



**THE DATASHEET OF  
JS28F640P30BF75A**





# Numonyx<sup>®</sup> Axcell<sup>™</sup> P30-65nm Flash Memory

**128-Mbit, 64-Mbit Single Bit per Cell (SBC)**

---

**Datasheet**

## Product Features

- **High Performance:**
  - 65ns initial access time for Easy BGA and QUAD+
  - 75ns initial access time for TSOP
  - 25ns 8-word asynchronous-page read mode
  - 52MHz with zero WAIT states, 17ns clock-to-data output synchronous-burst read mode
  - 4-, 8-, 16- and continuous-word options for burst mode
  - 1.8V Low Power buffered programming at 1.8MByte/s (Typ) using 256-word buffer
  - Buffered Enhanced Factory Programming at 3.2MByte/s (typ) using 256-word buffer
- **Architecture:**
  - Asymmetrically-blocked architecture
  - Four 32-KByte parameter blocks: top or bottom configuration
  - 128-KByte array blocks
  - Blank Check to verify an erased block
- **Voltage and Power:**
  - VCC (core) voltage: 1.7V – 2.0V
  - VCCQ (I/O) voltage: 1.7V – 3.6V
  - Standby current: 30µA(Typ)/55µA(Max)
  - Continuous synchronous read current: 23mA (Typ)/28mA (Max) at 52MHz
- **Enhanced Security:**
  - Absolute write protection: VPP = Vss
  - Power-transition erase/program lockout
  - Individual zero-latency block locking
  - Individual block lock-down capability
  - Password Access feature
  - One-Time Programmable Register:
    - 64 OTP bits, programmed with unique information by Numonyx
    - 2112 OTP bits, available for customer programming
- **Software:**
  - 20µs (Typ) program suspend
  - 20µs (Typ) erase suspend
  - Basic Command Set and Extended Function Interface (EFI) Command Set compatible
  - Common Flash Interface capable
- **Density and Packaging:**
  - 56-Lead TSOP (128-Mbit, 64-Mbit)
  - 64-Ball Easy BGA (128-Mbit, 64-Mbit)
  - 88-Ball QUAD+ Package (128-Mbit)
  - 16-bit wide data bus
- **Quality and Reliability:**
  - JESD47E Compliant
  - Operating temperature: –40°C to +85°C
  - Minimum 100,000 erase cycles
  - 65nm process technology

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 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 Axcell 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 © 2010, Numonyx, B.V., All Rights Reserved.

# Contents

---

|             |  |    |
|-------------|--|----|
| <b>1.0</b>  | <b>Functional Description</b> .....        | 5  |
| 1.1         | Introduction .....                         | 5  |
| 1.2         | Overview .....                             | 5  |
| 1.3         | Memory Map.....                            | 6  |
| <b>2.0</b>  | <b>Package Information</b> .....           | 7  |
| 2.1         | 56-Lead TSOP.....                          | 7  |
| 2.2         | 64-Ball Easy BGA Package.....              | 8  |
| 2.3         | QUAD+ SCSP Packages .....                  | 10 |
| <b>3.0</b>  | <b>Pinouts/Ballouts</b> .....              | 11 |
| <b>4.0</b>  | <b>Signals</b> .....                       | 14 |
| <b>5.0</b>  | <b>Bus Operations</b> .....                | 16 |
| 5.1         | Read - Asynchronous Mode.....              | 16 |
| 5.2         | Read - Synchronous Mode .....              | 16 |
| 5.3         | Write .....                                | 17 |
| 5.4         | Output Disable.....                        | 17 |
| 5.5         | Standby .....                              | 17 |
| 5.6         | Reset.....                                 | 18 |
| <b>6.0</b>  | <b>Command Set</b> .....                   | 19 |
| 6.1         | Device Command Codes.....                  | 19 |
| 6.2         | Device Command Bus Cycles .....            | 20 |
| <b>7.0</b>  | <b>Read Operation</b> .....                | 22 |
| 7.1         | Read Array .....                           | 22 |
| 7.2         | Read Device Identifier .....               | 22 |
| 7.3         | Read CFI.....                              | 23 |
| 7.4         | Read Status Register .....                 | 23 |
| 7.5         | Clear Status Register .....                | 23 |
| <b>8.0</b>  | <b>Program Operation</b> .....             | 24 |
| 8.1         | Word Programming .....                     | 24 |
| 8.2         | Buffered Programming .....                 | 24 |
| 8.3         | Buffered Enhanced Factory Programming..... | 25 |
| 8.4         | Program Suspend.....                       | 27 |
| 8.5         | Program Resume.....                        | 28 |
| 8.6         | Program Protection.....                    | 28 |
| <b>9.0</b>  | <b>Erase Operation</b> .....               | 29 |
| 9.1         | Block Erase .....                          | 29 |
| 9.2         | Blank Check .....                          | 29 |
| 9.3         | Erase Suspend .....                        | 30 |
| 9.4         | Erase Resume.....                          | 30 |
| 9.5         | Erase Protection .....                     | 30 |
| <b>10.0</b> | <b>Security</b> .....                      | 31 |
| 10.1        | Block Locking.....                         | 31 |
| 10.2        | Selectable OTP Blocks .....                | 33 |
| 10.3        | Password Access .....                      | 33 |
| <b>11.0</b> | <b>Register</b> .....                      | 34 |
| 11.1        | Status Register (SR) .....                 | 34 |
| 11.2        | Read Configuration Register (RCR) .....    | 34 |

|             |   |           |
|-------------|---|-----------|
| 11.3        | One-Time Programmable (OTP) Registers .....           | 40        |
| <b>12.0</b> | <b>Power and Reset Specifications .....</b>           | <b>43</b> |
| 12.1        | Power-Up and Power-Down .....                         | 43        |
| 12.2        | Reset Specifications .....                            | 43        |
| 12.3        | Power Supply Decoupling .....                         | 44        |
| <b>13.0</b> | <b>Maximum Ratings and Operating Conditions .....</b> | <b>45</b> |
| 13.1        | Absolute Maximum Ratings .....                        | 45        |
| 13.2        | Operating Conditions .....                            | 45        |
| <b>14.0</b> | <b>Electrical Specifications .....</b>                | <b>46</b> |
| 14.1        | DC Current Characteristics .....                      | 46        |
| 14.2        | DC Voltage Characteristics .....                      | 47        |
| <b>15.0</b> | <b>AC Characteristics .....</b>                       | <b>48</b> |
| 15.1        | AC Test Conditions .....                              | 48        |
| 15.2        | Capacitance .....                                     | 49        |
| 15.3        | AC Read Specifications .....                          | 50        |
| 15.4        | AC Write Specifications .....                         | 54        |
| 15.5        | Program and Erase Characteristics .....               | 58        |
| <b>16.0</b> | <b>Ordering Information .....</b>                     | <b>59</b> |
| <b>A</b>    | <b>Supplemental Reference Information .....</b>       | <b>61</b> |
| A.1         | Common Flash Interface .....                          | 61        |
| A.2         | Flowcharts .....                                      | 73        |
| A.3         | Write State Machine .....                             | 83        |
| <b>B</b>    | <b>Conventions - Additional Documentation .....</b>   | <b>87</b> |
| B.1         | Acronyms .....  | 87        |
| B.2         | Definitions and Terms .....                           | 87        |
| <b>C</b>    | <b>Revision History .....</b>                         | <b>89</b> |

## 1.0 Functional Description

### 1.1 Introduction

This document provides information about the Numonyx® Axcell™ P30-65nm Single Bit per Cell (SBC) Flash memory and describes its features, operations, and specifications.

P30-65nm SBC device is offered in 64-Mbit and 128-Mbit. Benefits include high-speed interface NOR device, and support for code and data storage. Features include high-performance synchronous-burst read mode, a dramatical improvement in buffer program time through larger buffer size, fast asynchronous access times, low power, flexible security options, and three industry-standard package choices.

P30-65nm SBC device is manufactured using 65nm process technology.

### 1.2 Overview

P30-65nm SBC device provides high performance 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 (RCR) enables synchronous burst-mode reads. In synchronous burst mode, output data is synchronized with a user-supplied clock signal. A WAIT signal provides easy CPU-to-flash memory synchronization.

In addition to the enhanced architecture and interface, the device incorporates technology that enables fast buffer program and erase operations. The device features a 256-word buffer to enable optimum programming performance, which can improve system programming throughput time significantly to 1.8MByte/s.

Designed for low-voltage systems, the P30-65nm SBC device supports read operations with VCC at 1.8V, and erase and program operations with VPP at 1.8V or 9.0V. Buffered Enhanced Factory Programming provides the fastest flash array programming performance with VPP at 9.0V, which increases factory throughput with 3.2Mbyte/s. With VPP at 1.8V, VCC and VPP can be tied together for a simple, ultra low power design. In addition to voltage flexibility, a dedicated VPP connection provides complete data protection when  $VPP \leq V_{PPLK}$ .

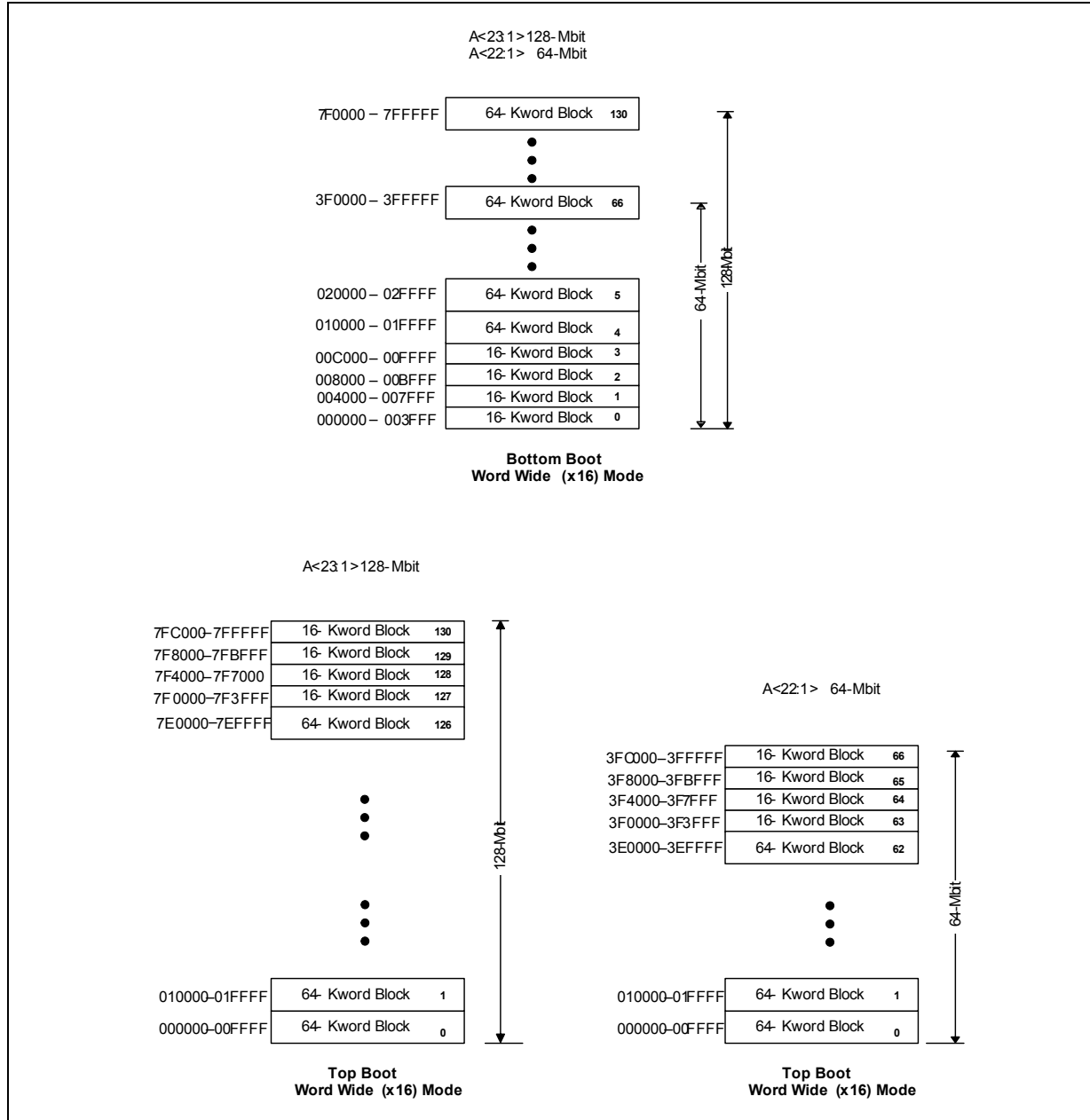
The Command User Interface is the interface between the system processor and all internal operations of the device. An internal Write State Machine automatically executes the algorithms and timings necessary for block erase and program. A Status Register indicates erase or program completion and any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

A device 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.

The OTP 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. The P30-65nm SBC device adds enhanced protection via Password Access; this new feature allows write and/or read access protection of user-defined blocks. In addition, the P30-65nm SBC device also has backward-compatible One-Time Programmable (OTP) permanent block locking security feature.

### 1.3 Memory Map

**Figure 1: P30-65nm SBC Memory Map (64-Mbit and 128-Mbit Densities)**



**Note:** A1 is the least significant address bit for TSOP and Easy BGA while A0 for the QUAD+ package. Unless otherwise indicated, for the purpose of brevity, this document will consolidate all discussions to A1 as the least significant Address bit.

## 2.0 Package Information

### 2.1 56-Lead TSOP

Figure 2: TSOP Mechanical Specifications

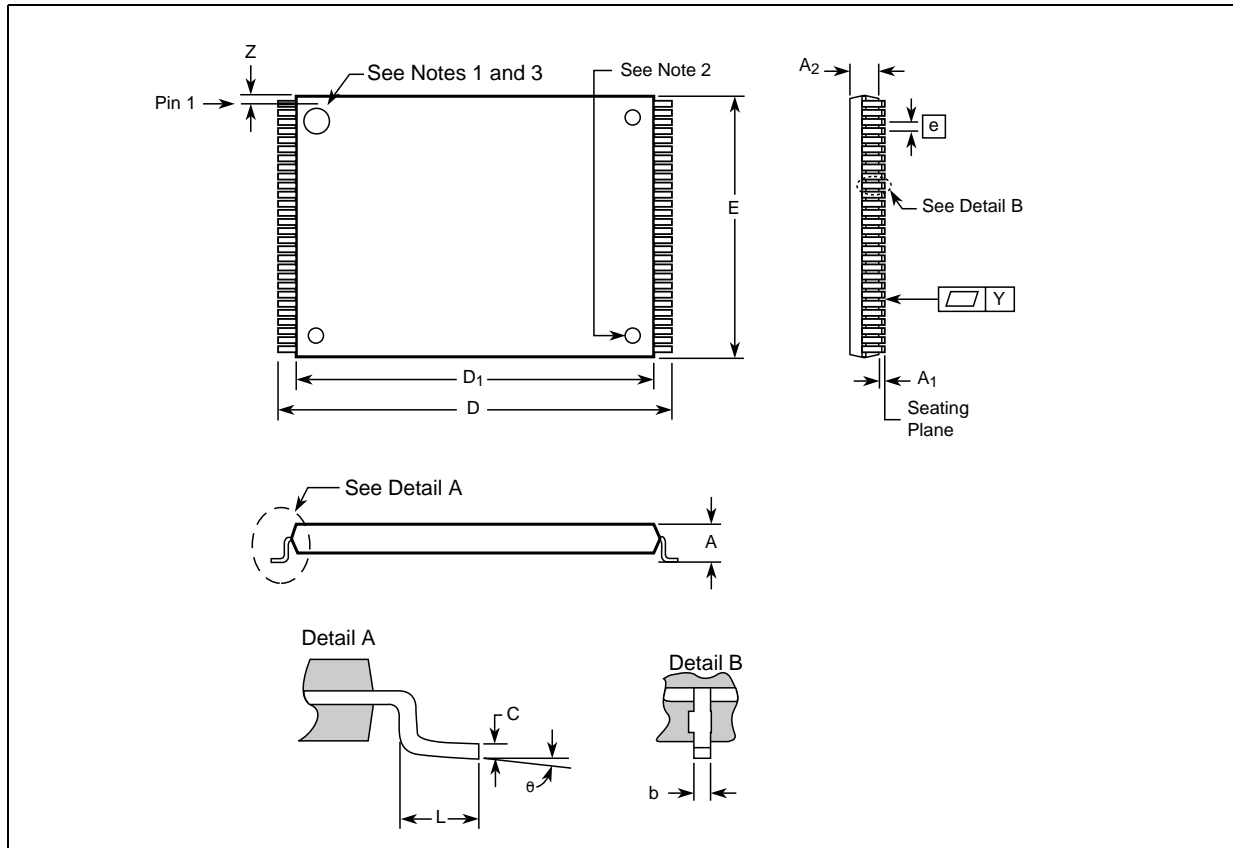


Table 1: TSOP Package Dimensions (Sheet 1 of 2)

| Product Information    | Symbol | Millimeters |        |        | Inches |        |       |
|------------------------|--------|-------------|--------|--------|--------|--------|-------|
|                        |        | Min         | Nom    | Max    | Min    | Nom    | Max   |
| Package Height         | A      | -           | -      | 1.200  | -      | -      | 0.047 |
| Standoff               | $A_1$  | 0.050       | -      | -      | 0.002  | -      | -     |
| Package Body Thickness | $A_2$  | 0.965       | 0.995  | 1.025  | 0.038  | 0.039  | 0.040 |
| Lead Width             | b      | 0.100       | 0.150  | 0.200  | 0.004  | 0.006  | 0.008 |
| Lead Thickness         | C      | 0.100       | 0.150  | 0.200  | 0.004  | 0.006  | 0.008 |
| Package Body Length    | $D_1$  | 18.200      | 18.400 | 18.600 | 0.717  | 0.724  | 0.732 |
| Package Body Width     | E      | 13.800      | 14.000 | 14.200 | 0.543  | 0.551  | 0.559 |
| Lead Pitch             | e      | -           | 0.500  | -      | -      | 0.0197 | -     |
| Terminal Dimension     | D      | 19.800      | 20.00  | 20.200 | 0.780  | 0.787  | 0.795 |
| Lead Tip Length        | L      | 0.500       | 0.600  | 0.700  | 0.020  | 0.024  | 0.028 |

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

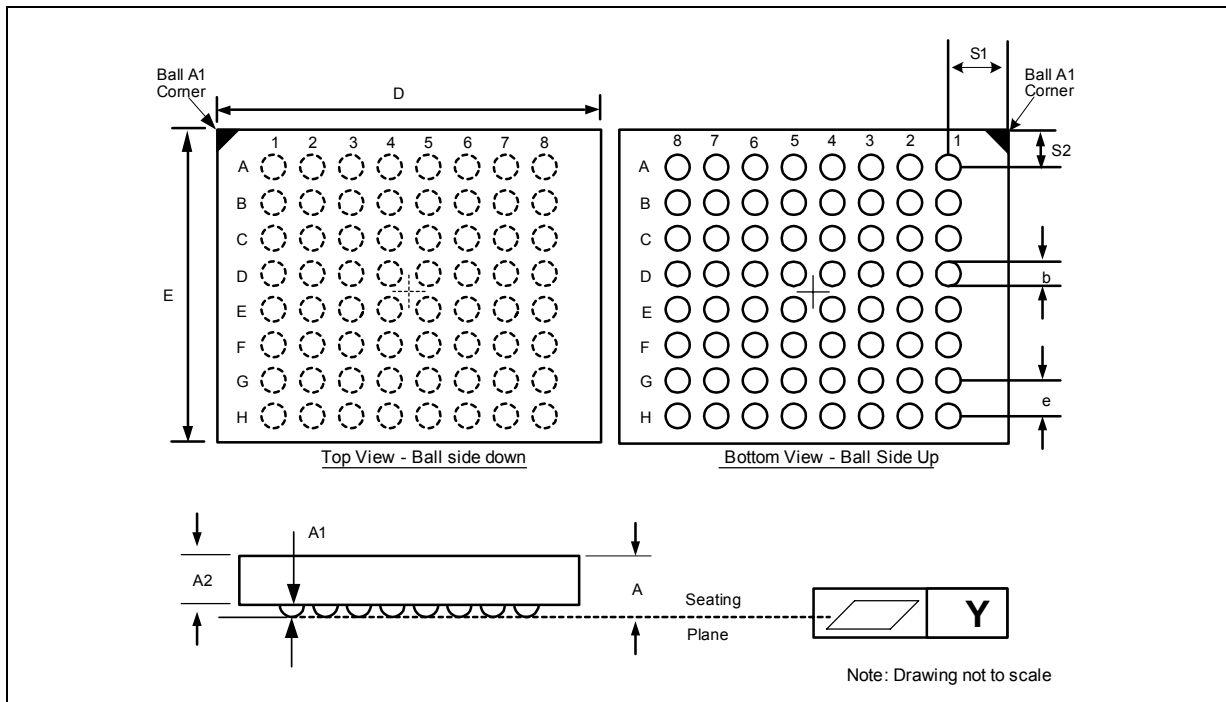
| Product Information       | Symbol   | Millimeters |       |       | Inches |       |       |
|---------------------------|----------|-------------|-------|-------|--------|-------|-------|
|                           |          | Min         | Nom   | Max   | Min    | Nom   | Max   |
| Lead Count                | N        | -           | 56    | -     | -      | 56    | -     |
| Lead Tip Angle            | $\theta$ | 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.

