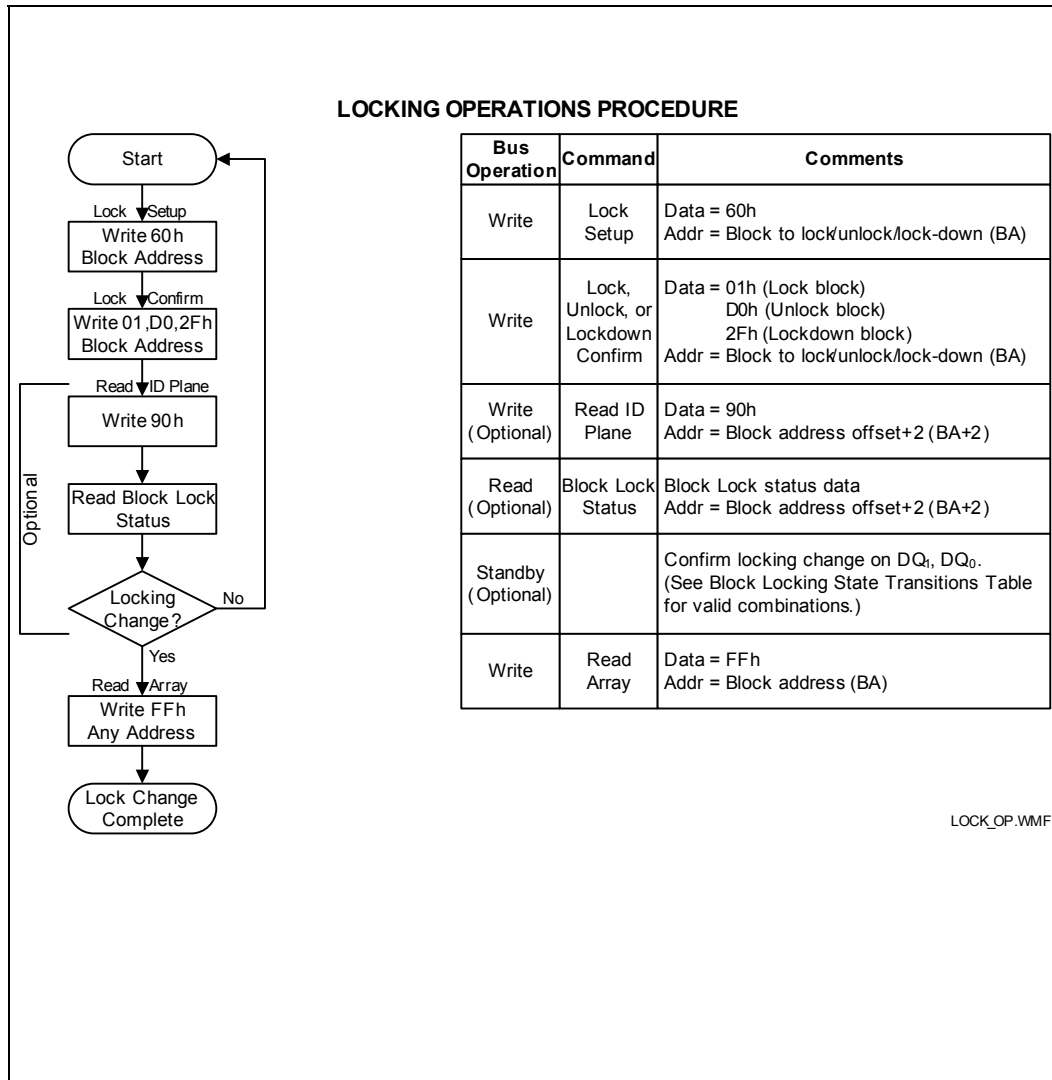


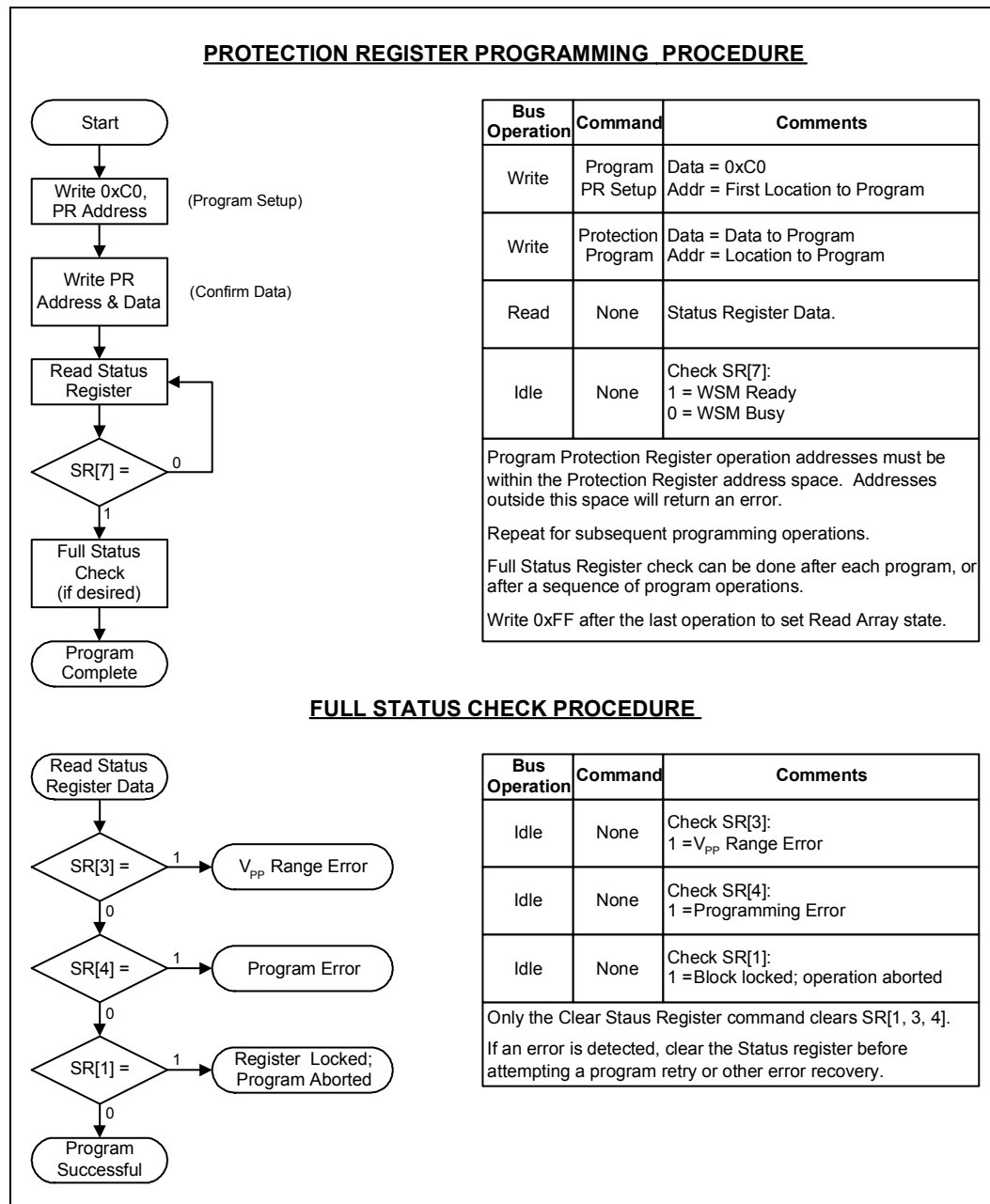
Figure 43: Erase Suspend/Resume Flowchart



**Figure 44: Block Lock Operations Flowchart**



**Figure 45: Protection Register Programming Flowchart**



## Appendix C Common Flash Interface

The Common Flash Interface (CFI) is part of an overall specification for multiple command-set and control-interface descriptions. This appendix describes the database structure containing the data returned by a read operation after issuing the CFI Query command (see [Section 9.2, "Device Commands" on page 45](#)). System software can parse this database structure to obtain information about the flash device, such as block size, density, bus width, and electrical specifications. The system software will then know which command set(s) to use to properly perform flash writes, block erases, reads and otherwise control the flash device.

### C.1 Query Structure Output

The Query database allows system software to obtain information for controlling the flash 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<sub>7-0</sub>) 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<sub>7-0</sub>) and 00h in the high byte (DQ<sub>15-8</sub>).

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 Mode**

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

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

| Word Addressing:               |                                 |           | Byte Addressing:               |                                |          |
|--------------------------------|---------------------------------|-----------|--------------------------------|--------------------------------|----------|
| Offset                         | Hex Code                        | Value     | Offset                         | Hex Code                       | Value    |
| A <sub>x</sub> -A <sub>0</sub> | D <sub>15</sub> -D <sub>0</sub> |           | A <sub>x</sub> -A <sub>0</sub> | D <sub>7</sub> -D <sub>0</sub> |          |