## 2.2 64-Ball Easy BGA Package

**Figure 3: Easy BGA Mechanical Specifications (10x13x1.2 mm)**



**Table 2: Easy BGA Package Dimensions for 10x13x1.2 mm (Sheet 1 of 2)**

| Product Information    | Symbol | Millimeters |        |        | Inches |        |        |
|------------------------|--------|-------------|--------|--------|--------|--------|--------|
|                        |        | Min         | Nom    | Max    | Min    | Nom    | Max    |
| Package Height         | A      | -           | -      | 1.200  | -      | -      | 0.0472 |
| Ball Height            | A1     | 0.250       | -      | -      | 0.0098 | -      | -      |
| Package Body Thickness | A2     | -           | 0.780  | -      | -      | 0.0307 | -      |
| Ball (Lead) Width      | b      | 0.310       | 0.410  | 0.510  | 0.0120 | 0.0160 | 0.0200 |
| Package Body Width     | D      | 9.900       | 10.000 | 10.100 | 0.3898 | 0.3937 | 0.3976 |

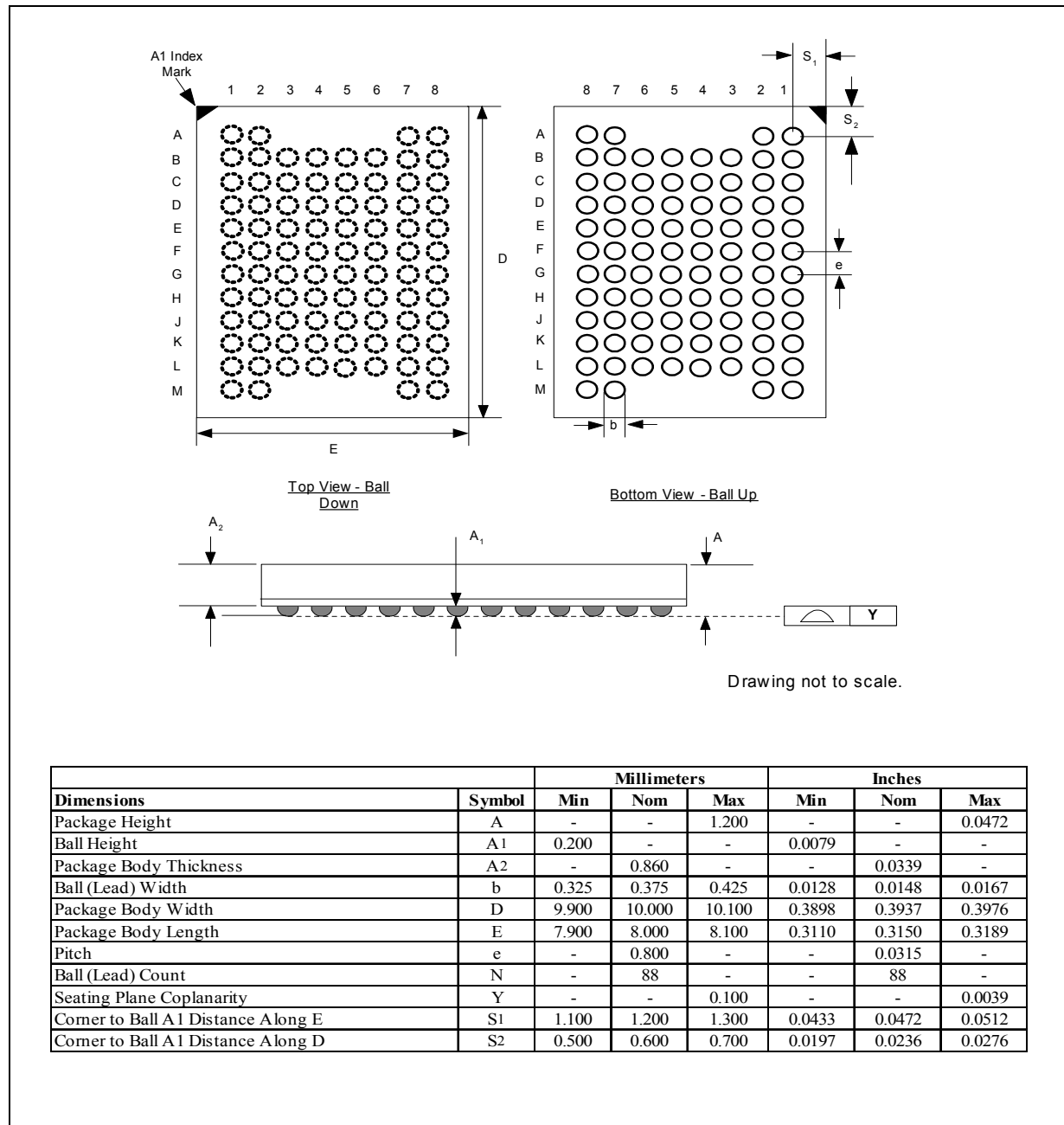
**Table 2: Easy BGA Package Dimensions for 10x13x1.2 mm (Sheet 2 of 2)**

| Product Information                | Symbol | Millimeters |        |        | Inches |        |        |
|------------------------------------|--------|-------------|--------|--------|--------|--------|--------|
|                                    |        | Min         | Nom    | Max    | Min    | Nom    | Max    |
| 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 |

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

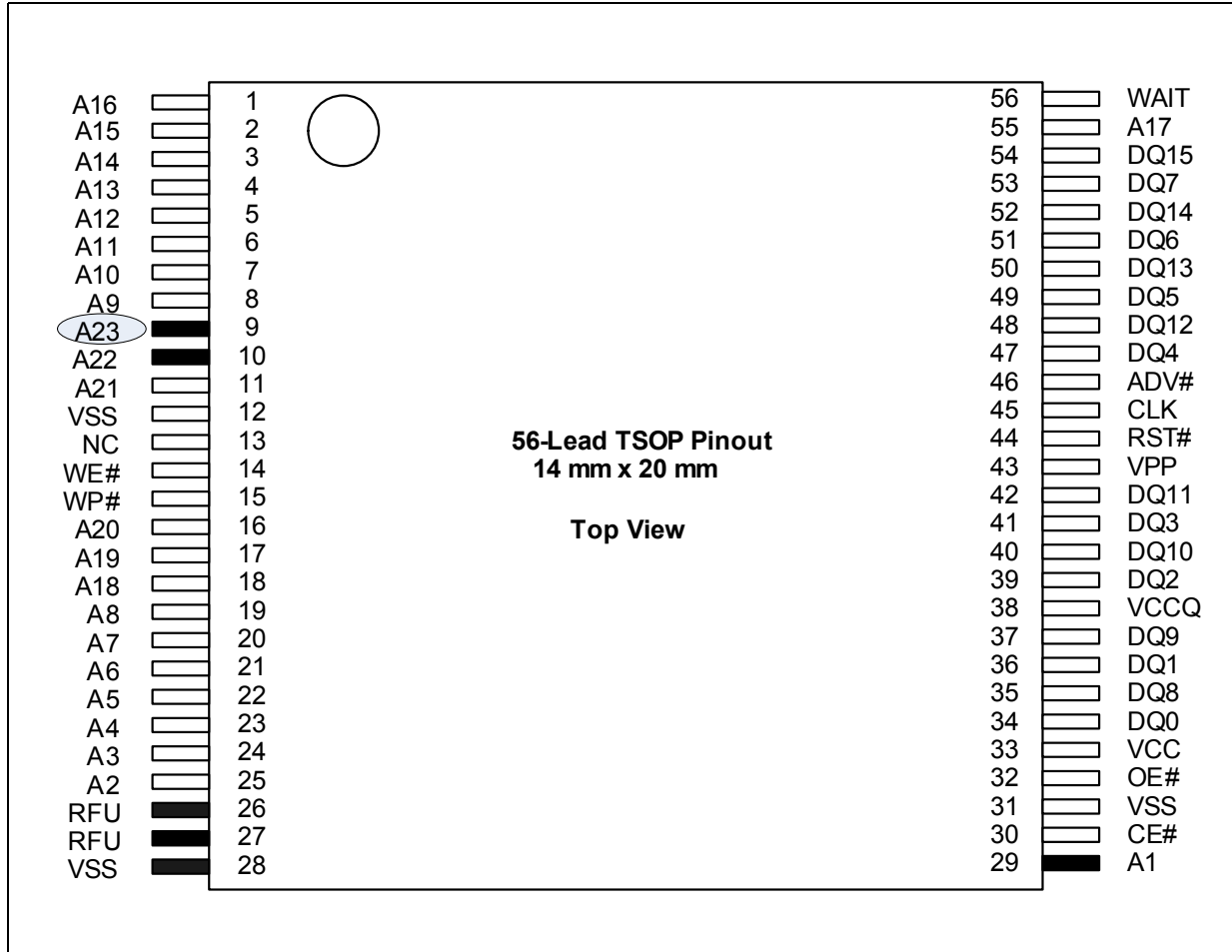
## 2.3 QUAD+ SCSP Packages

Figure 4: 128-Mbit, 88-ball (80 active) QUAD+ SCSP Specifications (8x10x1.2 mm)



### 3.0 Pinouts/Ballouts

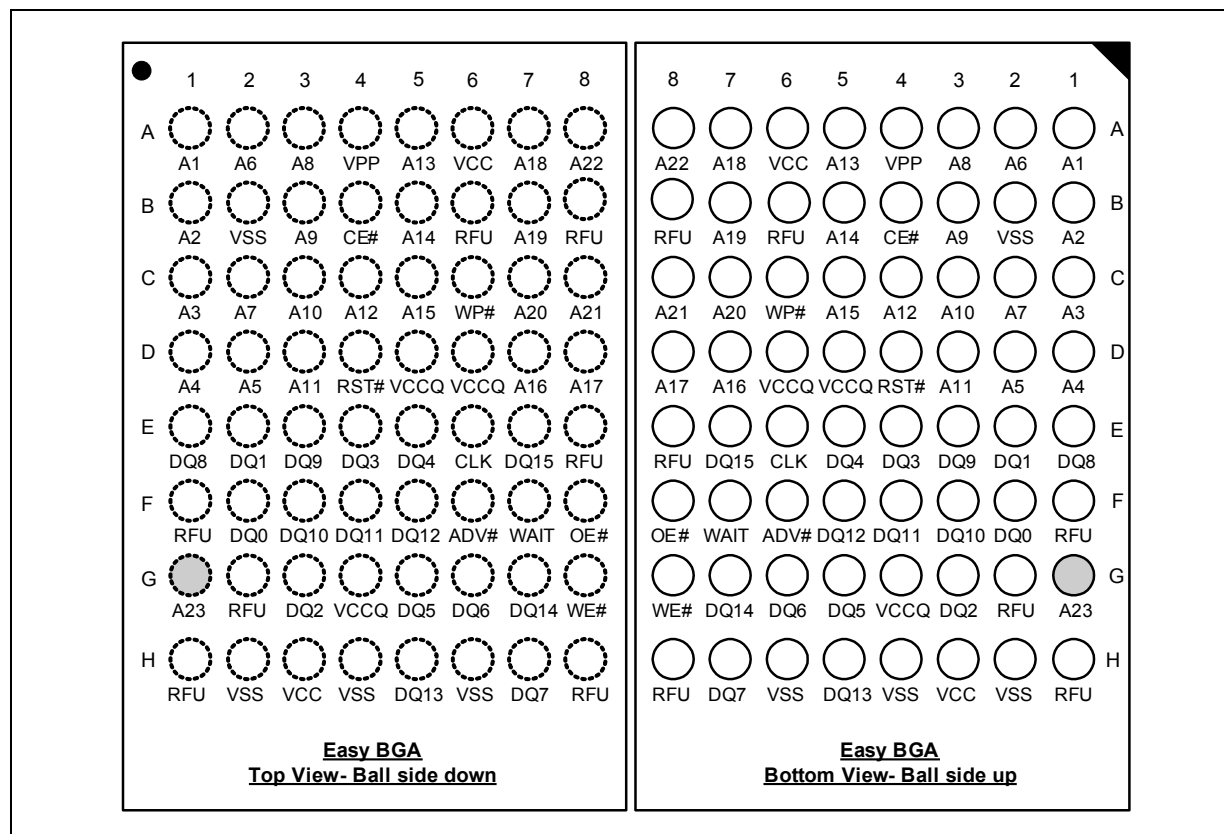
Figure 5: 56-Lead TSOP Pinout (64-Mbit and 128-Mbit Densities)



**Notes:**

1. A1 is the least significant address bit.
2. A23 is valid for 128-Mbit densities; otherwise, it is a no connect (NC).
3. A22 is valid for 64-Mbit densities and above; otherwise, it is a no connect (NC).
4. No Internal Connection on Pin 13; it may be driven or floated. For legacy 130nm designs, this pin can be tied to Vcc.
5. One dimple on package denotes Pin 1 which will always be in the upper left corner of the package, in reference to the product mark.

**Figure 6: 64-Ball Easy BGA Ballout (64-Mbit and 128-Mbit Densities)**



**Notes:**

1. A1 is the least significant address bit.
2. A23 is valid for 128-Mbit densities; otherwise, it is a no connect.
3. A22 is valid for 64-Mbit densities and above; otherwise, it is a no connect (NC).
4. One dimple on package denotes Pin 1 which will always be in the upper left corner of the package, in reference to the product mark.



## 4.0 Signals

**Table 3: TSOP and Easy BGA Signal Descriptions**

| Symbol   | Type             | Name and Function  |
|----------|------------------|--|
| A[MAX:1] | Input            | <b>ADDRESS INPUTS:</b> Device address inputs. 128-Mbit: A[23:1], 64-Mbit: A[22:1].   |
| DQ[15:0] | Input/<br>Output | <b>DATA INPUT/OUTPUTS:</b> Inputs data and commands during write cycles; outputs data during reads of memory, Status Register, OTP Register, and Read Configuration Register. 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><b>WARNING:</b> Designs not using ADV# must tie it to VSS to allow addresses to flow through.   |
| CE#      | Input            | <b>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><b>WARNING:</b> Chip Enable must be 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><b>WARNING:</b> 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. 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 in valid 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/<br>Input  | <b>ERASE AND PROGRAM POWER:</b> A valid voltage on this pin allows erasing or programming. Memory contents cannot be altered when $VPP \leq V_{ppLK}$ . Block erase and program at invalid VPP voltages should not be attempted.<br>Set $VPP = V_{ppL}$ for in-system program and erase operations. To accommodate resistor or diode drops from the system supply, the $V_{IH}$ level of VPP can be as low as $V_{ppL}$ min. VPP 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 array blocks for 1000 cycles maximum and to parameter blocks for 2500. 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. |
| VCC      | Power            | <b>DEVICE CORE POWER SUPPLY:</b> Core (logic) source voltage. Writes to the flash array are inhibited when $VCC \leq V_{LKO}$ . Operations at invalid VCC 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 Don't Use (DU) signal.   |
| DU       | —                | <b>DON'T 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 4: QUAD+ SCSP Signal Descriptions

| Symbol   | Type             | Name and Function  |
|----------|------------------|--|
| A[MAX:0] | Input            | <b>ADDRESS INPUTS:</b> Device address inputs. 128-Mbit: A[22:0].<br>Note: Unlike TSOP and Easy BGA, A0 is the least significant address bit for the QUAD+ package. Unless otherwise indicated, for the purpose of brevity, this document will consolidate all discussions to A1 as the least significant Address bit.  |
| DQ[15:0] | Input/<br>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.<br>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. F1-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. F1-OE# low enables the device's output data buffers during read cycles. F1-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 <sub>OL</sub> or V <sub>OH</sub> when F1-CE# and F1-OE# are V <sub>IL</sub> . WAIT is high-Z if F1-CE# or F1-OE# is V <sub>IH</sub> .<br><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/<br>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 <sub>PP</sub> 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 <sub>IH</sub> level of V <sub>PP</sub> can be as low as V <sub>PPL</sub> min. V <sub>PP</sub> must remain above V <sub>PPL</sub> min to perform in-system flash modification. VPP may be 0 V during read operations.<br>V <sub>PPH</sub> 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. |
| 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 <sub>CC</sub> 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.   |

## 5.0 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.

Bus cycles to/from the P30-65nm SBC device conform to standard microprocessor bus operations. Table 5, "Bus Operations Summary" summarizes the bus operations and the logic levels that must be applied to the device control signal inputs.

**Table 5: Bus Operations Summary**

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

**Notes:**

1. Refer to the Table 7, "Command Bus Cycles" on page 21 for valid DQ[15:0] during a write operation.
2. X = Don't Care (H or L).
3. RST# must be at V<sub>SS</sub> ± 0.2V to meet the maximum specified power-down current.

### 5.1 Read - Asynchronous Mode

To perform an asynchronous page or single word read, an address is driven onto the address bus, and CE# is asserted. ADV# can be driven high to latch the address, or it must be held low throughout the read cycle. WE# and RST# must already have been deasserted. WAIT is set to a deasserted state during asynchronous page mode and single word mode as determined by RCR.10. CLK is not used for asynchronous page-mode reads, and is ignored. After OE# is asserted, the data is driven onto DQ[15:0] after an initial access time t<sub>AVQV</sub> or t<sub>GLQV</sub> delay. (See Table 25, "AC Read Specifications" on page 50).

**Note:** If only asynchronous reads are to be performed, CLK should be tied to a valid V<sub>IH</sub> level, WAIT signal can be floated and ADV# must be tied to ground.

In asynchronous page mode, eight 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 three address bits determine which word of the 8-word page is output from the data buffer at any given time.

Refer to the following waveforms for more detailed information: Figure 19, "Asynchronous Single-Word Read (ADV# Low)" on page 51, and Figure 20, "Asynchronous Single-Word Read (ADV# Latch)" on page 52, and Figure 21, "Asynchronous Page-Mode Read Timing" on page 52.