| 00010h                         | 0051                            | "Q"       | 00010h                         | 51                             | "Q"      |
| 00011h                         | 0052                            | "R"       | 00011h                         | 52                             | "R"      |
| 00012h                         | 0059                            | "Y"       | 00012h                         | 59                             | "Y"      |
| 00013h                         | P_ID <sub>LO</sub>              | PrVendor  | 00013h                         | P_ID <sub>LO</sub>             | PrVendor |
| 00014h                         | P_ID <sub>HI</sub>              | ID #      | 00014h                         | P_ID <sub>LO</sub>             | ID #     |
| 00015h                         | P <sub>LO</sub>                 | PrVendor  | 00015h                         | P_ID <sub>HI</sub>             | ID #     |
| 00016h                         | P <sub>HI</sub>                 | TblAdr    | 00016h                         | ...                            | ...      |
| 00017h                         | A_ID <sub>LO</sub>              | AltVendor | 00017h                         |                                |          |
| 00018h                         | A_ID <sub>HI</sub>              | ID #      | 00018h                         |                                |          |
| ...                            | ...                             | ...       | ...                            |                                |          |

## C.2 Query Structure Overview

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

**Table 37: Query Structure**

| Offset           | Sub-Section Name                            | Description <sup>(1)</sup>   |
|------------------|---|--|
| 00001-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 <sup>(3)</sup> | 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 32-KWord).
3. Offset 15 defines "P" which points to the Primary Numonyx-specific Extended Query Table.