### 5.2 Read - Synchronous Mode

To perform a synchronous burst read on array or non-array, an initial address is driven onto the address bus, and CE# is 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. Once OE# is asserted, the first word is driven onto DQ[15:0] on the next valid CLK edge after initial access latency delay (see [Section 11.2.2, "Latency Count \(RCR\[13:11\]\)" on page 36](#)). Subsequent data is output on valid CLK edges following a minimum delay  $T_{CHQV}$  (see [Table 25, "AC Read Specifications" on page 50](#)).

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.

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.

Refer to the following waveforms for more detailed information: [Figure 22, "Synchronous Single-Word Array or Non-array Read Timing" on page 53](#), and [Figure 23, "Continuous Burst Read, showing an Output Delay Timing" on page 53](#), and [Figure 24, "Synchronous Burst-Mode Four-Word Read Timing" on page 54](#).

### 5.3 Write

To perform a write operation, both CE# and WE# are asserted while RST# and OE# are deasserted. During a write operation, address and data are latched on the rising edge of WE# or CE#, whichever occurs first. [Table 7, "Command Bus Cycles" on page 21](#) shows the bus cycle sequence for each of the supported device commands, while [Table 6, "Command Codes and Definitions" on page 19](#) describes each command. See [Table 26, "AC Write Specifications" on page 54](#) for signal-timing details.

When the device is operating in write operations, WAIT is set to a deasserted state as determined by RCR.10.

*Note:* Write operations with invalid VCC and/or VPP voltages can produce spurious results and should not be attempted.

### 5.4 Output Disable

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

### 5.5 Standby

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

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

## 5.6 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.

After initial power-up or reset, the device defaults to asynchronous Read Array mode, and the Status Register is set to 0x80.

When RST# is driven low (RST# asserted), the flash device enters reset mode. Then all internal circuits are de-energized, and the output drivers are placed in High-Z. 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. A device reset also clears the Status Register. See [Table 18, "Power and Reset" on page 43](#) for RST# timing detail.

When RST# is driven high (RST# deasserted), a minimum wait is required before the flash device is able to perform normal operations. Please consider  $T_{PHQV}$  (R5) and  $T_{PHWL}$  (W1) during system design. see [Table 25, "AC Read Specifications" on page 50](#), and [Section 26, "AC Write Specifications" on page 54](#). After this wake-up interval passes, normal operation is ready for execution.

## 6.0 Command Set

### 6.1 Device Command Codes

The flash Command User Interface (CUI) provides access to device read, write, and erase operations. The CUI does not occupy an addressable memory location; it is part of the internal logic which allows the flash device to be controlled. The Write State Machine provides the management for its internal erase and program algorithms.

Commands are written to the CUI to control flash device operations. [Table 6, "Command Codes and Definitions"](#) describes all valid command codes.

For operations that involve multiple command cycles, the possibility exists that the subsequent command does not get issued in the proper sequence. When this happens, the CUI sets Status Register bits SR[5,4] to indicate a command sequence error.

**Table 6: 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. SR data is output on DQ[7:0].  |
|       | 0x90 | Read Device ID or Read Configuration Register (RCR) | Places device in Read Device Identifier mode. Subsequent reads output manufacturer/device codes, Configuration Register data, Block Lock status, or OTP Register data on DQ[15:0].   |
|       | 0x98 | Read CFI  | Places the device in Read Query mode. Subsequent reads output Common Flash Interface (CFI) information on DQ[7:0].   |
|       | 0x50 | Clear Status Register                               | The WSM can only set SR 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 SR Array Data for synchronous Non-array reads. The Read Array command must be issued to read array data after programming has finished. |
|       | 0xE8 | Buffered Program                                    | This command loads a variable number of words up to the buffer size of 256 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 the BEFP mode. The CUI then waits for the BEFP Confirm command, 0xD0, that initiates the BEFP algorithm. All other commands are ignored when BEFP mode begins.   |
|       | 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.  |
| Erase | 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.  |
|       | 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 SR Data for synchronous Non-array reads.   |

**Table 6: Command Codes and Definitions (Sheet 2 of 2)**

| Mode          | Code | Device Mode                                 | Description  |
|---------------|------|---|--|
| Suspend       | 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 WSM 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.  |
| Protection    | 0x60 | Block Lock Setup                            | First cycle of a 2-cycle command; prepares the CUI for block lock configuration changes.   |
|               | 0x01 | Block Lock                                  | If the previous command was Block Lock Setup (0x60), the addressed block is locked.  |
|               | 0xD0 | Block Unlock                                | 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 | Block Lock-Down                             | If the previous command was Block Lock Setup (0x60), the addressed block is locked down.   |
|               | 0xC0 | OTP Register or Lock Register program setup | First cycle of a 2-cycle command; prepares the device for a OTP Register or Lock Register program operation. The second cycle latches the register address and data, and starts the programming algorithm to program data into the OTP array.  |
| Configuration | 0x60 | Read Configuration Register Setup           | First cycle of a 2-cycle command; prepares the CUI for device read configuration.  |
|               | 0x03 | Read Configuration Register                 | If the previous command was Read Configuration Register Setup (0x60), the CUI latches the address and writes A[16:1] to the Read Configuration Register. Following a Configure RCR command, subsequent read operations access array data.  |
| Blank Check   | 0xBC | Block Blank Check                           | First cycle of a 2-cycle command; initiates the Blank Check operation on a array block.  |
|               | 0xD0 | Block Blank Check Confirm                   | Second cycle of blank check command sequence; it latches the block address and executes blank check on the array block.  |
| EFI           | 0xEB | Extended Function Interface                 | This command is used in extended function interface. first cycle of a multiple-cycle command second cycle is a Sub-Op-Code, the data written on third cycle is one less than the word count; the allowable value on this cycle are 0 through 511. The subsequent cycles load data words into the program buffer at a specified address until word count is achieved.                     |

## 6.2 Device Command Bus Cycles

Device operations are initiated by writing specific device commands to the CUI. See [Table 7, "Command Bus Cycles" on page 21](#). 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 7: 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                  |
|               | Read CFI  | ≥ 2        | Write           | DnA                 | 0x98                | Read             | DBA + CFI-A         | CFI-D               |
|               | Read Status Register                                    | 2          | Write           | DnA                 | 0x70                | Read             | DnA                 | SRD                 |
|               | Clear Status Register                                   | 1          | Write           | DnA                 | 0x50                | -                | -                   | -                   |
| Program       | Word Program  | 2          | Write           | WA                  | 0x40                | 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                | -                | -                   | -                   |
| Protection    | Block Lock  | 2          | Write           | BA                  | 0x60                | Write            | BA                  | 0x01                |
|               | Block Unlock  | 2          | Write           | BA                  | 0x60                | Write            | BA                  | 0xD0                |
|               | Block Lock-down   | 2          | Write           | BA                  | 0x60                | Write            | BA                  | 0x2F                |
|               | Program OTP Register                                    | 2          | Write           | OTP-RA              | 0xC0                | Write            | OTP-RA              | OTP-D               |
|               | Program Lock Register                                   | 2          | Write           | LRA                 | 0xC0                | Write            | LRA                 | LRD                 |
| Configuration | Configure Read Configuration Register                   | 2          | Write           | RCD                 | 0x60                | Write            | RCD                 | 0x03                |
| Blank Check   | Block Blank Check                                       | 2          | Write           | BA                  | 0xBC                | Write            | BA                  | D0                  |
| EFI           | Extended Function Interface <sup>(5)</sup>              | >2         | Write           | WA                  | 0xEB                | Write            | WA                  | Sub-Op code         |

**Notes:**

- First command cycle address should be the same as the operation's target address.  
DBA = Device Base Address.  
DnA = Address within the device.  
IA = Identification code address offset.  
CFI-A = Read CFI address offset.  
WA = Word address of memory location to be written.  
BA = Address within the block.  
OTP-RA = OTP Register address.  
LRA = Lock Register address.  
RCD = Read Configuration Register data on A[16:1] for TSOP and BGA package; on A[15:0] for QUAD+ package.
- ID = Identifier data.  
CFI-D = CFI data on DQ[15:0].  
SRD = Status Register data.  
WD = Word data.  
N = Word count of data to be loaded into the write buffer.  
OTP-D = OTP Register data.  
LRD = Lock Register data.
- 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 256 words of data. Then the confirm command (0xD0) is issued, triggering the array programming operation.
- The confirm command (0xD0) is followed by the buffer data.
- The second cycle is a Sub-Op-Code, the data written on third cycle is N-1;  $1 \leq N \leq 256$ . The subsequent cycles load data words into the program buffer at a specified address until word count is achieved. After the data words are loaded, the final cycle is the confirm cycle 0xD0).

## 7.0 Read Operation

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 mode. To change the read state, the appropriate read command must be written to the device (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)). The following sections describe read-mode operations in detail.

In order to enable synchronous burst reads, the RCR must be configured. Please see [Section 11.2, "Read Configuration Register \(RCR\)" on page 34](#) for RCR detail. Please refer to [Section 5.1, "Read - Asynchronous Mode" on page 16](#) and [Section 5.2, "Read - Synchronous Mode" on page 16](#) for bus operation detail. See [Section 25, "AC Read Specifications" on page 50](#) for timing specification.

### 7.1 Read Array

Following a device power-up or reset, the device is set to Read Array mode. 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. Please refer to [Section 5.1, "Read - Asynchronous Mode" on page 16](#) and [Section 5.2, "Read - Synchronous Mode" on page 16](#) for bus operation detail. See [Section 25, "AC Read Specifications" on page 50](#) for timing specification.

### 7.2 Read Device Identifier

The Read Device Identifier command instructs the device to output manufacturer code, device identifier code, block-lock status, OTP Register data, or Read Configuration Register data (see [Section 6.2, "Device Command Bus Cycles" on page 20](#) for details on issuing the Read Device Identifier command). [Table 8, "Device Identifier Information" on page 22](#) and [Table 9, "Device ID codes" on page 23](#) show the address offsets and data values for this device.

**Table 8: Device Identifier Information (Sheet 1 of 2)**

| Item  | Address <sup>(1,2)</sup>  | Data(x16)   |
|---|---------------------------|---|
| Manufacturer Code   | 0x00                      | 0x89h   |
| Device ID Code  | 0x01                      | ID (See <a href="#">Table 9</a> )                             |
| Block Lock Configuration:<br><ul style="list-style-type: none"> <li>• Block Is Unlocked</li> <li>• Block Is Locked</li> <li>• Block Is not Locked-Down</li> <li>• Block Is Locked-Down</li> </ul> | BBA <sup>(1)</sup> + 0x02 | Lock Bit:<br>DQ0 = 0b0<br>DQ0 = 0b1<br>DQ1 = 0b0<br>DQ1 = 0b1 |
| Read Configuration Register   | 0x05                      | RCR Contents  |
| General Purpose Register <sup>(3)</sup>   | DBA <sup>(2)</sup> + 0x07 | GPR data  |
| Lock Register 0   | 0x80                      | PR-LK0  |
| 64-bit Factory-Programmed OTP Register  | 0x81–0x84                 | Numonyx Factory OTP Register data                             |
| 64-bit User-Programmable OTP Register   | 0x85–0x88                 | User OTP Register data  |

**Table 8: Device Identifier Information (Sheet 2 of 2)**

| Item                                    | Address <sup>(1,2)</sup> | Data(x16)                     |
|---|--------------------------|-------------------------------|
| Lock Register 1                         | 0x89                     | PR-LK1 OTP Register lock data |
| 128-bit User-Programmable OTP Registers | 0x8A-0x109               | User OTP Register data        |

**Notes:**

1. BBA = Block Base Address.
2. DBA = Device base Address, Numonyx reserves other configuration address locations.
3. The GPR is used as read out register for Extended Function interface command.

**Table 9: Device ID codes**

| ID Code Type | Device Density | Device Identifier Codes |
|--------------|----------------|-------------------------|
| Top Boot     | 64-Mbit        | 8817                    |
|              | 128-Mbit       | 8818                    |
| Bottom Boot  | 64-Mbit        | 881A                    |
|              | 128-Mbit       | 881B                    |

## 7.3 Read CFI

The Read CFI command instructs the device to output Common Flash Interface data when read. See [Figure 6.1, "Device Command Codes" on page 19](#). [Section A.1, "Common Flash Interface" on page 61](#) shows CFI information and address offsets within the CFI database.

## 7.4 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. SRD 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. the Read Array command).

The Status Register is read using single asynchronous-mode or synchronous burst mode reads. SRD 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, when reading the Status Register in synchronous burst mode, CE# or ADV# must be toggled to update SRD.

The Device Ready Status bit (SR.7) provides overall status of the device. SR[6:1] present status and error information about the program, erase, suspend, VPP, and block-locked operations.

See [Table 12, "Status Register Description" on page 34](#) for the description of the Status Register.

## 7.5 Clear Status Register

The Clear Status Register command clears the Status Register. It functions independent of VPP. The WSM sets and clears SR.7, 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.

## 8.0 Program Operation

The device supports three programming methods: Word Programming (40h/10h), Buffered Programming (E8h, D0h), and Buffered Enhanced Factory Programming (80h, D0h). 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[4,1] set) and termination of the operation. See [Section 10.0, "Security" on page 31](#) for details on locking and unlocking blocks.

### 8.1 Word Programming

Word programming operations are initiated by writing the Word Program Setup command to the device. 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 32, "Word Program Flowchart" on page 73](#). VPP must be above  $V_{PPLK}$ , and within the specified  $V_{PPL}$  min/max values.

During programming, the 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.

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 Read Status Register, Read Device Identifier, Read CFI, 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 VPP 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.

### 8.2 Buffered Programming

The device features a 256-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. (see [Figure 35, "Buffer Program Flowchart" on page 76](#)).

When the Buffered Programming Setup command is issued, 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.

*Note:* *The device defaults to output SR data after the Buffered Programming Setup Command (E8h) is issued. CE# or OE# must be toggled to update Status Register. Don't issue the*

Read SR command (70h), which would be interpreted by the internal state machines as Buffer Word Count.

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 256-word boundary ( $A[8:1] = 0x00$ ).

*Note:* If a misaligned address range is issued during buffered programming, the program region must also be within an 256-word aligned boundary.

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 an error occurs while writing to the array, the device stops programming, and 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 13.2, "Operating Conditions" on page 45 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 SR[5,4] are set.

If Buffered programming is attempted while  $V_{PP}$  is at or below  $V_{PPLK}$ , 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.

### 8.3 Buffered Enhanced Factory Programming

Buffered Enhanced Factory Programming (BEFP) speeds up the flash programming performance. 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 37, "BEFP Flowchart" on page 78). It uses a write buffer to spread up the program performance across 256 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 256 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 256-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.

### 8.3.1 BEFP Requirements and Considerations

**Table 10: BEFP Requirements**

| Parameter/Issue   | Requirement  | Notes |
|-------------------|--|-------|
| Case Temperature  | $T_C = 30^{\circ}\text{C} \pm 10^{\circ}\text{C}$  | -     |
| VCC               | Nominal Vcc  | -     |
| VPP               | Driven to $V_{PPH}$  | -     |
| Setup and Confirm | Target block must be 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:** Word buffer boundaries in the array are determined by A[8:1] (0x00 through 0xFF). The alignment start point is A[8:1] = 0x000.

**Table 11: BEFP Considerations**

| Parameter/Issue                    | Requirement   | Notes |
|------------------------------------|---|-------|
| Cycling                            | For optimum performance, cycling must be limited below 50 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     |

**Notes:**

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 256, remaining locations must be filled with 0xFFFF.

### 8.3.2 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, VPP 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 VPP 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.

### 8.3.3 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 256 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 256, 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.

### 8.3.4 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.

## 8.4 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 36, "Program Suspend/Resume Flowchart" on page 77](#)).

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 15.5, "Program and Erase Characteristics" on page 58](#).

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

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

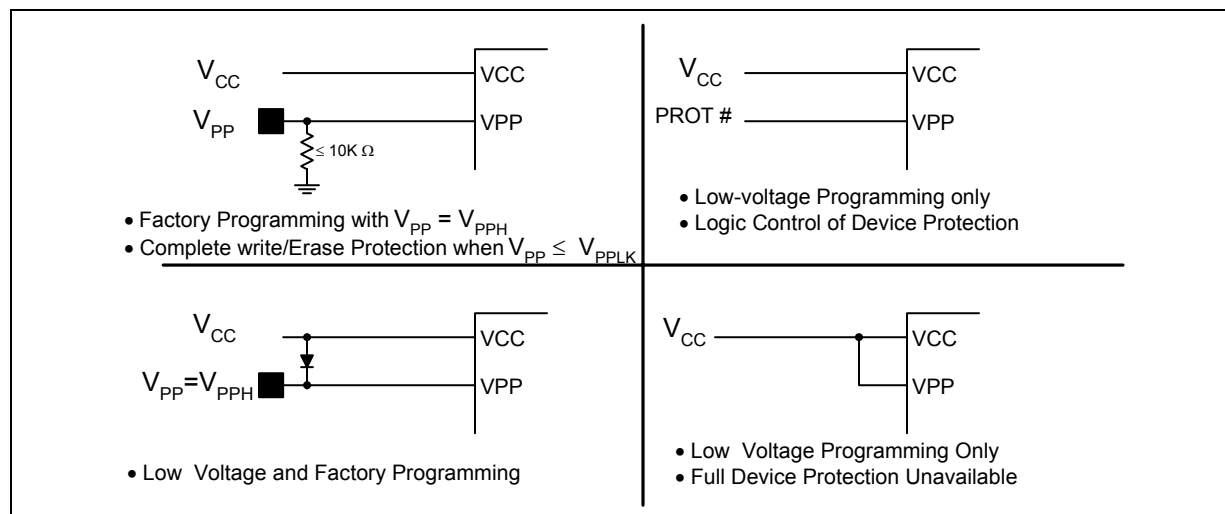
### 8.5 Program Resume

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 33, "Program Suspend/Resume Flowchart" on page 74).

### 8.6 Program Protection

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

Figure 8: Example VPP Supply Connections



## 9.0 Erase Operation

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.

### 9.1 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 6.2, "Device Command Bus Cycles" on page 20](#)). 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. VPP must be above  $V_{PPLK}$  and the block must be unlocked (see [Figure 38, "Block Erase Flowchart" on page 79](#)).

During a block erase, the 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 block array data that are ones can be changed to zeros by programming block.

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 VPP 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.

### 9.2 Blank Check

The Blank Check operation determines whether a specified array block is blank (i.e. completely erased). Without Blank Check, Block Erase would be the only other way to ensure a block is completely erased. Blank Check is especially useful in the case of erase operation interrupted by a power loss event.

Blank check can apply to only one block at a time, and no operations other than Status Register Reads are allowed during Blank Check (e.g. reading array data, program, erase etc). Suspend and resume operations are not supported during Blank Check, nor is Blank Check supported during any suspended operations.

Blank Check operations are initiated by writing the Block Blank Check command to the block address. Next, the Blank Check Confirm command is issued along with the same block address. When a successful command sequence is entered, the device automatically enters the Read Status State. The WSM then reads the entire specified block, and determines whether any bit in the block is programmed or over-erased.

The Status Register can be examined for Blank Check progress and errors by reading any address within the block being accessed. During a blank check operation, the Status Register indicates a busy status (SR.7 = 0). Upon completion, the Status

Register indicates a ready status ( $SR.7 = 1$ ). The Status Register should be checked for any errors, and then cleared. If the Blank Check operation fails, which means the block is not completely erased, the Status Register bit SR.5 will be set ("1"). CE# or OE# toggle (during polling) updates the Status Register.

After examining the Status Register, it should be cleared by the Clear Status Register command before issuing a new command. The device remains in Status Register Mode until another command is written to the device. Any command can follow once the Blank Check command is complete.