## C.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 38: CFI Identification**

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



**Table 39: System Interface Information**

| Offset | Length | Description   | Add. | Hex Code | Value  |
|--------|--------|---|------|----------|--------|
| 1Bh    | 1      | V <sub>CC</sub> logic supply minimum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 BCD volts         | 1B:  | --17     | 1.7V   |
| 1Ch    | 1      | V <sub>CC</sub> logic supply maximum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 BCD volts         | 1C:  | --20     | 2.0V   |
| 1Dh    | 1      | V <sub>PP</sub> [programming] supply minimum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 HEX volts | 1D:  | --85     | 8.5V   |
| 1Eh    | 1      | V <sub>PP</sub> [programming] supply maximum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 HEX volts | 1E:  | --95     | 9.5V   |
| 1Fh    | 1      | "n" such that typical single word program time-out = 2 <sup>n</sup> μ-sec                                       | 1F:  | --08     | 256μs  |
| 20h    | 1      | "n" such that typical max. buffer write time-out = 2 <sup>n</sup> μ-sec   | 20:  | --09     | 512μs  |
| 21h    | 1      | "n" such that typical block erase time-out = 2 <sup>n</sup> m-sec   | 21:  | --0A     | 1s     |
| 22h    | 1      | "n" such that typical full chip erase time-out = 2 <sup>n</sup> m-sec   | 22:  | --00     | NA     |
| 23h    | 1      | "n" such that maximum word program time-out = 2 <sup>n</sup> times typical                                      | 23:  | --01     | 512μs  |
| 24h    | 1      | "n" such that maximum buffer write time-out = 2 <sup>n</sup> times typical                                      | 24:  | --01     | 1024μs |
| 25h    | 1      | "n" such that maximum block erase time-out = 2 <sup>n</sup> times typical                                       | 25:  | --02     | 4s     |
| 26h    | 1      | "n" such that maximum chip erase time-out = 2 <sup>n</sup> times typical  | 26:  | --00     | NA     |

## C.4 Device Geometry Definition

**Table 40: Device Geometry Definition**

| Offset | Length | Description   | Code                     |                 |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------|---|--------------------------|-----------------|-----|---|---|---|-----|-----|-----|----|---|---|---|---|---|-----|-----|-----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 27h    | 1      | "n" such that device size = 2 <sup>n</sup> in number of bytes   | 27:                      | See table below |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 28h    | 2      | Flash device interface code assignment:<br>"n" such that n+1 specifies the bit field that represents the flash device width capabilities as described in the table:   | 28:                      | --01            | x16 |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |        | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>—</td><td>—</td><td>—</td><td>—</td><td>x64</td><td>x32</td><td>x16</td><td>x8</td> </tr> <tr> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> </tr> <tr> <td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td> </tr> </table> |                          |                 |     | 7 | 6 | 5 | 4   | 3   | 2   | 1  | 0 | — | — | — | — | x64 | x32 | x16 | x8 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | — | — | — | — | — | — | — | — |
|        |        | 7   |                          |                 |     | 6 | 5 | 4 | 3   | 2   | 1   | 0  |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |        | —   |                          |                 |     | — | — | — | x64 | x32 | x16 | x8 |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 15     | 14     | 13  | 12                       | 11              | 10  | 9 | 8 |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| —      | —      | —   | —                        | —               | —   | — | — |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 29:    | --00   |   |                          |                 |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2Ah    | 2      | "n" such that maximum number of bytes in write buffer = 2 <sup>n</sup>  | 2A:<br>2B:               | --06<br>--00    | 64  |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2Ch    | 1      | Number of erase block regions (x) within device:<br>1. x = 0 means no erase blocking; the device erases in bulk<br>2. x specifies the number of device regions with one or more contiguous same-size erase blocks.<br>3. Symmetrically blocked partitions have one blocking region  | 2C:                      | See table below |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2Dh    | 4      | Erase Block Region 1 Information<br>bits 0–15 = y, y+1 = number of identical-size erase blocks<br>bits 16–31 = z, region erase block(s) size are z x 256 bytes  | 2D:<br>2E:<br>2F:<br>30: | See table below |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 31h    | 4      | Erase Block Region 2 Information<br>bits 0–15 = y, y+1 = number of identical-size erase blocks<br>bits 16–31 = z, region erase block(s) size are z x 256 bytes  | 31:<br>32:<br>33:<br>34: | See table below |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 35h    | 4      | Reserved for future erase block region information  | 35:<br>36:<br>37:<br>38: | See table below |     |   |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Address | 64-Mbit |      | 128-Mbit |      | 256-Mbit |      |
|---------|---------|------|----------|------|----------|------|
|         | --B     | --T  | --B      | --T  | --B      | --T  |
| 27:     | --17    | --17 | --18     | --18 | --19     | --19 |
| 28:     | --01    | --01 | --01     | --01 | --01     | --01 |
| 29:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 2A:     | --06    | --06 | --06     | --06 | --06     | --06 |
| 2B:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 2C:     | --02    | --02 | --02     | --02 | --02     | --02 |
| 2D:     | --03    | --3E | --03     | --7E | --03     | --FE |
| 2E:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 2F:     | --80    | --00 | --80     | --00 | --80     | --00 |
| 30:     | --00    | --02 | --00     | --02 | --00     | --02 |
| 31:     | --3E    | --03 | --7E     | --03 | --FE     | --03 |
| 32:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 33:     | --00    | --80 | --00     | --80 | --00     | --80 |
| 34:     | --02    | --00 | --02     | --00 | --02     | --00 |
| 35:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 36:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 37:     | --00    | --00 | --00     | --00 | --00     | --00 |
| 38:     | --00    | --00 | --00     | --00 | --00     | --00 |

## C.5 Numonyx-Specific Extended Query Table

**Table 41: Primary Vendor-Specific Extended Query**

| Offset <sup>(1)</sup><br>P = 10Ah    | Length | Description<br>(Optional flash features and commands)  | Add.                         | Hex Code   | Value   |
|--------------------------------------|--------|--|------------------------------|--|---|
| (P+0)h<br>(P+1)h<br>(P+2)h           | 3      | Primary extended query table<br>Unique ASCII string "PRI"  | 10A<br>10B:<br>10C:          | --50<br>--52<br>--49   | "P"<br>"R"<br>"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<br>(P+6)h<br>(P+7)h<br>(P+8)h | 4      | Optional feature and command support (1=yes, 0=no)<br><i>bits 11–29 are reserved; undefined bits are "0." If bit 31 is "1" then another 31 bit field of Optional features follows at the end of the bit–30 field.</i><br>bit 0 Chip erase supported<br>bit 1 Suspend erase supported<br>bit 2 Suspend program supported<br>bit 3 Legacy lock/unlock supported<br>bit 4 Queued erase supported<br>bit 5 Instant individual block locking supported<br>bit 6 Protection bits supported<br>bit 7 Pagemode read supported<br>bit 8 Synchronous read supported<br>bit 9 Simultaneous operations supported<br>bit 10 Extended Flash Array Blocks supported<br>bit 30 CFI Link(s) to follow<br>bit 31 Another "Optional Features" field to follow | 10F:<br>110:<br>111:<br>112: | --E6<br>--01<br>--00<br>See table below                          | No<br>Yes<br>Yes<br>No<br>No<br>Yes<br>Yes<br>Yes<br>Yes<br>No<br>No<br>No<br>See table below |
| (P+9)h                               | 1      | Supported functions after suspend: read Array, Status, Query<br>Other supported operations are:<br>bits 1–7 reserved; undefined bits are "0"<br>bit 0 Program supported after erase suspend  | 113:                         | --01<br>bit 0 = 1  | <br>Yes   |
| (P+A)h<br>(P+B)h                     | 2      | Block status register mask<br><i>bits 2–15 are Reserved; undefined bits are "0"</i><br>bit 0 Block Lock-Bit Status register active<br>bit 1 Block Lock-Down Bit Status active<br>bit 4 EFA Block Lock-Bit Status register active<br>bit 5 EFA Block Lock-Down Bit Status active  | 114:<br>115:                 | --03<br>--00<br>bit 0 = 1<br>bit 1 = 1<br>bit 4 = 0<br>bit 5 = 0 | <br>Yes<br>Yes<br>No<br>No  |
| (P+C)h                               | 1      | V <sub>CC</sub> logic supply highest performance program/erase voltage<br>bits 0–3 BCD value in 100 mV<br>bits 4–7 BCD value in volts  | 116:                         | --18   | 1.8V  |
| (P+D)h                               | 1      | V <sub>PP</sub> optimum program/erase supply voltage<br>bits 0–3 BCD value in 100 mV<br>bits 4–7 HEX value in volts  | 117:                         | --90   | 9.0V  |

| Address | Discrete |      | 512-Mbit  |           |           |           |
|---------|----------|------|-----------|-----------|-----------|-----------|
|         | –B       | –T   | –B        |           | –T        |           |
|         | —        | —    | die 1 (B) | die 2 (T) | die 1 (T) | die 2 (B) |
| 112:    | --00     | --00 | --40      | --00      | --40      | --00      |

**Table 42: Protection Register Information**

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

**Table 43: Burst Read Information**

| Offset <sup>(1)</sup><br>P = 10Ah | Length | Description<br>(Optional flash features and commands)   | Add. | Hex Code | Value  |
|-----------------------------------|--------|---|------|----------|--------|
| (P+1D)h                           | 1      | Page Mode Read capability<br>bits 0–7 = "n" such that 2 <sup>n</sup> 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: | --03     | 8 byte |
| (P+1E)h                           | 1      | Number of synchronous mode read configuration fields that follow. 00h indicates no burst capability.  | 128: | --04     | 4      |
| (P+1F)h                           | 1      | Synchronous mode read capability configuration 1<br>Bits 3–7 = Reserved<br>bits 0–2 "n" such that 2 <sup>n+1</sup> HEX value represents the maximum number of continuous synchronous reads when the device is configured for its maximum word width. A value of 07h indicates that the device is capable of continuous linear bursts that will output data until the internal burst counter reaches the end of the device's burstable address space. This field's 3-bit value can be written directly to the Read Configuration Register bits 0–2 if the device is configured for its maximum word width. See offset 28h for word width to determine the burst data output width. | 129: | --01     | 4      |
| (P+20)h                           | 1      | Synchronous mode read capability configuration 2  | 12A: | --02     | 8      |
| (P+21)h                           | 1      | Synchronous mode read capability configuration 3  | 12B: | --03     | 16     |
| (P+22)h                           | 1      | Synchronous mode read capability configuration 4  | 12C: | --07     | Cont   |