### 9.3 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 34, "Erase Suspend/Resume Flowchart" on page 75](#)).

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 15.5, "Program and Erase Characteristics" on page 58](#).

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, Read CFI, 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. VPP must remain at a valid level, and WP# must remain unchanged while in erase suspend. If RST# is asserted, the device is reset.

### 9.4 Erase Resume

The Erase Resume command instructs the device to continue erasing, and automatically clears 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.

### 9.5 Erase Protection

When  $VPP = V_{IL}$ , absolute hardware erase protection is provided for all device blocks. If  $VPP \leq V_{pPLK}$ , erase operations halt and SR.3 is set indicating a VPP-level error.

## 10.0 Security

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

### 10.1 Block Locking

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, VPP data security can be used to inhibit program and erase operations (see [Section 8.6, "Program Protection" on page 28](#) and [Section 9.5, "Erase Protection" on page 30](#)).

#### 10.1.1 Lock Block

To lock a block, issue the Block Lock Setup command. The next command must be the Block Lock command issued to the desired block's address (see [Section 6.2, "Device Command Bus Cycles" on page 20](#) and [Figure 39, "Block Lock Operations Flowchart" on page 80](#)). If the Configure 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 VPP. The block lock bits may be modified and/or read even if VPP is at or below  $V_{PPLK}$ .

#### 10.1.2 Unlock Block

The Block Unlock command is used to unlock blocks (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)). 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 9, "Block Locking State Diagram" on page 32](#)).

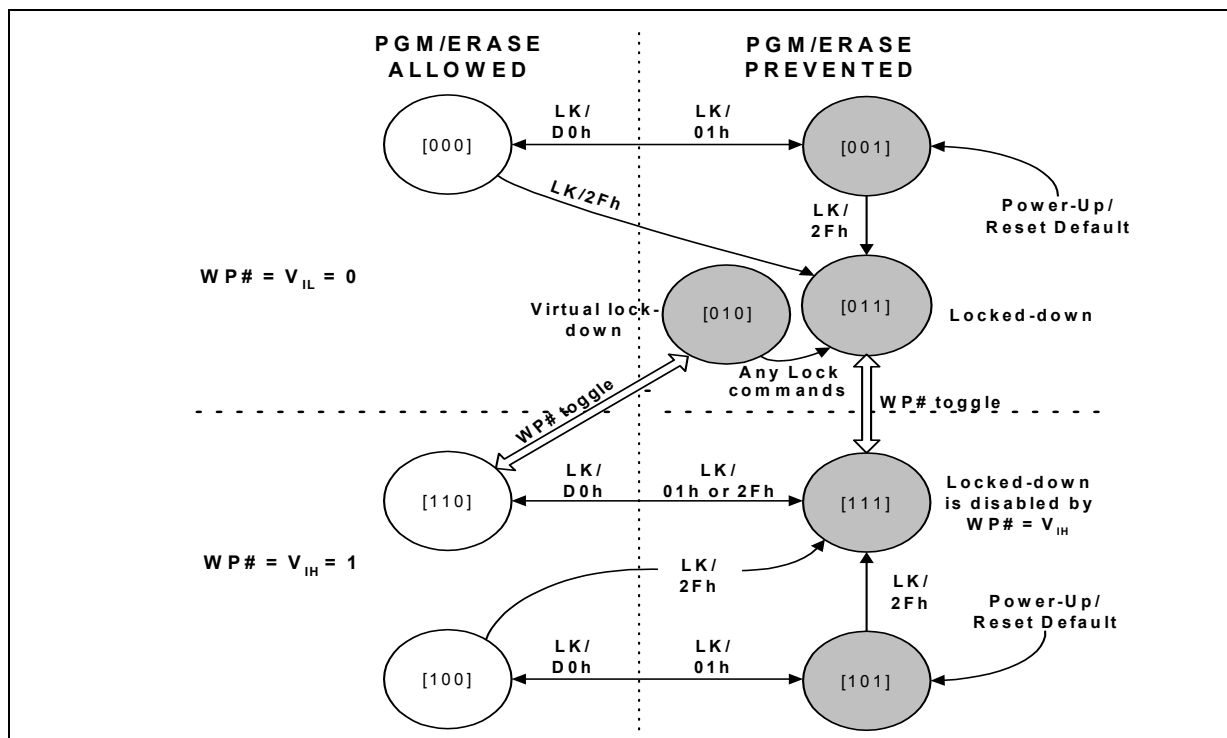
#### 10.1.3 Lock-Down Block

A locked or unlocked block can be locked-down by writing the Block Lock-Down command sequence (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)). 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 Block Unlock command with WP# deasserted. To return an unlocked block to locked-down state, a Block 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 9, "Block Locking State Diagram" on page 32](#)).

#### 10.1.4 Block Lock Status

The Read Device Identifier command is used to determine a block's lock status (see [Section 7.2, "Read Device Identifier" on page 22](#)). 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 9: Block Locking State Diagram



**Note:** LK: Lock Setup Command, 60h; LK/D0h: Unlock Command; LK/01h: Lock Command; LK/2Fh: Lock-Down Command.

### 10.1.5 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 83](#), which shows valid commands during an erase suspend.

## **10.2 Selectable OTP Blocks**

P30-65nm SBC device is backward-compatible with the OTP permanent block lock security feature of the legacy P30-130nm device. Blocks from the main array can be optionally configured as OTP. Ask your local Numonyx Sales representative for details about these selectable OTP implementations.

## **10.3 Password Access**

The Password Access is a security enhancement offered on the P30-65nm SBC device. This feature protects information stored in array blocks by preventing content alteration or reads until a valid 64-bit password is received. The Password Access may be combined with Flexible block blocking to create a multi-tiered solution.

Please contact representative Numonyx Sales for further details concerning the Password Access.

## 11.0 Register

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.

### 11.1 Status Register (SR)

The Status Register provides the ready/busy information of the device, as well as the error information about the program, erase, VPP and block-locked operations. please refer to [Section 7.4, "Read Status Register" on page 23](#) and [Section 7.5, "Clear Status Register" on page 23](#) for detail operations.

**Table 12: Status Register Description**

| Status Register (SR) |                                   |                        |   |   |  |                     | Default Value = 0x80 |  |
|----------------------|-----------------------------------|------------------------|---|---|--|---------------------|----------------------|--|
| Device Ready Status  | Erase Suspend Status <sup>1</sup> | Erase Status           | Program Status  | VPP Status                                | Program Suspend Status                 | Block-Locked Status | BEFP Write Status    |  |
| DRS                  | ESS                               | ES                     | PS  | VPPS                                      | PSS                                    | BLS                 | BWS                  |  |
| 7                    | 6                                 | 5                      | 4   | 3   | 2                                      | 1                   | 0                    |  |
| Bit                  | Name                              |                        | Description   |   |  |                     |                      |  |
| 7                    | Device Ready Status               |                        | 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              |                        | 0 = Erase suspend not in effect.<br>1 = Erase suspend in effect.  |   |  |                     |                      |  |
| 5                    | Erase Status                      | Command Sequence Error | SR.5  | SR.4                                      | Description                            |                     |                      |  |
| 4                    | Program Status                    |                        | 0   | 0   | Program or Erase operation successful. |                     |                      |  |
|                      |                                   |                        | 0   | 1   | Program error - operation aborted.     |                     |                      |  |
|                      |                                   |                        | 1   | 0   | Erase error - operation aborted.       |                     |                      |  |
|                      |                                   | 1                      | 1   | Command sequence error - command aborted. |  |                     |                      |  |
| 3                    | VPP Status                        |                        | 0 = VPP within acceptable limits during program or erase operation.<br>1 = $VPP \leq V_{PPLK}$ during program or erase operation. |   |  |                     |                      |  |
| 2                    | Program Suspend Status            |                        | 0 = Program suspend not in effect.<br>1 = Program suspend in effect.  |   |  |                     |                      |  |
| 1                    | Block-Locked Status               |                        | 0 = Block not locked during program or erase.<br>1 = Block locked during program or erase; operation aborted.                     |   |  |                     |                      |  |
| 0                    | BEFP Write Status <sup>2</sup>    |                        | After Buffered Enhanced Factory Programming (BEFP) data is loaded into the buffer:<br>0 = BEFP complete.<br>1 = BEFP in-progress. |   |  |                     |                      |  |

1. Always clear the Status Register before resuming erase operations after an Erase Suspend command; this prevents ambiguity in Status Register information. For example, 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.
2. BEFP mode is only valid in array.

### 11.2 Read Configuration Register (RCR)

The RCR is a 16-bit read/write register used to select bus-read mode (synchronous or asynchronous), and to configure synchronous burst read characteristics of the device. To modify RCR settings, use the Configure Read Configuration Register command (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)).

RCR contents can be examined using the Read Device Identifier command, and then reading from offset 0x05 (see Section 7.2, "Read Device Identifier" on page 22).

Upon power-up or exit from reset, the RCR defaults to asynchronous mode.

The RCR is shown in Table 13. The following sections describe each RCR bit function.

**Table 13: Read Configuration Register Description**

| Read Configuration Register (RCR) |                                 |               |    |    |   |                    |            |           |          |     |     |            |              |   |   |
|-----------------------------------|---------------------------------|---------------|----|----|---|--------------------|------------|-----------|----------|-----|-----|------------|--------------|---|---|
| Read Mode                         | RES                             | Latency Count |    |    | WAIT Polarity   | Data Output Config | WAIT Delay | Burst Seq | CLK Edge | RES | RES | Burst Wrap | Burst Length |   |   |
| RM                                | R                               | LC[3:0]       |    |    | WP  | DOC                | 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)                    |               |    |    | Set to 0. This bit cannot be altered by customer.   |                    |            |           |          |     |     |            |              |   |   |
| 13:11                             | Latency Count (LC[2:0])         |               |    |    | 000 =Code 0 reserved<br>001 =Code 1 reserved<br>010 =Code 2<br>011 =Code 3<br>100 =Code 4<br>101 =Code 5<br>110 =Code 6<br>111 =Code 7(default) |                    |            |           |          |     |     |            |              |   |   |
| 10                                | WAIT Polarity (WP)              |               |    |    | 0 =WAIT signal is active low<br>1 =WAIT signal is active high (default)   |                    |            |           |          |     |     |            |              |   |   |
| 9                                 | Data Output Configuration (DOC) |               |    |    | 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)                    |               |    |    | Set to 0. This bit cannot be altered by customer.   |                    |            |           |          |     |     |            |              |   |   |
| 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)       |                    |            |           |          |     |     |            |              |   |   |

### 11.2.1 Read Mode (RCR.15)

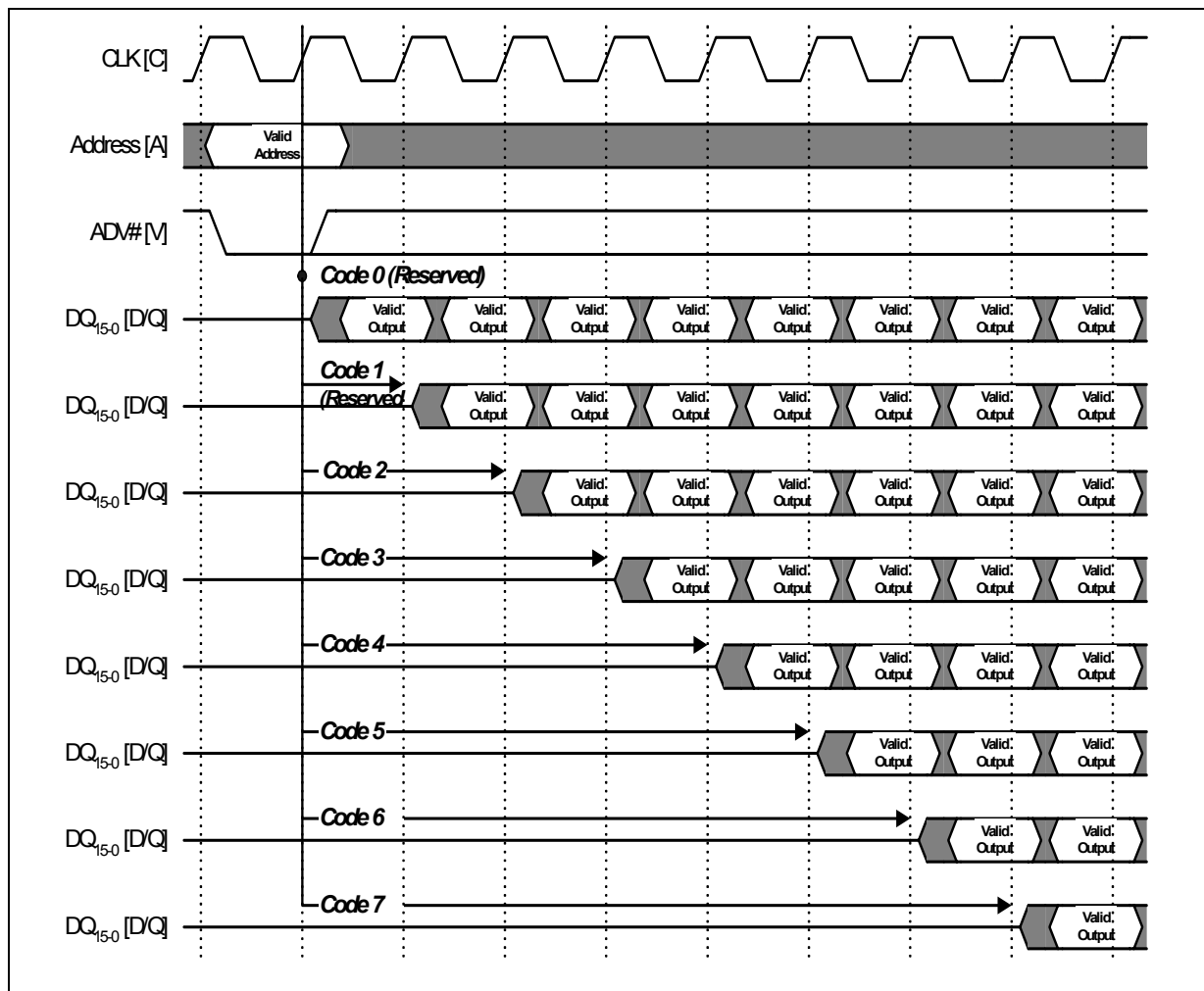
The Read Mode (RM) bit selects synchronous burst-mode or asynchronous page-mode operation for the device. When the RM bit is set, asynchronous page mode is selected (default). When RM is cleared, synchronous burst mode is selected.

### 11.2.2 Latency Count (RCR[13:11])

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 driven onto DQ[15:0]. The input clock frequency is used to determine this value and **Figure 10** shows the data output latency for the different settings of LC. The maximum Latency Count for P30-65nm SBC device would be Code 4 based on the Max clock frequency specification of 52MHz, and there will be zero WAIT States when bursting within the word line. Please also refer to **Section 11.2.3, "End of Word Line (EOWL) Considerations"** on page 37 for more information on EOWL.

Refer to **Table 14, "LC and Frequency Support"** on page 36 for Latency Code Settings.

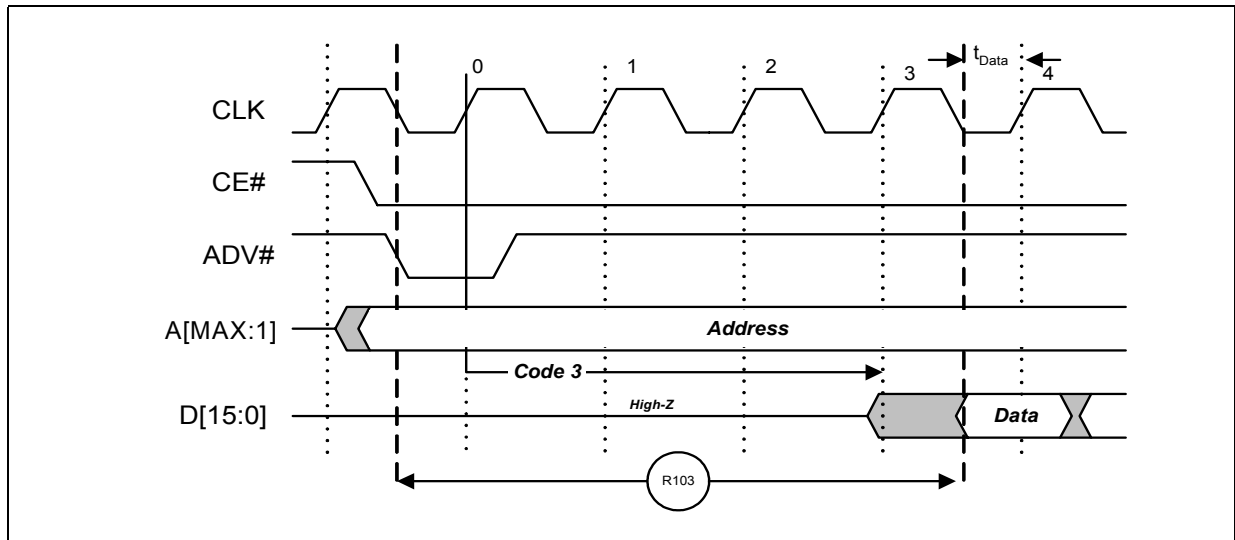
**Figure 10: First-Access Latency Count**



**Table 14: LC and Frequency Support**

| Latency Count Settings | Frequency Support (MHz) |
|------------------------|-------------------------|
| 3                      | ≤ 40                    |
| 4                      | ≤ 52                    |

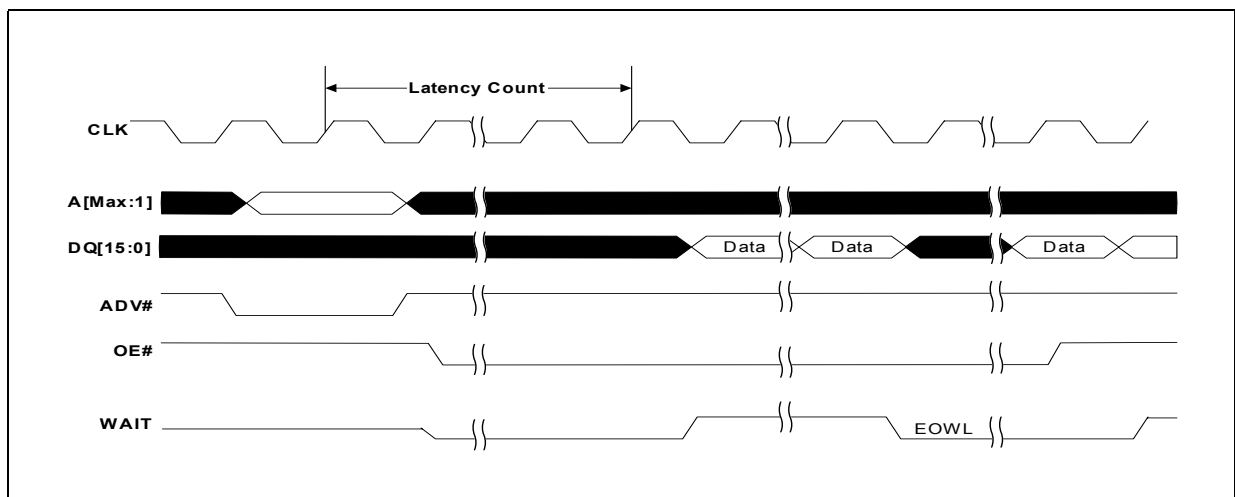
**Figure 11: Example Latency Count Setting Using Code 3**



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

The delay may occur when a burst sequence access crosses a 8-word boundary. That is, A[3:1] of start address does not equal 0x0. Figure 12, "End of Wordline Timing Diagram" on page 37 illustrates the end of wordline WAIT state(s), which occur after the first 8-word boundary is reached. The number of data words and the number of WAIT states is summarized in Table 15, "End of Wordline Data and WAIT state Comparison" on page 38 for both P30-130nm and P30-65nm SBC devices.