**Table 44: Partition and Erase Block Region Information**

| Offset <sup>(1)</sup><br>P = 10Ah |         | Description<br><br>(Optional flash features and commands)  | See table below |         |      |
|-----------------------------------|---------|--|-----------------|---------|------|
| Bottom                            | Top     |  | Len             | Address |      |
|                                   |         |  |                 | Bot     | Top  |
| (P+23)h                           | (P+23)h | Number of device hardware-partition regions within the device.<br>x = 0: a single hardware partition device (no fields follow).<br>x specifies the number of device partition regions containing one or more contiguous erase block regions. | 1               | 12D:    | 12D: |

**Table 45: Partition Region 1 Information**

| Offset <sup>(1)</sup><br>P = 10Ah |         | Description<br><br>(Optional flash features and commands)   | See table below |         |      |
|-----------------------------------|---------|---|-----------------|---------|------|
| Bottom                            | Top     |   | Len             | Address |      |
|                                   |         |   |                 | Bot     | Top  |
| (P+24)h                           | (P+24)h | Data size of this Partition Region Information field  | 2               | 12E:    | 12E  |
| (P+25)h                           | (P+25)h | (# addressable locations, including this field)   |                 | 12F     | 12F  |
| (P+26)h                           | (P+26)h | Number of identical partitions within the partition region  | 2               | 130:    | 130: |
| (P+27)h                           | (P+27)h |   |                 | 131:    | 131: |
| (P+28)h                           | (P+28)h | Number of program or erase operations allowed in a partition<br>bits 0–3 = number of simultaneous Program operations<br>bits 4–7 = number of simultaneous Erase operations  | 1               | 132:    | 132: |
| (P+29)h                           | (P+29)h | Simultaneous program or erase operations allowed in other partitions while a partition in this region is in Program mode<br>bits 0–3 = number of simultaneous Program operations<br>bits 4–7 = number of simultaneous Erase operations  | 1               | 133:    | 133: |
| (P+2A)h                           | (P+2A)h | Simultaneous program or erase operations allowed in other partitions while a partition in this region is in Erase mode<br>bits 0–3 = number of simultaneous Program operations<br>bits 4–7 = number of simultaneous Erase operations  | 1               | 134:    | 134: |
| (P+2B)h                           | (P+2B)h | Types of erase block regions in this Partition Region.<br>x = 0 = no erase blocking; the Partition Region erases in bulk<br>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               | 135:    | 135: |

Table 46: Partition Region 1 Information (continued)

| Offset <sup>(1)</sup><br>P = 10Ah                              |  | Description<br><br>(Optional flash features and commands)   | See table below |  |  |
|--|--|---|-----------------|--|--|
| Bottom   | Top  |   | Len             | Address                                      |  |
|  |  |   |                 | Bot  | Top  |
| (P+2C)h<br>(P+2D)h<br>(P+2E)h<br>(P+2F)h                       | (P+2C)h<br>(P+2D)h<br>(P+2E)h<br>(P+2F)h                       | Partition Region 1 Erase Block Type 1 Information<br>bits 0–15 = y, y+1 = # identical-size erase blks in a partition<br>bits 16–31 = z, region erase block(s) size are z x 256 bytes  | 4               | 136:<br>137:<br>138:<br>139:                 | 136:<br>137:<br>138:<br>139:                 |
| (P+30)h<br>(P+31)h   | (P+30)h<br>(P+31)h   | Partition 1 (Erase Block Type 1)<br>Block erase cycles x 1000   | 2               | 13A:<br>13B:                                 | 13A:<br>13B:                                 |
| (P+32)h  | (P+32)h  | Partition 1 (erase block Type 1) bits per cell; internal EDAC<br>bits 0–3 = bits per cell in erase region<br>bit 4 = internal EDAC used (1=yes, 0=no)<br>bits 5–7 = reserve for future use  | 1               | 13C:   | 13C:   |
| (P+33)h  | (P+33)h  | Partition 1 (erase block Type 1) page mode and synchronous mode capabilities defined in Table 10.<br>bit 0 = page-mode host reads permitted (1=yes, 0=no)<br>bit 1 = synchronous host reads permitted (1=yes, 0=no)<br>bit 2 = synchronous host writes permitted (1=yes, 0=no)<br>bits 3–7 = reserved for future use  | 1               | 13D:   | 13D:   |
| (P+34)h<br>(P+35)h<br>(P+36)h<br>(P+37)h<br>(P+38)h<br>(P+39)h | (P+34)h<br>(P+35)h<br>(P+36)h<br>(P+37)h<br>(P+38)h<br>(P+39)h | Partition Region 1 (Erase Block Type 1) Programming Region Information<br>bits 0–7 = x, 2 <sup>x</sup> = Programming Region aligned size ( <b>bytes</b> )<br>bits 8–14 = Reserved; bit 15 = Legacy flash operation (ignore 0:7)<br>bits 16–23 = y = Control Mode <b>valid</b> size in bytes<br>bits 24–31 = Reserved<br>bits 32–39 = z = Control Mode <b>invalid</b> size in bytes<br>bits 40–46 = Reserved; bit 47 = Legacy flash operation (ignore 23:16 & 39:32) | 6               | 13E:<br>13F:<br>140:<br>141:<br>142:<br>143: | 13E:<br>13F:<br>140:<br>141:<br>142:<br>143: |
| (P+3A)h<br>(P+3B)h<br>(P+3C)h<br>(P+3D)h                       | (P+3A)h<br>(P+3B)h<br>(P+3C)h<br>(P+3D)h                       | Partition Region 1 Erase Block Type 2 Information<br>bits 0–15 = y, y+1 = # identical-size erase blks in a partition<br>bits 16–31 = z, region erase block(s) size are z x 256 bytes  | 4               | 144:<br>145:<br>146:<br>147:                 | 144:<br>145:<br>146:<br>147:                 |
| (P+3E)h<br>(P+3F)h   | (P+3E)h<br>(P+3F)h   | Partition 1 (Erase Block Type 2)<br>Block erase cycles x 1000   | 2               | 148:<br>149:                                 | 148:<br>149:                                 |
| (P+40)h  | (P+40)h  | Partition 1 (erase block Type 2) bits per cell; internal EDAC<br>bits 0–3 = bits per cell in erase region<br>bit 4 = internal EDAC used (1=yes, 0=no)<br>bits 5–7 = reserve for future use  | 1               | 14A:   | 14A:   |
| (P+41)h  | (P+41)h  | Partition 1 (erase block Type 2) page mode and synchronous mode capabilities defined in Table 10.<br>bit 0 = page-mode host reads permitted (1=yes, 0=no)<br>bit 1 = synchronous host reads permitted (1=yes, 0=no)<br>bit 2 = synchronous host writes permitted (1=yes, 0=no)<br>bits 3–7 = reserved for future use  | 1               | 14B:   | 14B:   |
| (P+42)h<br>(P+43)h<br>(P+44)h<br>(P+45)h<br>(P+46)h<br>(P+47)h | (P+42)h<br>(P+43)h<br>(P+44)h<br>(P+45)h<br>(P+46)h<br>(P+47)h | Partition Region 1 (Erase Block Type 2) Programming Region Information<br>bits 0–7 = x, 2 <sup>x</sup> = Programming Region aligned size ( <b>bytes</b> )<br>bits 8–14 = Reserved; bit 15 = Legacy flash operation (ignore 0:7)<br>bits 16–23 = y = Control Mode <b>valid</b> size in bytes<br>bits 24–31 = Reserved<br>bits 32–39 = z = Control Mode <b>invalid</b> size in bytes<br>bits 40–46 = Reserved; bit 47 = Legacy flash operation (ignore 23:16 & 39:32) | 6               | 14C:<br>14D:<br>14E:<br>14F:<br>150:<br>151: | 14C:<br>14D:<br>14E:<br>14F:<br>150:<br>151: |

**Table 47: Partition and Erase Block Region Information**

| Address | 64-Mbit |      | 128-Mbit |      | 256-Mbit |      |
|---------|---------|------|----------|------|----------|------|
|         | -B      | -T   | -B       | -T   | -B       | -T   |
| 12D:    | --01    | --01 | --01     | --01 | --01     | --01 |
| 12E:    | --24    | --24 | --24     | --24 | --24     | --24 |
| 12F:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 130:    | --01    | --01 | --01     | --01 | --01     | --01 |
| 131:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 132:    | --11    | --11 | --11     | --11 | --11     | --11 |
| 133:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 134:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 135:    | --02    | --02 | --02     | --02 | --02     | --02 |
| 136:    | --03    | --3E | --03     | --7E | --03     | --FE |
| 137:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 138:    | --80    | --00 | --80     | --00 | --80     | --00 |
| 139:    | --00    | --02 | --00     | --02 | --00     | --02 |
| 13A:    | --64    | --64 | --64     | --64 | --64     | --64 |
| 13B:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 13C:    | --02    | --02 | --02     | --02 | --02     | --02 |
| 13D:    | --03    | --03 | --03     | --03 | --03     | --03 |
| 13E:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 13F:    | --80    | --80 | --80     | --80 | --80     | --80 |
| 140:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 141:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 142:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 143:    | --80    | --80 | --80     | --80 | --80     | --80 |
| 144:    | --3E    | --03 | --7E     | --03 | --FE     | --03 |
| 145:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 146:    | --00    | --80 | --00     | --80 | --00     | --80 |
| 147:    | --02    | --00 | --02     | --00 | --02     | --00 |
| 148:    | --64    | --64 | --64     | --64 | --64     | --64 |
| 149:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 14A:    | --02    | --02 | --02     | --02 | --02     | --02 |
| 14B:    | --03    | --03 | --03     | --03 | --03     | --03 |
| 14C:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 14D:    | --80    | --80 | --80     | --80 | --80     | --80 |
| 14E:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 14F:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 150:    | --00    | --00 | --00     | --00 | --00     | --00 |
| 151:    | --80    | --80 | --80     | --80 | --80     | --80 |