**Figure 12: End of Wordline Timing Diagram**



**Table 15: End of Wordline Data and WAIT state Comparison**

| Latency Count | P30-130nm     |               | P30-65nm SBC  |               |
|---------------|---------------|---------------|---------------|---------------|
|               | Data States   | WAIT States   | Data States   | WAIT States   |
| 1             | Not Supported | Not Supported | Not Supported | Not Supported |
| 2             | 4             | 0 to 1        | Not Supported | Not Supported |
| 3             | 4             | 0 to 2        | 8             | 0 to 2        |
| 4             | 4             | 0 to 3        | 8             | 0 to 3        |
| 5             | 4             | 0 to 4        | 8             | 0 to 4        |
| 6             | 4             | 0 to 5        | 8             | 0 to 5        |
| 7             | 4             | 0 to 6        | 8             | 0 to 6        |

### 11.2.4 WAIT Polarity (RCR.10)

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. When WP is cleared, WAIT is asserted low (default). WAIT changes state on valid clock edges during active bus cycles (CE# asserted, OE# asserted, RST# deasserted).

**Table 16: WAIT Functionality Table**

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

**Notes:**

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

### 11.2.5 Data Output Configuration (RCR.9)

The Data Output Configuration (DOC) bit, RCR.9 determines whether a data word remains valid on the data bus for one or two clock cycles. This period of time is called the "data cycle". When DOC is set, output data is held for two clocks (default). When DOC is cleared, output data is held for one clock (see [Figure 13, "Data Hold Timing" on page 39](#)). 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:

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

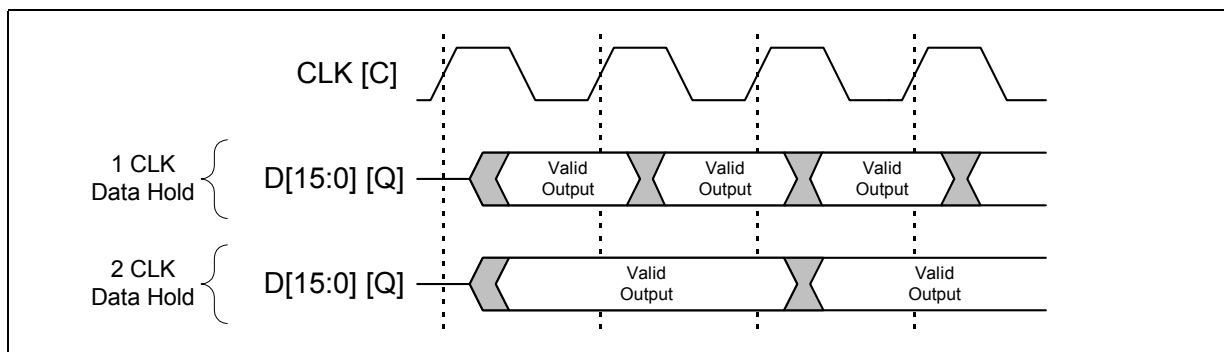
$$t_{DATA} = \text{Data set up to Clock (defined by CPU)}$$

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 13: Data Hold Timing



### 11.2.6 WAIT Delay (RCR.8)

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.

### 11.2.7 Burst Sequence (RCR.7)

The Burst Sequence (BS) bit selects linear-burst sequence (default). Only linear-burst sequence is supported. Table 17 shows the synchronous burst sequence for all burst lengths, as well as the effect of the Burst Wrap (BW) setting.

Table 17: Burst Sequence Word Ordering (Sheet 1 of 2)

| 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-...                  |

**Table 17: Burst Sequence Word Ordering (Sheet 2 of 2)**

|    |   |         |                      |                        |                              |
|----|---|---------|----------------------|------------------------|------------------------------|
| 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-<br>... |
| 15 | 1 |         |                      | 15-16-17-18-19...29-30 | 15-16-17-18-19-20-21-<br>... |

### 11.2.8 Clock Edge (RCR.6)

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.

### 11.2.9 Burst Wrap (RCR.3)

The Burst Wrap (BW) bit determines whether 4, 8, 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.

### 11.2.10 Burst Length (RCR[2:0])

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

Continuous burst accesses are linear only, and do not wrap within any word length boundaries (see [Table 17, "Burst Sequence Word Ordering" on page 39](#)). When a burst cycle begins, the device outputs synchronous burst data until it reaches the end of the "burstable" address space.

## 11.3 One-Time Programmable (OTP) Registers

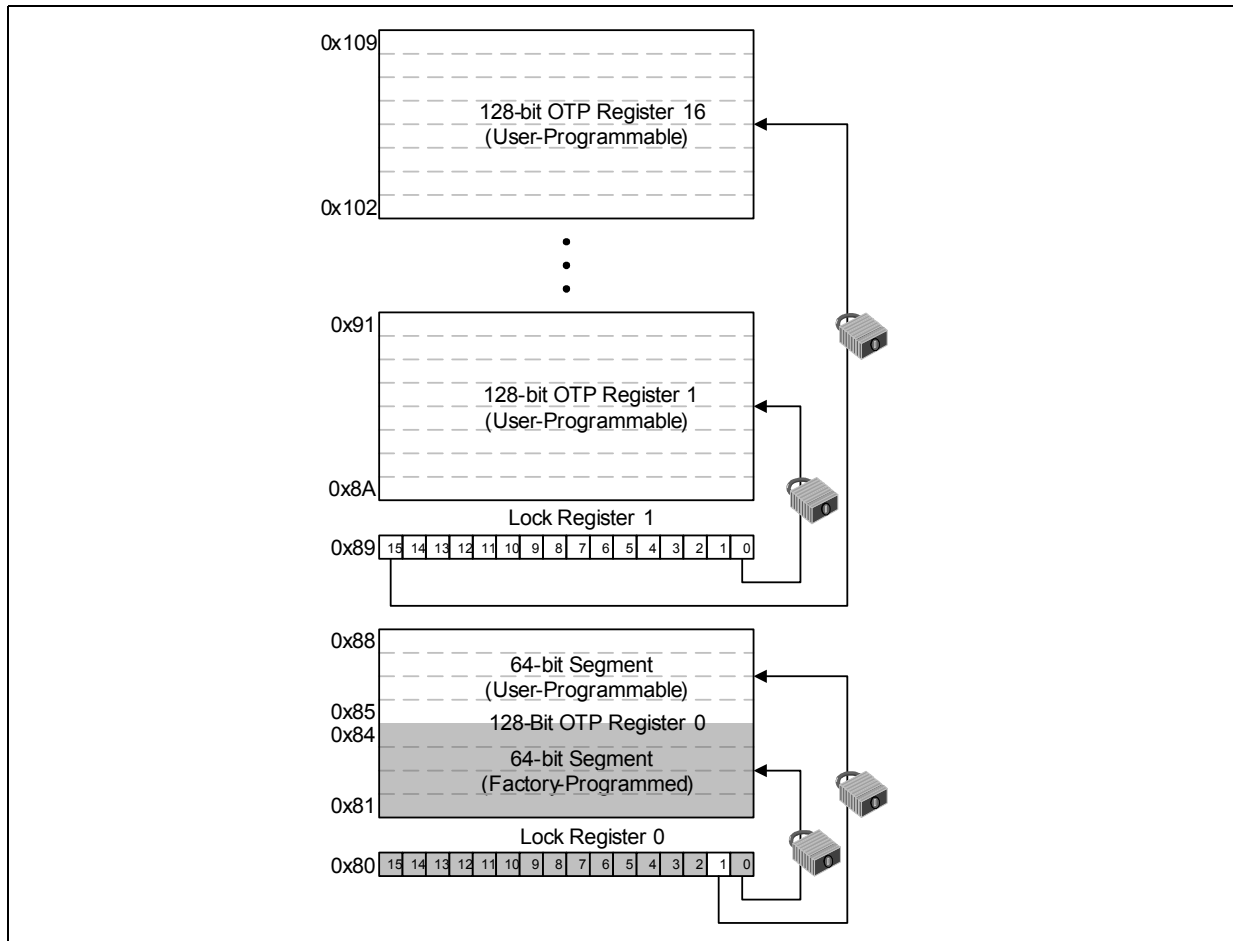
The device contains 17 OTP Registers that can be used to implement system security measures and/or device identification. Each OTP Register can be individually locked.

The first 128-bit OTP 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 upper 64-bit segment, as well as the other sixteen 128-bit OTP Registers, are blank. Users can program these registers as needed. Once programmed, users can then lock the OTP Register(s) to prevent additional bit programming (see [Figure 14, "OTP Register Map" on page 41](#)).

Each OTP Register has an associated Lock Register bit. When a Lock Register bit is programmed, the associated OTP Register can only be read; it can no longer be programmed. Each OTP Register can be accessed multiple times to program individual

bits, as long as the register remains unlocked. Additionally, because the Lock Register bits themselves are OTP, when programmed, Lock Register bits cannot be erased. Therefore, when a OTP Register is locked, it cannot be unlocked.

**Figure 14: OTP Register Map**



### 11.3.1 Reading the OTP Registers

The OTP Registers can be read from OTP-RA address. To read the OTP Register, first issue the Read Device Identifier command at OTP-RA address to place the device in the Read Device Identifier state (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)). Next, perform a read operation using the address offset corresponding to the register to be read. [Table 8, "Device Identifier Information" on page 22](#) shows the address offsets of the OTP Registers and Lock Registers. OTP Registers and Lock Registers data is read 16 bits at a time.

### 11.3.2 Programming the OTP Registers

To program an OTP Register, first issue the OTP Register Program Setup command at the device base address plus the offset address of the desired OTP Register location (OTP-RA: see [Figure 14, "OTP Register Map" on page 41](#)). Next, write the desired OTP Register data to the same OTP Register address. See [Section 6.2, "Device Command Bus Cycles" on page 20](#).

The device programs the 64-bit and Sixteen 128-bit user-programmable OTP Register data 16 bits at a time (see [Figure 40, "OTP Register Programming Flowchart" on page 81](#)). Issuing the OTP Register Program Setup command outside of the OTP Register's address space causes a program error (SR.4 set). Attempting to program a locked OTP Register causes a program error (SR.4 set) and a lock error (SR.1 set).

### 11.3.3 Locking the OTP Registers

Each OTP Register can be locked by programming its respective lock bit in the Lock Register. To lock an OTP Register, program the corresponding bit in the Lock Register by issuing the Lock Register Program Setup command, followed by the desired Lock Register data (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)). 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 8, "Device Identifier Information" on page 22](#)).

Bit 0 of Lock Register 0 is already programmed during the manufacturing process at Numonyx factory, locking the lower half segment of the first 128-bit OTP Register. Bit 1 of Lock Register 0 can be programmed by user to the upper half segment of the first 128-bit OTP Register. When programming Bit 1 of Lock Register 0, all other bits need to be left as '1' such that the data programmed is 0xFFFFD.

Lock Register 1 controls the locking of the upper sixteen 128-bit OTP Registers. Each bit of Lock Register 1 corresponds to a specific 128-bit OTP Register; e.g., programming LR1.0 locks the corresponding OTP Register 1.

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

## 12.0 Power and Reset Specifications

### 12.1 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.

### 12.2 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.

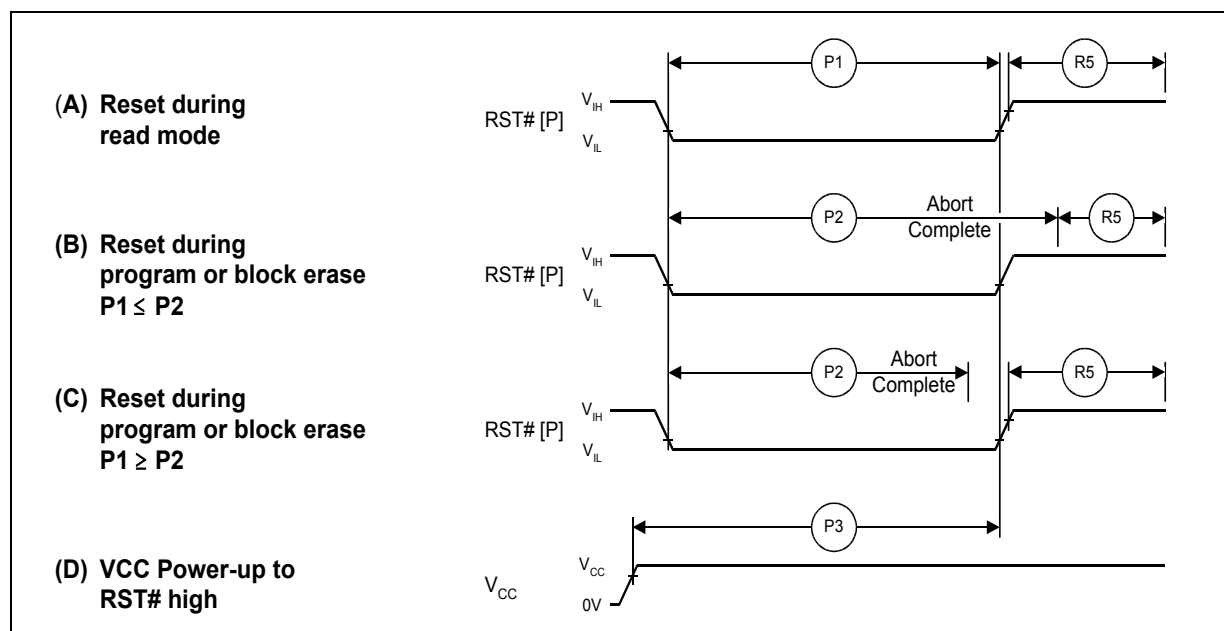
**Table 18: Power and Reset**

| Num | Symbol      | Parameter                                   | Min | Max | Unit    | Notes   |
|-----|-------------|---|-----|-----|---------|---------|
| P1  | $t_{PLPH}$  | RST# pulse width low                        | 100 | -   | ns      | 1,2,3,4 |
| P2  | $t_{PLRH}$  | RST# low to device reset during erase       | -   | 25  | $\mu$ s | 1,3,4,7 |
|     |             | RST# low to device reset during program     | -   | 25  |         | 1,3,4,7 |
| P3  | $t_{VCCPH}$ | VCC Power valid to RST# de-assertion (high) | 60  | -   |         | 1,4,5,6 |

**Notes:**

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

Figure 15: Reset Operation Waveforms



## 12.3 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 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.

## 13.0 Maximum Ratings and Operating Conditions

### 13.1 Absolute Maximum Ratings

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

**Table 19: Absolute Maximum Ratings**

| Parameter  | Maximum Rating  | Notes |
|--|-----------------|-------|
| Temperature under bias                           | -40°C to +85°C  | -     |
| Storage temperature                              | -65°C to +125°C | -     |
| Voltage on any signal (except VCC, VPP and VCCQ) | -0.5V to +4.1V  | 1     |
| VPP voltage                                      | -0.2V to +10.0V | 1,2,3 |
| VCC voltage                                      | -0.2V to +2.5V  | 1     |
| VCCQ voltage                                     | -0.2V to +4.1V  | 1     |
| Output short circuit current                     | 100mA           | 4     |

**Notes:**

1. Voltages shown are specified with respect to  $V_{SS}$ . Minimum DC voltage is -0.5V on input/output signals and -0.2V on VCC, VCCQ, and VPP. During transitions, this level may undershoot to -2.0V for periods less than 20ns. Maximum DC voltage on VCC is  $VCC + 0.5V$ , which, during transitions, may overshoot to  $VCC + 2.0V$  for periods less than 20ns. Maximum DC voltage on input/output signals and VCCQ is  $VCCQ + 0.5V$ , which, during transitions, may overshoot to  $VCCQ + 2.0V$  for periods less than 20ns.
2. Maximum DC voltage on VPP may overshoot to +11.5V for periods less than 20ns.
3. Program/erase voltage is typically 1.7V – 2.0V. 9.0V can be applied for 80 hours maximum total, to any blocks for 1000 cycles maximum. 9.0V 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.

### 13.2 Operating Conditions

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

**Table 20: Operating Conditions**

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

**Notes:**

1.  $T_C$  = Case Temperature.
2. In typical operation  $V_{PP}$  program voltage is  $V_{PPL}$ .

## 14.0 Electrical Specifications

### 14.1 DC Current Characteristics

Table 21: DC Current Characteristics (Sheet 1 of 2)

| Sym  | Parameter   |  | CMOS Inputs<br>(VCCQ = 1.7V - 3.6V) |      | TTL Inputs<br>(VCCQ = 2.4V - 3.6V) |      | Unit | Test Conditions   |   | Notes |       |
|--|---|--|-------------------------------------|------|------------------------------------|------|------|---|---|-------|-------|
|  |   |  | Typ                                 | Max  | Typ                                | Max  |      |   |   |       |       |
| I <sub>LI</sub>  | Input Load Current  |  | -                                   | ±1   | -                                  | ±2   | μA   | VCC = V <sub>CC</sub> Max<br>VCCQ = VCCQ Max<br>V <sub>IN</sub> = VCCQ or V <sub>SS</sub> |   | 1,6   |       |
| I <sub>LO</sub>  | Output Leakage Current  | DQ[15:0], WAIT                               | -                                   | ±1   | -                                  | ±10  | μA   | VCC = VCC Max<br>VCCQ = VCCQ Max<br>V <sub>IN</sub> = VCCQ or V <sub>SS</sub>             |   |       |       |
| I <sub>CCS</sub> ,<br>I <sub>CCD</sub>                         | VCC Standby,<br>Power-Down  |  | 128-Mbit                            | 30   | 55                                 | 710  | 2000 | μA  | VCC = VCC Max<br>VCCQ = VCC Max<br>CE# = VCCQ<br>RST# = VCCQ (for I <sub>CCS</sub> )<br>RST# = V <sub>SS</sub> (for I <sub>CCD</sub> )<br>WP# = V <sub>IH</sub> |       | 1,2   |
|  |   | 64-Mbit                                      | 30                                  | 55   | 710                                | 2000 |      |   |   |       |       |
| I <sub>CCR</sub>   | Average VCC Read Current  | Asynchronous Single-Word<br>f = 5MHz (1 CLK) | 20                                  | 25   | -                                  | -    | mA   | 8-Word Read   | VCC = VCCMax<br>CE# = V <sub>IL</sub><br>OE# = V <sub>IH</sub><br>Inputs: V <sub>IL</sub> or V <sub>IH</sub>  |       | 1     |
|  |   | Page-Mode Read<br>f = 13MHz (17 CLK)         | 12                                  | 16   | -                                  | -    | mA   | 8-Word Read   |   |       |       |
|  |   | Synchronous Burst<br>f = 52MHz, LC=4         | 16                                  | 19   | -                                  | -    | mA   | 4-Word Read   |   |       |       |
|  |   |  | 19                                  | 22   | -                                  | -    | mA   | 8-Word Read   |   |       |       |
|  |   |  | 22                                  | 26   | -                                  | -    | mA   | 16-Word Read  |   |       |       |
|  |   |  | 23                                  | 28   | -                                  | -    | mA   | Continuous Read   |   |       |       |
| I <sub>CCW</sub> ,<br>I <sub>CC E</sub>                        | VCC Program Current,<br>VCC Erase Current   |  | 35                                  | 50   | 35                                 | 50   | mA   | VPP = V <sub>PP L</sub> , Pgm/Ers in progress   | 1,3,5   |       |       |
|  |   |  | 26                                  | 33   | 26                                 | 33   |      | VPP = V <sub>PP H</sub> , Pgm/Ers in progress   | 1,3,5   |       |       |
| I <sub>CCWS</sub> ,<br>I <sub>CC ES</sub>                      | VCC Program Suspend Current,<br>VCC Erase Suspend Current                         |  | 128-Mbit                            | 30   | 55                                 | 710  | 2000 | μA  | CE# = VCCQ; suspend in progress   |       | 1,3,4 |
|  |   |  | 64-Mbit                             | 30   | 55                                 | 710  | 2000 |   |   |       |       |
| I <sub>PPS</sub> ,<br>I <sub>PPWS</sub> ,<br>I <sub>PPES</sub> | VPP Standby Current,<br>VPP Program Suspend Current,<br>VPP Erase Suspend Current |  | 0.2                                 | 5    | 0.2                                | 5    | μA   | VPP = V <sub>PP L</sub> , suspend in progress   |   | 1,3,7 |       |
| I <sub>PPR</sub>   | VPP Read  |  | 2                                   | 15   | 2                                  | 15   | μA   | VPP = V <sub>PP L</sub>   |   | 1,3   |       |
| I <sub>PPW</sub>   | VPP Program Current   |  | 0.05                                | 0.10 | 0.05                               | 0.10 | mA   | VPP = V <sub>PP L</sub> , program in progress   |   | 3     |       |
|  |   |  | 5                                   | 10   | 5                                  | 10   |      | VPP = V <sub>PP H</sub> , program in progress   |   |       |       |
| I <sub>PP E</sub>  | V <sub>PP</sub> Erase Current   |  | 0.05                                | 0.10 | 0.05                               | 0.10 | mA   | VPP = V <sub>PP L</sub> , erase in progress   |   | 3     |       |
|  |   |  | 5                                   | 10   | 5                                  | 10   |      | VPP = V <sub>PP H</sub> , erase in progress   |   |       |       |

**Table 21: DC Current Characteristics (Sheet 2 of 2)**

| Sym               | Parameter       | CMOS Inputs<br>(VCCQ = 1.7V - 3.6V) |      | TTL Inputs<br>(VCCQ = 2.4V - 3.6V) |      | Unit | Test Conditions                            | Notes |
|-------------------|-----------------|-------------------------------------|------|------------------------------------|------|------|--|-------|
|                   |                 | Typ                                 | Max  | Typ                                | Max  |      |  |       |
| I <sub>PPBC</sub> | VPP Blank Check | 0.05                                | 0.10 | 0.05                               | 0.10 | mA   | VPP = V <sub>PPL</sub> , erase in progress | 3     |
|                   |                 | 5                                   | 10   | 5                                  | 10   |      | VPP = V <sub>PPH</sub> , erase in progress |       |

**Notes:**

- All currents are RMS unless noted. Typical values at typical VCC, T<sub>C</sub> = +25°C.
- I<sub>CCS</sub> is the average current measured over any 5ms time interval 5μs after CE# is deasserted.
- Sampled, not 100% tested.
- I<sub>CCES</sub> is specified with the device deselected. If device is read while in erase suspend, current is I<sub>CCES</sub> plus I<sub>CCR</sub>.
- I<sub>CCW</sub>, I<sub>CCE</sub> measured over typical or max times specified in [Section 15.5, "Program and Erase Characteristics" on page 58](#).
- if V<sub>IN</sub> > VCC the input load current increases to 10μA max.
- the I<sub>PPS</sub>, I<sub>PPWS</sub>, I<sub>PPES</sub> Will increase to 200μA when VPP/WP# is at V<sub>PPH</sub>.

## 14.2 DC Voltage Characteristics

**Table 22: DC Voltage Characteristics**

| Sym               | Parameter            | CMOS Inputs<br>(VCCQ = 1.7V - 3.6V) |            | TTL Inputs <sup>(1)</sup><br>(VCCQ = 2.4V - 3.6V) |            | Unit | Test Conditions  | Notes |
|-------------------|----------------------|-------------------------------------|------------|---|------------|------|--|-------|
|                   |                      | Min                                 | Max        | Min   | Max        |      |  |       |
| V <sub>IL</sub>   | Input Low Voltage    | -0.5                                | 0.4        | -0.5  | 0.6        | V    | -  | 2     |
| V <sub>IH</sub>   | Input High Voltage   | VCCQ - 0.4                          | VCCQ + 0.5 | 2.0   | VCCQ + 0.5 | V    | -  |       |
| V <sub>OL</sub>   | Output Low Voltage   | -                                   | 0.2        | -   | 0.2        | V    | VCC = VCC Min<br>VCCQ = VCCQ Min<br>I <sub>OL</sub> = 100μA  | -     |
| V <sub>OH</sub>   | Output High Voltage  | VCCQ - 0.2                          | -          | VCCQ - 0.2  | -          | V    | VCC = VCC Min<br>VCCQ = VCCQ Min<br>I <sub>OH</sub> = -100μA | -     |
| V <sub>PPLK</sub> | VPP Lock-Out Voltage | -                                   | 0.4        | -   | 0.4        | V    | -  | 3     |
| V <sub>LKO</sub>  | VCC Lock Voltage     | 1.0                                 | -          | 1.0   | -          | V    | -  | -     |
| V <sub>LKOQ</sub> | VCCQ Lock Voltage    | 0.9                                 | -          | 0.9   | -          | V    | -  | -     |

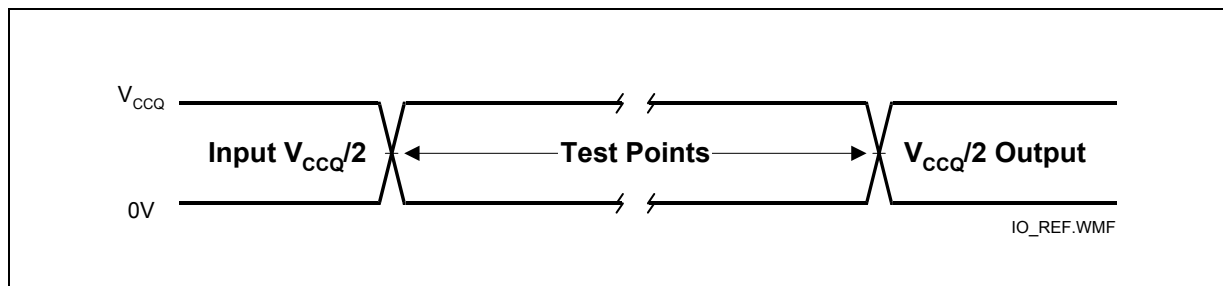
**Notes:**

- Synchronous read mode is not supported with TTL inputs.
- V<sub>IL</sub> can undershoot to -0.4V and V<sub>IH</sub> can overshoot to VCCQ + 0.4V for durations of 20ns or less.
- VPP ≤ V<sub>PPLK</sub> inhibits erase and program operations. Do not use V<sub>PPL</sub> and V<sub>PPH</sub> outside their valid ranges.

## 15.0 AC Characteristics

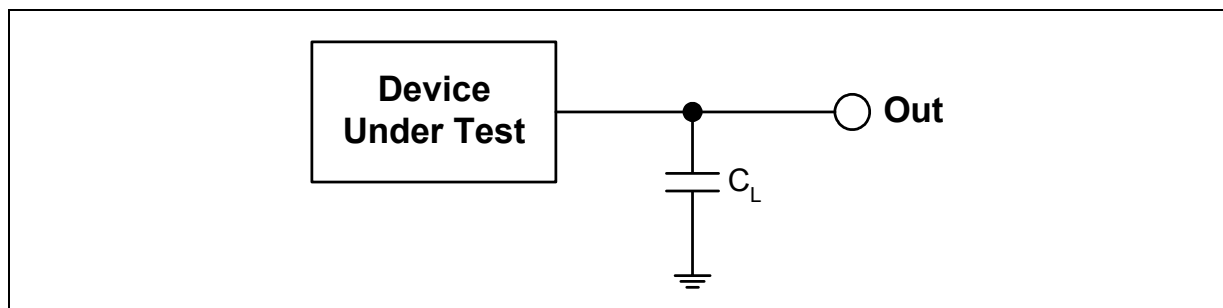
### 15.1 AC Test Conditions

**Figure 16: AC Input/Output Reference Waveform**



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

**Figure 17: Transient Equivalent Testing Load Circuit**



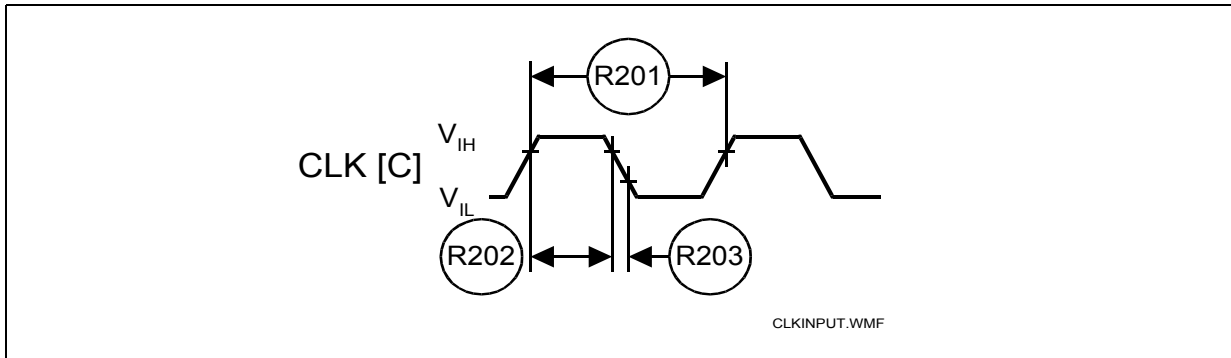
**Notes:**

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 23: Test Configuration Component Value for Worst Case Speed Conditions**

| Test Configuration     | $C_L$ (pF) |
|------------------------|------------|
| VCCQ Min Standard Test | 30         |

Figure 18: Clock Input AC Waveform



## 15.2 Capacitance

Table 24: Capacitance

| Symbol    | Parameter          | Signals                                      | Min | Typ | Max | Unit | Condition  | Notes |
|-----------|--------------------|--|-----|-----|-----|------|--|-------|
| $C_{IN}$  | Input Capacitance  | Address, CE#, WE#, OE#, RST#, CLK, ADV#, WP# | 3   | 7   | 8   | pF   | Typ temp = 25°C,<br>Max temp = 85°C,<br>VCC = (0V - 2.0V),<br>VCCQ = (0V - 3.6V) | 1     |
| $C_{OUT}$ | Output Capacitance | Data, WAIT                                   | 3   | 5   | 6   | pF   |  |       |

**Notes:**

1. Sampled, not 100% tested.

## 15.3 AC Read Specifications

Table 25: AC Read Specifications (Sheet 1 of 2)

| Num                                | Symbol          | Parameter  | Min            | Max  | Unit | Notes |         |
|------------------------------------|-----------------|--|----------------|------|------|-------|---------|
| <b>Asynchronous Specifications</b> |                 |  |                |      |      |       |         |
| R1                                 | $t_{AVAV}$      | Read cycle time  | Easy BGA/QUAD+ | 65   | -    | ns    | -       |
|                                    |                 |  | TSOP           | 75   | -    | ns    | -       |
| R2                                 | $t_{AVQV}$      | Address to output valid                                      | Easy BGA/QUAD+ | -    | 65   | ns    | -       |
|                                    |                 |  | TSOP           | -    | 75   | ns    | -       |
| R3                                 | $t_{ELQV}$      | CE# low to output valid                                      | Easy BGA/QUAD+ | -    | 65   | ns    | -       |
|                                    |                 |  | TSOP           | -    | 75   | ns    | -       |
| 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                                 | -              | 20   | ns   | 1,3   |         |
| R9                                 | $t_{GHQZ}$      | OE# high to output in high-Z                                 | -              | 15   | ns   |       |         |
| R10                                | $t_{OH}$        | Output hold from first occurring address, CE#, or OE# change | 0              | -    | ns   |       |         |
| R11                                | $t_{EHEL}$      | CE# pulse width high   | 17             | -    | 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                                     | Easy BGA/QUAD+ | -    | 65   |       | ns      |
|                                    |                 |  | TSOP           | -    | 75   |       | ns      |
| 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  | Easy BGA/QUAD+ | -    | 52   | MHz   | 1,3,5,6 |
|                                    |                 |  | TSOP           | -    | 40   | MHz   |         |
| R201                               | $t_{CLK}$       | CLK period   | Easy BGA/QUAD+ | 19.2 | -    | ns    |         |
|                                    |                 |  | TSOP           | 25   | -    | ns    |         |
| R202                               | $t_{CH/CL}$     | CLK high/low time  | Easy BGA/QUAD+ | 5    | -    | ns    |         |
|                                    |                 |  | TSOP           | 9    | -    | ns    |         |
| R203                               | $t_{FCLK/RCLK}$ | CLK fall/rise time   | 0.3            | 3    | ns   |       |         |

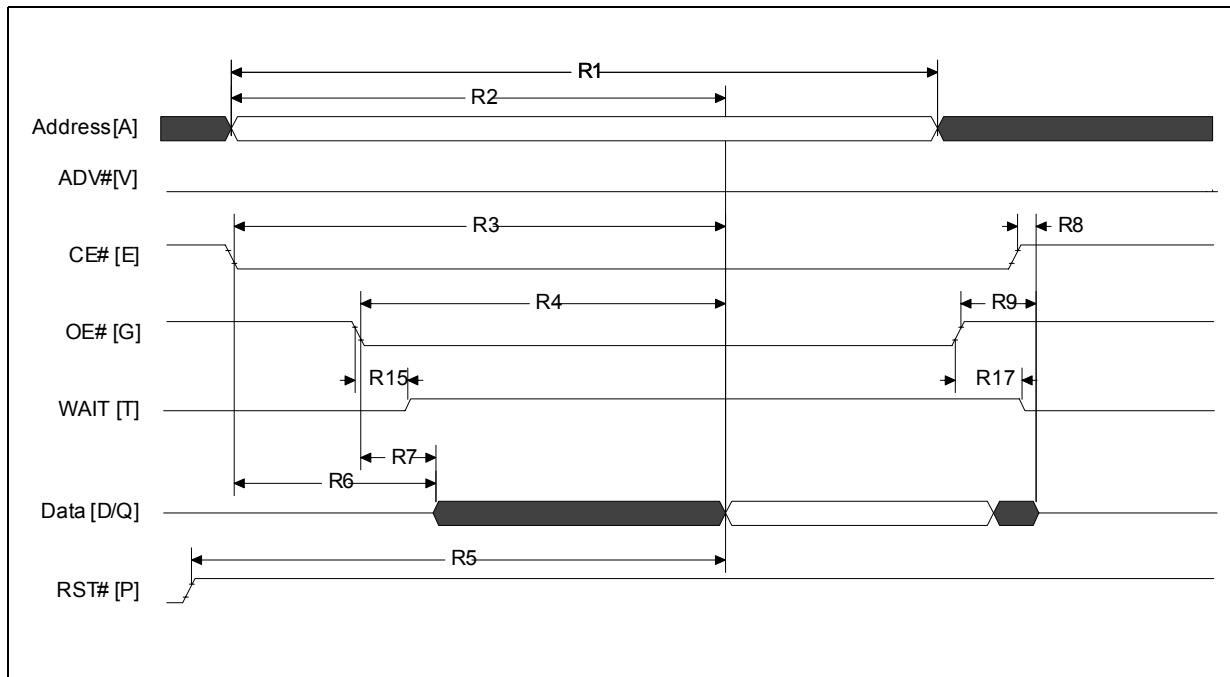
**Table 25: AC Read Specifications (Sheet 2 of 2)**

| Num   | Symbol                | Parameter               | Min            | Max | Unit | Notes |     |
|---|-----------------------|-------------------------|----------------|-----|------|-------|-----|
| <b>Synchronous Specifications<sup>(5)</sup></b> |                       |                         |                |     |      |       |     |
| R301  | $t_{AVCH/L}$          | Address setup to CLK    | 9              | -   | ns   | 1,6   |     |
| 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     | Easy BGA/QUAD+ | -   | 17   | ns    | 1,6 |
|   |                       |                         | TSOP           | -   | 20   | ns    |     |
| R305  | $t_{CHQX}$            | Output hold from CLK    | Easy BGA/QUAD+ | 3   | -    | ns    | 1,6 |
|   |                       |                         | TSOP           | 5   | -    | ns    | 1,6 |
| R306  | $t_{CHAX}$            | Address hold from CLK   | 10             | -   | ns   | 1,4,6 |     |
| R307  | $t_{CHTV}$            | CLK to WAIT valid       | Easy BGA/QUAD+ | -   | 17   | ns    | 1,6 |
|   |                       |                         | TSOP           | -   | 20   | ns    | 1,6 |
| R311  | $t_{CHVL}$            | CLK Valid to ADV# Setup | 3              | -   | ns   | 1     |     |
| R312  | $t_{CHTX}$            | WAIT Hold from CLK      | Easy BGA/QUAD+ | 3   | -    | ns    | 1,6 |
|   |                       |                         | TSOP           | 5   | -    | ns    | 1,6 |

**Notes:**

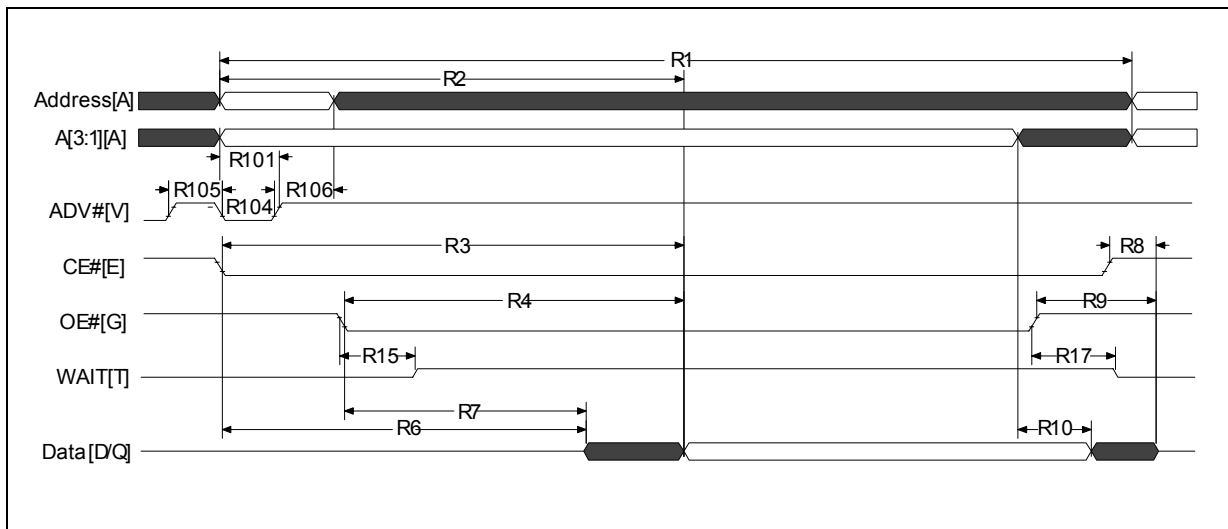
1. See Figure 16, "AC Input/Output Reference Waveform" on page 48 for timing measurements and max allowable input slew rate.
2. OE# may be delayed by up to  $t_{ELQV} - t_{GLQV}$  after CE#'s falling edge without impact to  $t_{ELQV}$ .
3. Sampled, not 100% tested.
4. Address hold in synchronous burst read mode is  $t_{CHAX}$  or  $t_{VHAX}$ , whichever timing specification is satisfied first.
5. Synchronous burst read mode is not supported with TTL level inputs.
6. Applies only to subsequent synchronous reads.