**Table 48: CFI Link Information**

| Offset <sup>(1)</sup>                    | Length | Description<br>(Optional flash features and commands)   | Add.                         | Hex Code | Value           |
|--|--------|---|------------------------------|----------|-----------------|
| P = 10Ah                                 |        |   |                              |          |                 |
| (P+48)h<br>(P+49)h<br>(P+4A)h<br>(P+4B)h | 4      | CFI Link Field bit definitions<br>Bits 0–9 = Address offset (within 32Mbit segment) of referenced CFI table<br>Bits 10–27 = nth 32Mbit segment of referenced CFI table<br>Bits 28–30 = Memory Type<br>Bit 31 = Another CFI Link field immediately follows | 152:<br>153:<br>154:<br>155: |          | See table below |
| (P+4C)h                                  | 1      | CFI Link Field Quantity Subfield definitions<br>Bits 0–3 = Quantity field (n such that n+1 equals quantity)<br>Bit 4 = Table & Die relative location<br>Bit 5 = Link Field & Table relative location<br>Bits 6–7 = Reserved                               | 156:                         |          | See table below |

| Address | Discrete |      | 512-Mbit  |           |           |           |
|---------|----------|------|-----------|-----------|-----------|-----------|
|         | -B       | -T   | -B        |           | -T        |           |
|         | —        | —    | die 1 (B) | die 2 (T) | die 1 (T) | die 2 (B) |
| 152:    | --FF     | --FF | --10      | --FF      | --10      | --FF      |
| 153:    | --FF     | --FF | --20      | --FF      | --20      | --FF      |
| 154:    | --FF     | --FF | --00      | --FF      | --00      | --FF      |
| 155:    | --FF     | --FF | --00      | --FF      | --00      | --FF      |
| 156:    | --FF     | --FF | --10      | --FF      | --10      | --FF      |



## Appendix D Additional Information

| Order/Document Number | Document/Tool   |
|-----------------------|---|
| 309045                | P30 Family Specification Update   |
| 308291                | Schematic Review Checklist for Numonyx™ StrataFlash® Embedded Memory (P30)  |
| 300783                | Using Numonyx™ Flash Memory: Asynchronous Page Mode and Synchronous Burst Mode  |
| 290667                | Numonyx™ StrataFlash® Memory (J3) Datasheet   |
| 306667                | Migration Guide for Numonyx™ StrataFlash® Memory (J3) to Numonyx™ StrataFlash® Embedded Memory (P30/P33) Application Note 812             |
| 314750                | Numonyx™ StrataFlash® Memory (P30) to Numonyx™ StrataFlash® Embedded Memory (P33) Conversion Guide Application Note 867                   |
| 290737                | Numonyx™ StrataFlash® Synchronous Memory (K3/K18) Datasheet   |
| 306669                | Migration Guide for Numonyx™ StrataFlash® Synchronous Memory (K3/K18) to Numonyx™ StrataFlash® Embedded Memory (P30) Application Note 825 |
| 290701                | Numonyx™ Wireless Flash Memory (W18) Datasheet  |
| 290702                | Numonyx™ Wireless Flash Memory (W30) Datasheet  |
| 252802                | Numonyx™ Flash Memory Design for a Stacked Chip Scale Package (SCSP)  |
| 298161                | Numonyx™ Flash Memory Chip Scale Package User's Guide   |
| 253418                | Numonyx™ Wireless Communications and Computing Package User's Guide   |
| 296514                | Numonyx™ Small Outline Package Guide  |
| 297833                | Numonyx™ Flash Data Integrator (Numonyx™ FDI) User Guide  |
| 298136                | Numonyx™ Persistent Storage Manager (Numonyx™ PSM) User Guide   |
| 306668                | Migration Guide for Spansion* S29GLxxxN to Numonyx™ StrataFlash® Embedded Memory (P30/P33) Application Note 813                           |

**Note:** Contact your local Numonyx or distribution sales office or visit Numonyx's World Wide Web home page at <http://www.numonyx.com> for technical documentation, tools, or the most current information on Numonyx™ Flash Memory.