**Figure 19: Asynchronous Single-Word Read (ADV# Low)**



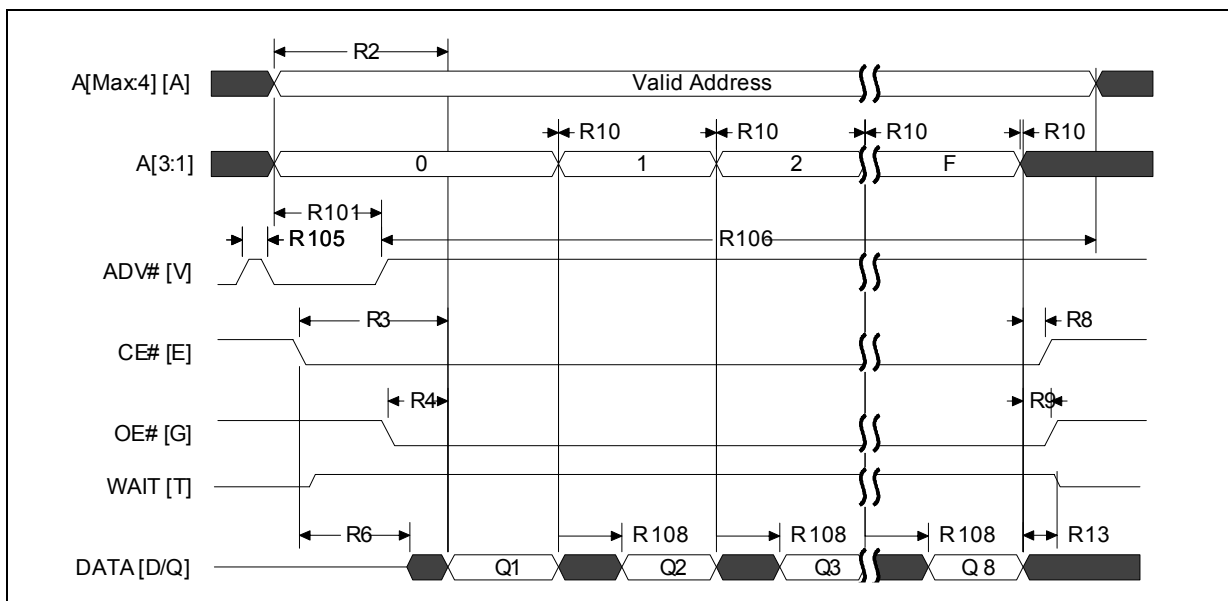
**Note:** WAIT shown deasserted during asynchronous read mode (RCR.10=0, WAIT asserted low).

**Figure 20: Asynchronous Single-Word Read (ADV# Latch)**



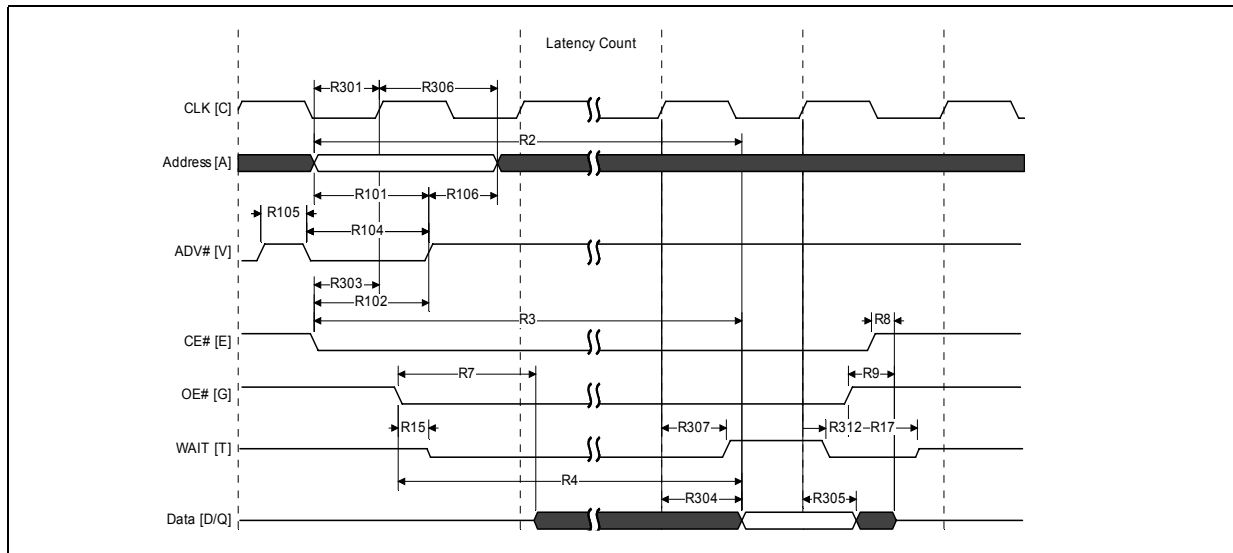
**Note:** WAIT shown deasserted during asynchronous read mode (RCR.10=0, WAIT asserted low).

**Figure 21: Asynchronous Page-Mode Read Timing**



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

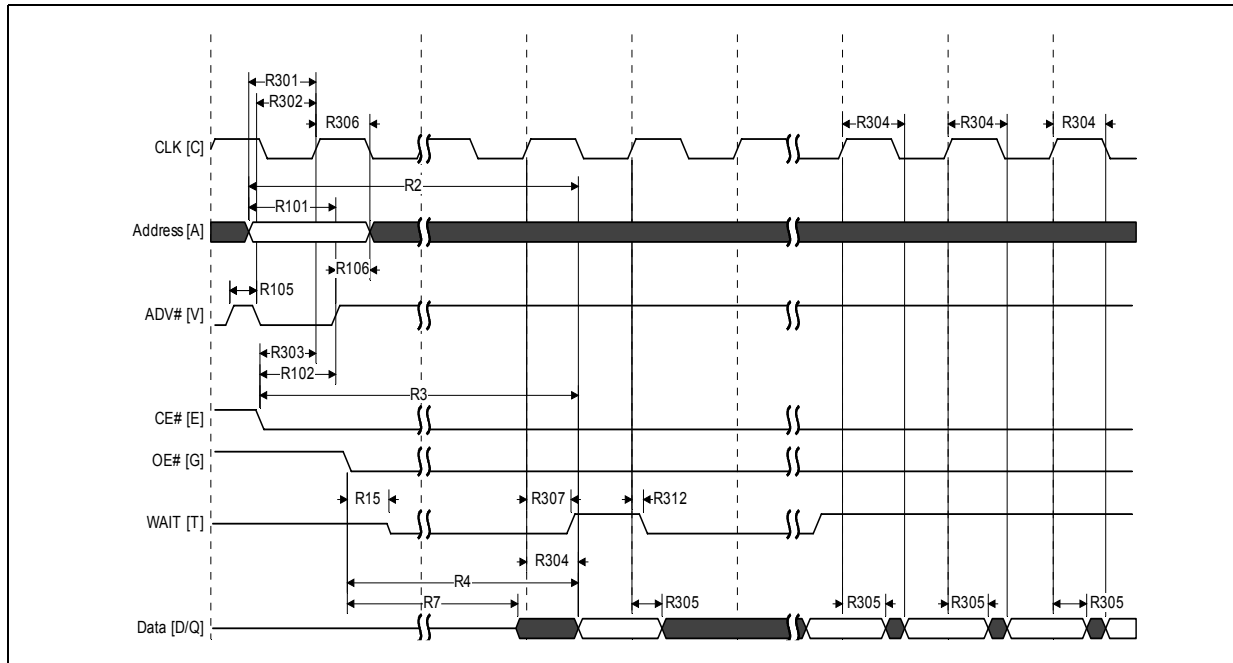
**Figure 22: Synchronous Single-Word Array or Non-array Read Timing**



**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. 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.

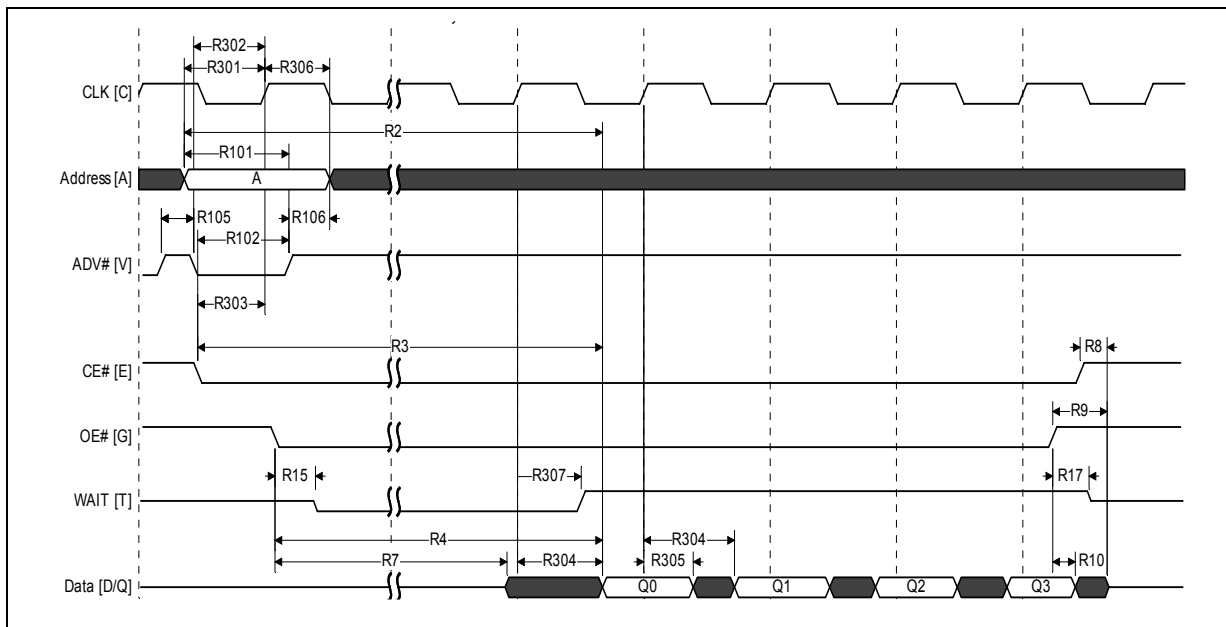
**Figure 23: Continuous Burst Read, showing an Output Delay Timing**



**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. See [Section 11.2.3, "End of Word Line \(EOWL\) Considerations" on page 37](#) for more information.

**Figure 24: 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).

## 15.4 AC Write Specifications

**Table 26: AC Write Specifications (Sheet 1 of 2)**

| 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  | $t_{DVWH}$ | Data setup to WE# high        | 50              | -   | ns   | 1,2,12     |
| W5  | $t_{AVWH}$ | Address setup to WE# high     | 50              | -   | ns   | 1,2        |
| 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                                       | $t_{VPWH}$ | VPP setup to WE# high         | 200             | -   | ns   | 1,2,3,7    |
| W11                                       | $t_{QVVL}$ | VPP hold from Status read     | 0               | -   | ns   |            |
| W12                                       | $t_{QVBL}$ | WP# hold from Status read     | 0               | -   | ns   | 1,2,3,7    |
| W13                                       | $t_{BHWH}$ | 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  |

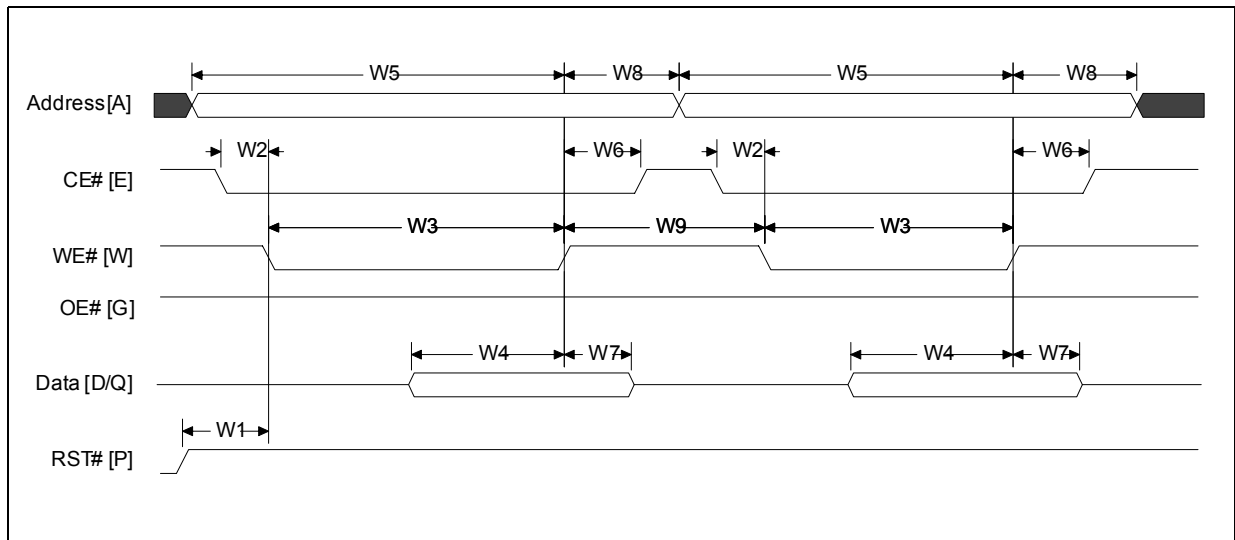
**Table 26: AC Write Specifications (Sheet 2 of 2)**

| Num                                      | Symbol       | Parameter               | Min | Max | Unit | Notes      |
|--|--------------|-------------------------|-----|-----|------|------------|
| 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    | -   | 27  | ns   | 1,2,3,11   |
| W22                                      | $t_{CHWL}$   | Clock high to WE# low   | -   | 27  | ns   |            |