## Appendix E Ordering Information for Discrete Products

Figure 46: Decoder for Discrete P30

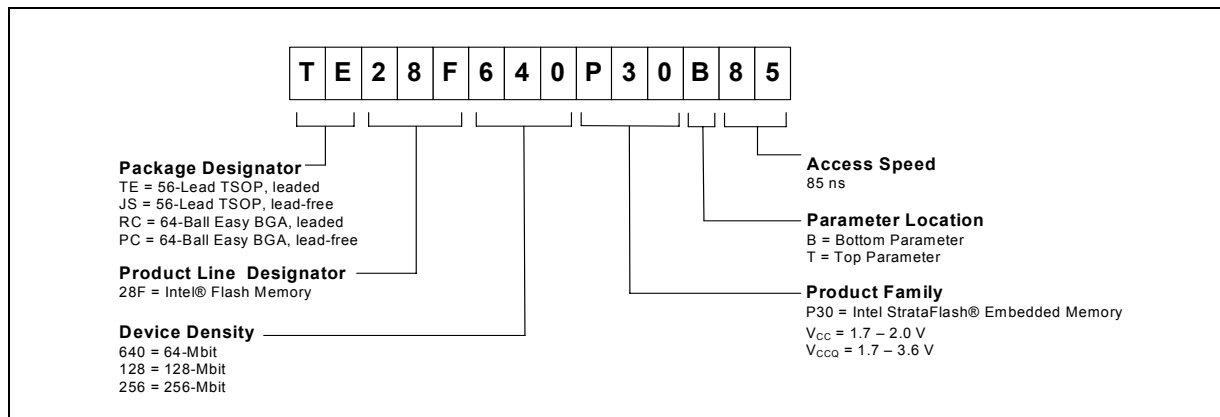
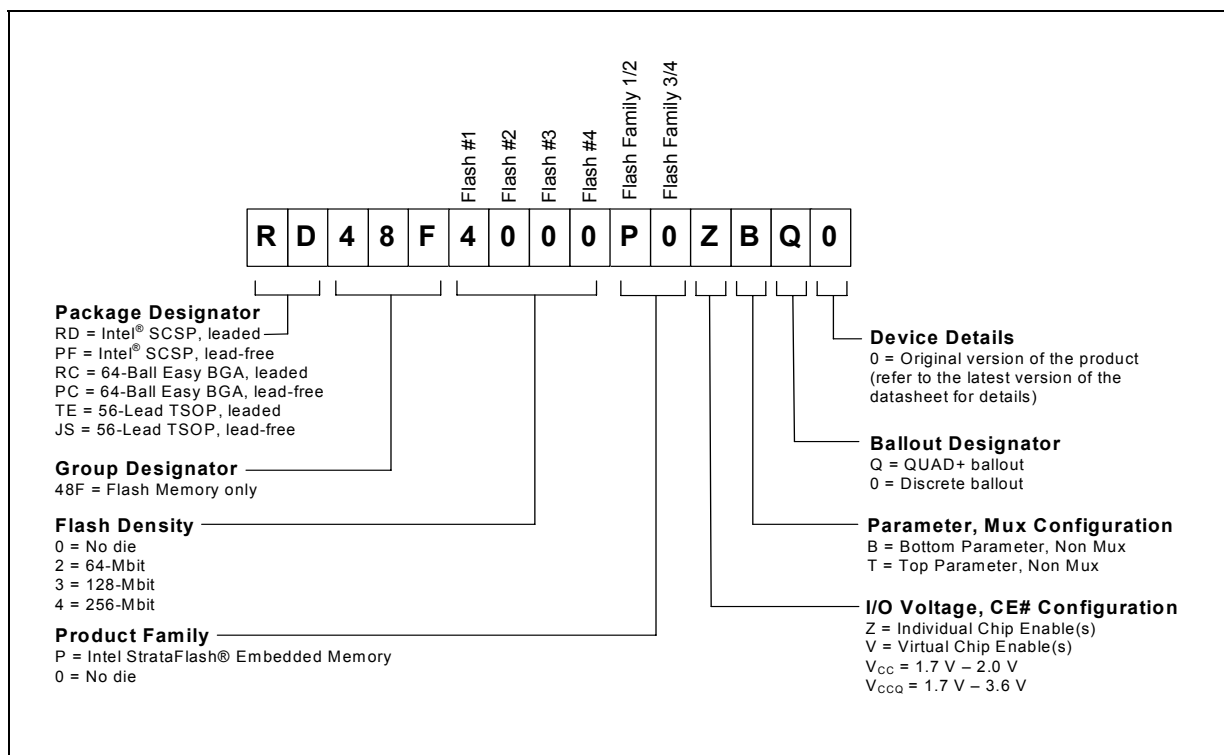


Table 49: Valid Combinations for Discrete Products

| 64-Mbit        | 128-Mbit       | 256-Mbit       |
|----------------|----------------|----------------|
| TE28F640P30B85 | TE28F128P30B85 | TE28F256P30B95 |
| TE28F640P30T85 | TE28F128P30T85 | TE28F256P30T95 |
| JS28F640P30B85 | JS28F128P30B85 | JS28F256P30B95 |
| JS28F640P30T85 | JS28F128P30T85 | JS28F256P30T95 |
| RC28F640P30B85 | RC28F128P30B85 | RC28F256P30B85 |
| RC28F640P30T85 | RC28F128P30T85 | RC28F256P30T85 |
| PC28F640P30B85 | PC28F128P30B85 | PC28F256P30B85 |
| PC28F640P30T85 | PC28F128P30T85 | PC28F256P30T85 |

## Appendix F Ordering Information for SCSP Products

Figure 47: Decoder for SCSP P30



Note: For 512-Mbit only, "B" is used for both top and bottom Parameter/Mux configurations.

Table 50: Valid Combinations for Dual- Die Products

| 64-Mbit         | 128-Mbit        | 256-Mbit        | 512-Mbit*       |
|-----------------|-----------------|-----------------|-----------------|
| RD48F2000P0ZBQ0 | RD48F3000P0ZBQ0 | RD48F4000P0ZBQ0 | RD48F4400P0VBQ0 |
| RD48F2000P0ZTQ0 | RD48F3000P0ZTQ0 | RD48F4000P0ZTQ0 | PF48F4400P0VBQ0 |
| PF48F2000P0ZBQ0 | PF48F3000P0ZBQ0 | PF48F4000P0ZBQ0 | RC48F4400P0VB00 |
| PF48F2000P0ZTQ0 | PF48F3000P0ZTQ0 | PF48F4000P0ZTQ0 | PC48F4400P0VB00 |
|                 |                 |                 | TE48F4400P0VB00 |
|                 |                 |                 | JS48F4400P0VB00 |

Note: \* The "B" parameter is used for both "top" and "bottom" options in the 512-Mbit density.