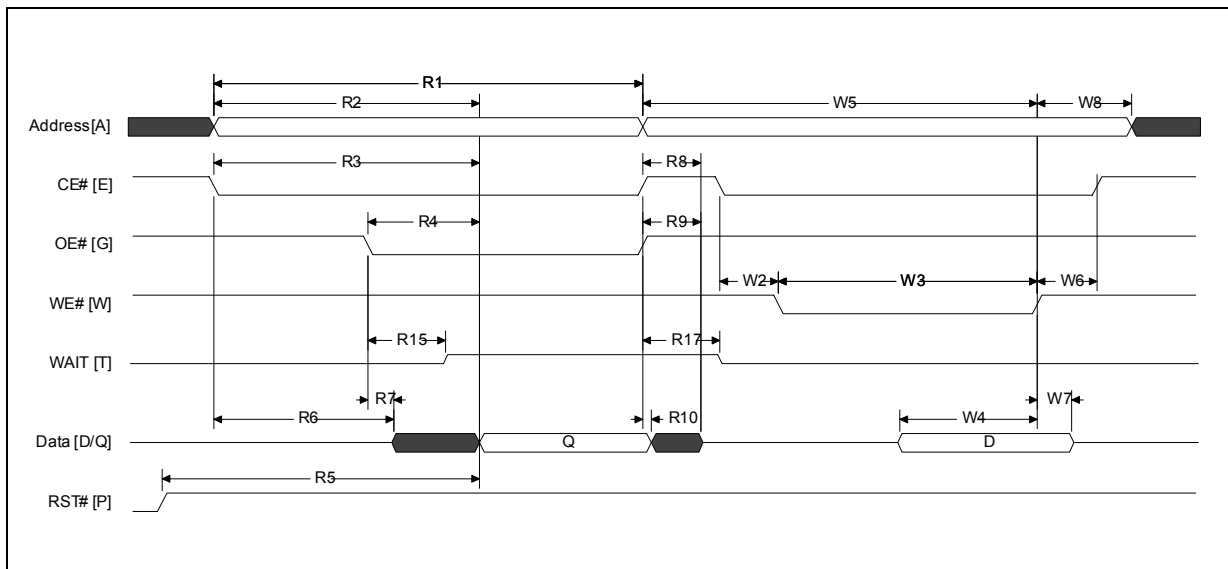
**Notes:**

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

**Figure 25: Write-to-Write Timing**



**Figure 26: Asynchronous Read-to-Write Timing**



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

**Figure 27: Write-to-Asynchronous Read Timing**

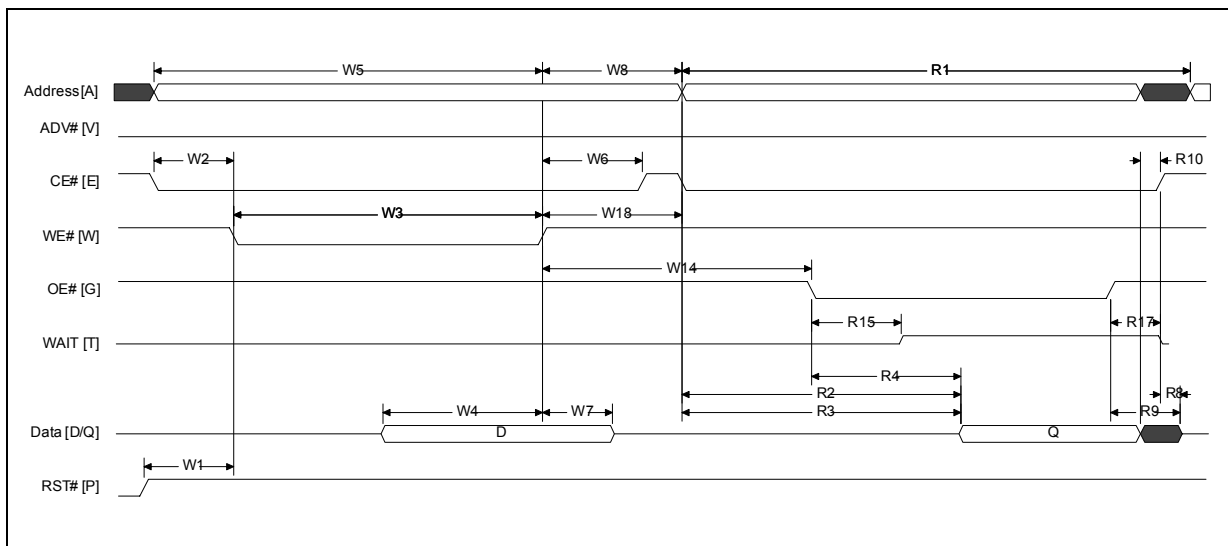
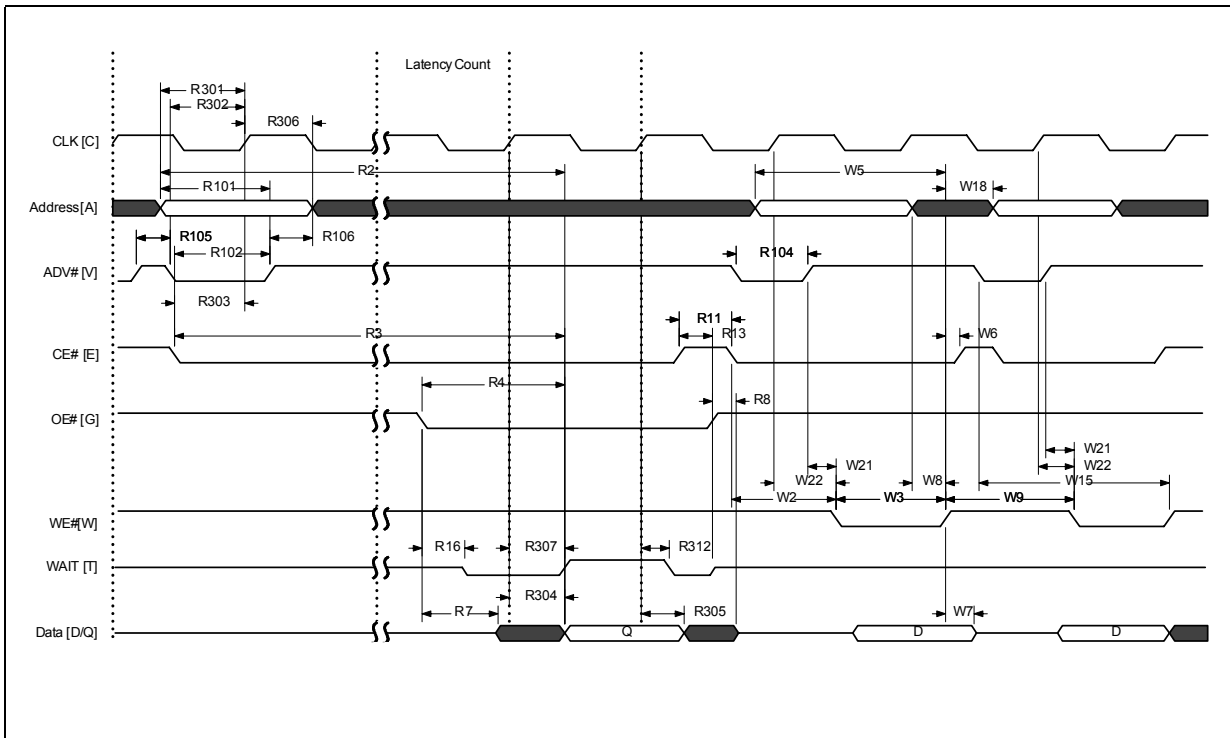
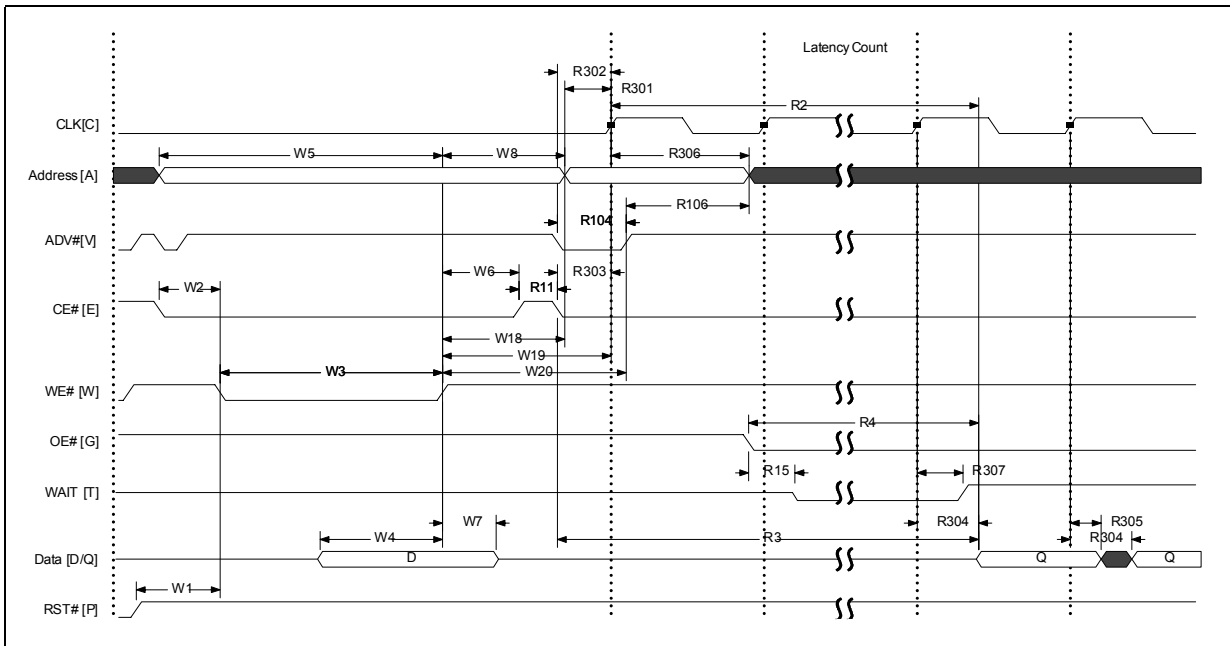


Figure 28: Synchronous Read-to-Write Timing



**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 29: Write-to-Synchronous Read Timing



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

## 15.5 Program and Erase Characteristics

**Table 27: Program and Erase Specifications**

| Num                                   | Symbol                   | Parameter       | V <sub>PPL</sub>                 |     |     | V <sub>PPH</sub> |     |      | Unit | Notes |     |
|---------------------------------------|--------------------------|-----------------|----------------------------------|-----|-----|------------------|-----|------|------|-------|-----|
|                                       |                          |                 | Min                              | Typ | Max | Min              | Typ | Max  |      |       |     |
| Conventional Word Programming         |                          |                 |                                  |     |     |                  |     |      |      |       |     |
| W200                                  | t <sub>PROG/W</sub>      | Program Time    | Single word                      | -   | 40  | 175              | -   | 40   | 175  | μs    | 1   |
| Buffered Programming                  |                          |                 |                                  |     |     |                  |     |      |      |       |     |
| W250                                  | t <sub>PROG/Buffer</sub> | Program Time    | Aligned 16-Wd, BP time (32 Byte) | -   | 70  | 200              | -   | 70   | 200  | μs    | 1   |
|                                       |                          |                 | Aligned 32-Wd, BP time (64 Byte) | -   | 85  | 200              | -   | 85   | 200  |       |     |
|                                       |                          |                 | one full buffer (256 Words)      | -   | 284 | 1280             | -   | 160  | 800  |       |     |
| Buffered Enhanced Factory Programming |                          |                 |                                  |     |     |                  |     |      |      |       |     |
| W451                                  | t <sub>BEFP/B</sub>      | Program         | Single byte                      | n/a | n/a | n/a              | -   | 0.31 | -    | μs    | 1,2 |
| W452                                  | t <sub>BEFP/Setup</sub>  |                 | BEFP Setup                       | n/a | n/a | n/a              | 10  | -    | -    |       | 1   |
| Erase and Suspend                     |                          |                 |                                  |     |     |                  |     |      |      |       |     |
| W501                                  | t <sub>ERS/B</sub>       | Erase Time      | 128-KByte Array Block            | -   | 0.5 | 4                | -   | 0.5  | 4    | s     | 1   |
|                                       |                          |                 | 32-KByte Parameter Block         | -   | 0.4 | 2.5              | -   | 0.4  | 2.5  |       |     |
| W600                                  | t <sub>SUSP/P</sub>      | Suspend Latency | Program suspend                  | -   | 20  | 25               | -   | 20   | 25   | μs    | 1,3 |
| W601                                  | t <sub>SUSP/E</sub>      |                 | Erase suspend                    | -   | 20  | 25               | -   | 20   | 25   |       |     |
| W602                                  | t <sub>ERS/SUSP</sub>    |                 | Erase to Suspend                 | -   | 500 | -                | -   | 500  | -    |       |     |
| Blank Check                           |                          |                 |                                  |     |     |                  |     |      |      |       |     |
| W702                                  | t <sub>BC/AB</sub>       | blank check     | Array Block                      | -   | 3.2 | -                | -   | 3.2  | -    | ms    |     |

**Notes:**

- 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.
- Averaged over entire device.
- 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.

## 16.0 Ordering Information

Figure 30: Decoder for discrete Products

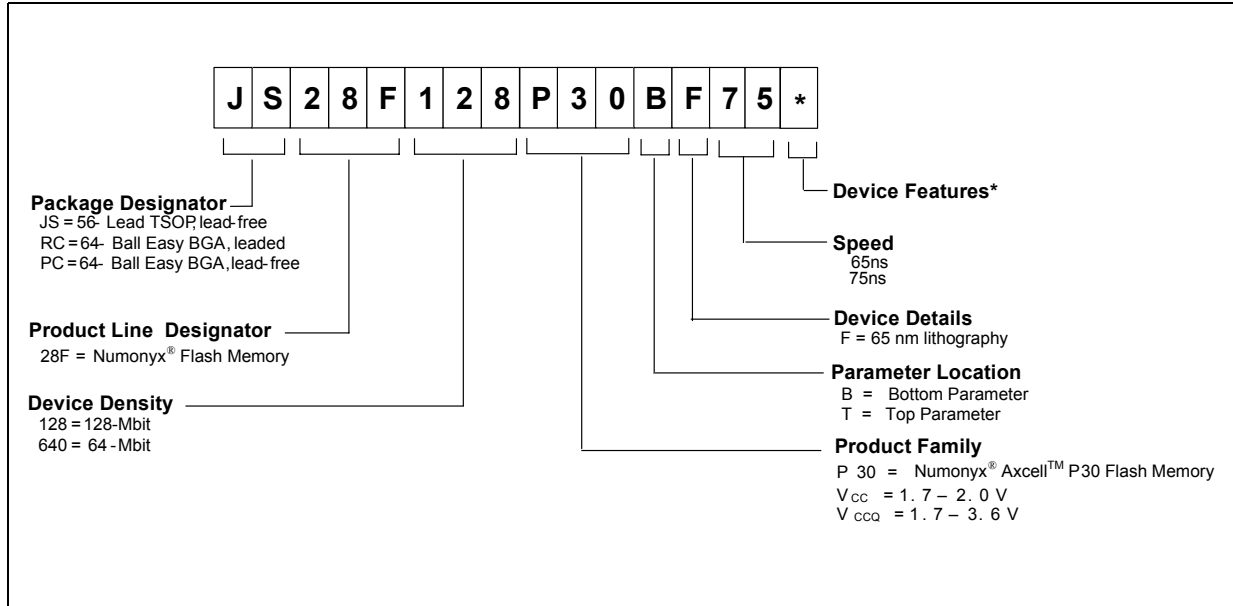


Table 28: Valid Combinations for Discrete Products

| 64-Mbit          | 128-Mbit         |
|------------------|------------------|
| JS28F640P30BF75* | JS28F128P30BF75* |
| JS28F640P30TF75* | JS28F128P30TF75* |
| PC28F640P30BF65* | PC28F128P30BF65* |
| PC28F640P30TF65* | PC28F128P30TF65* |
| RC28F640P30BF65* | RC28F128P30BF65* |
| RC28F640P30TF65* | RC28F128P30TF65* |

Note: The last digit is randomly assigned to cover packing media and/or features or other specific configuration. For further information on ordering products or for product part numbers, go to:  
<http://www.numonyx.com/en-US/MemoryProducts/Pages/PartNumberLookup.aspx>.

Figure 31: Decoder for SCSP package

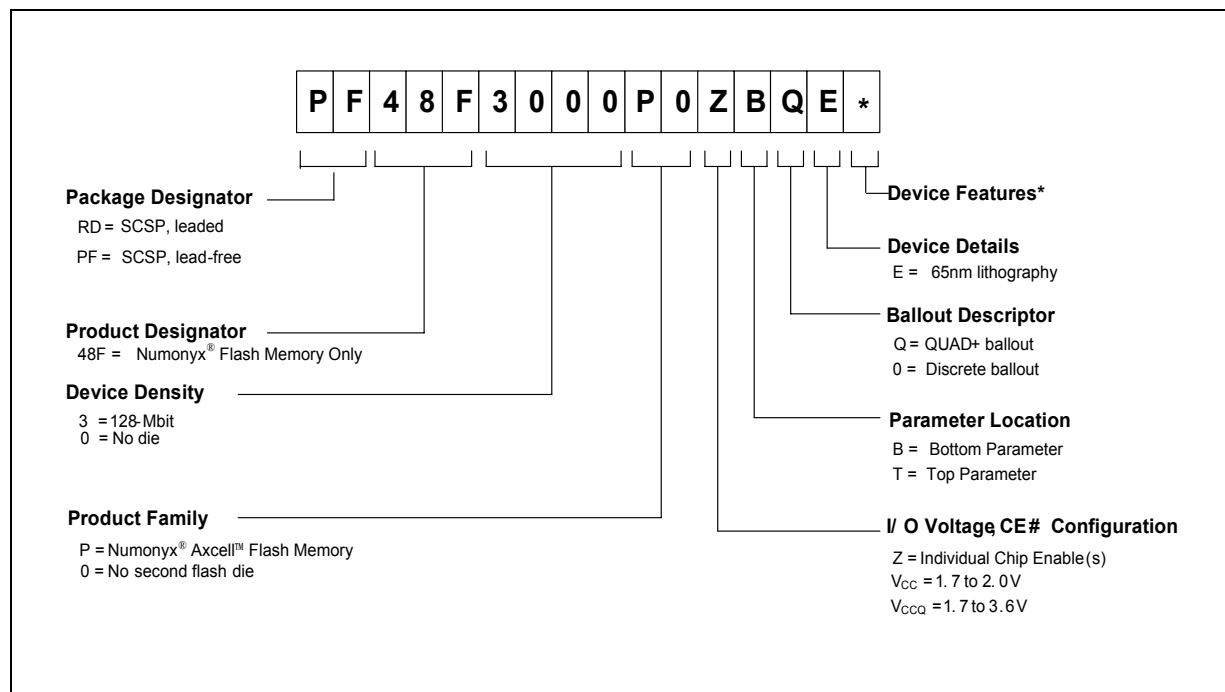


Table 29: Valid Combinations for QUAD+ Package Products

| 128-Mbit         |
|------------------|
| PF48F3000P0ZBQE* |
| PF48F3000P0ZTQE* |
| RD48F3000P0ZBQE* |
| RD48F3000P0ZTQE* |

Note: The last digit is randomly assigned to cover packing media and/or features or other specific configuration. For further information on ordering products or for product part numbers, go to: <http://www.numonyx.com/en-US/MemoryProducts/Pages/PartNumberLookup.aspx>.

## Appendix A Supplemental Reference Information

### A.1 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 Read CFI command (see [Section 6.2, "Device Command Bus Cycles" on page 20](#)). 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.

#### A.1.1 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 have 00h on the upper byte in this mode.

**Table 30: 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 31: Example of Query Structure Output of x16 Devices**

| Offset                             | Hex Code                            | Value           |
|------------------------------------|-------------------------------------|-----------------|
| <b>A<sub>X</sub>-A<sub>1</sub></b> | <b>D<sub>15</sub>-D<sub>0</sub></b> |                 |
| 00010h                             | 0051                                | "Q"             |
| 00011h                             | 0052                                | "R"             |
| 00012h                             | 0059                                | "Y"             |
| 00013h                             | P_ID <sub>LO</sub>                  | PrVendor ID#    |
| 00014h                             | P_ID <sub>HI</sub>                  |                 |
| 00015h                             | P <sub>LO</sub>                     | PrVendor TblAdr |
| 00016h                             | P <sub>HI</sub>                     |                 |
| 00017h                             | A_ID <sub>LO</sub>                  | AltVendor ID#   |
| 00018h                             | A_ID <sub>HI</sub>                  |                 |
| ...                                | ...                                 | ...             |

### A.1.2 Query Structure Overview

The Query command causes the flash component to display the Common Flash Interface (CFI) Query structure or *database*. Table 32 summarizes the structure sub-sections and address locations.

**Table 32: Query Structure**

|                  |   |  |
|------------------|---|--|
| 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 Numonyx-specific Extended Query | Vendor-defined additional information specific to the Primary Vendor Algorithm |

**Note:**

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.

### A.1.3 Read CFI 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 33: CFI Identification**

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

Table 34: System Interface Information

| Offset | Length | Description   | Add | Hex Code | Value        |
|--------|--------|---|-----|----------|--------------|
| 1Bh    | 1      | VCC 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      | VCC 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      | VPP [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      | VPP [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 = $2^n$ $\mu$ -sec                               | 1F: | --06     | 64 $\mu$ s   |
| 20h    | 1      | "n" such that typical full buffer write time-out = $2^n$ $\mu$ -sec                                 | 20: | --09     | 512 $\mu$ s  |
| 21h    | 1      | "n" such that typical block erase time-out = $2^n$ m-sec  | 21: | --09     | 0.5s         |
| 22h    | 1      | "n" such that typical full chip erase time-out = $2^n$ m-sec  | 22: | --00     | NA           |
| 23h    | 1      | "n" such that maximum word program time-out = $2^n$ times typical                                   | 23: | --02     | 256 $\mu$ s  |
| 24h    | 1      | "n" such that maximum buffer write time-out = $2^n$ times typical                                   | 24: | --02     | 2048 $\mu$ s |
| 25h    | 1      | "n" such that maximum block erase time-out = $2^n$ times typical                                    | 25: | --03     | 4s           |
| 26h    | 1      | "n" such that maximum chip erase time-out = $2^n$ times typical                                     | 26: | --00     | NA           |

### A.1.4 Device Geometry Definition

**Table 35: Device Geometry Definition**

| Offset   | Length | Description  | Add | Hex Code        | Value |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|--|--------|--|-----|-----------------|-------|----|---|---|-----|-----|-----|----|---|---|---|---|---|-----|-----|-----|----|
| 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> </table>                  |     |                 |       | 7  | 6 | 5 | 4   | 3   | 2   | 1  | 0 | – | – | – | – | x64 | x32 | x16 | x8 |
|  |        | 7  |     |                 |       | 6  | 5 | 4 | 3   | 2   | 1   | 0  |   |   |   |   |   |     |     |     |    |
|  |        | –  |     |                 |       | –  | – | – | x64 | x32 | x16 | x8 |   |   |   |   |   |     |     |     |    |
| <table border="1" style="width:100%; text-align:center;"> <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> | 15     | 14   | 13  | 12              | 11    | 10 | 9 | 8 | –   | –   | –   | –  | – | – | – | – |   |     |     |     |    |
| 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: | --09            | 512   |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 2B: | --00            |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
| 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 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
| 2D   | 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: | See Table Below |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 2E: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 2F: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 30: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
| 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: | See Table Below |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 32: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 33: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 34: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
| 35h  | 4      | Reserved for future erase block region information   | 35: | See Table Below |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 36: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 37: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |
|  |        |  | 38: |                 |       |    |   |   |     |     |     |    |   |   |   |   |   |     |     |     |    |

| Address | 64-Mbit |      | 128-Mbit |      |
|---------|---------|------|----------|------|
|         | --B     | --T  | --B      | --T  |
| 27:     | --17    | --17 | --18     | --18 |
| 28:     | --01    | --01 | --01     | --01 |
| 29:     | --00    | --00 | --00     | --00 |
| 2A      | --09    | --09 | --09     | --09 |
| 2B      | --00    | --00 | --00     | --00 |
| 2C:     | --02    | --02 | --02     | --02 |
| 2D:     | --03    | --3E | --03     | --7E |
| 2E:     | --00    | --00 | --00     | --00 |
| 2F:     | --80    | --00 | --80     | --00 |

| Address | 64-Mbit |      | 128-Mbit |      |
|---------|---------|------|----------|------|
|         | --B     | --T  | --B      | --T  |
| 30:     | --00    | --02 | --00     | --02 |
| 31:     | --3E    | --03 | --7E     | --03 |
| 32:     | --00    | --00 | --00     | --00 |
| 33:     | --00    | --80 | --00     | --80 |
| 34:     | --02    | --00 | --02     | --00 |
| 35:     | --00    | --00 | --00     | --00 |
| 36:     | --00    | --00 | --00     | --00 |
| 37:     | --00    | --00 | --00     | --00 |
| 38:     | --00    | --00 | --00     | --00 |



Table 37: OTP 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><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 2n = factory pre-programmed bytes<br>bits 24–31 = "n" such that 2n = 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><br>Bits 0–31 point to the Protection register physical Lock-word address in the Jedec-plane.<br><br>Following bytes are factory or user-programmable.<br><br>bits 32–39 = "n" such that n = factory pgm'd groups (low byte)<br>bits 40–47 = "n" such that n = factory pgm'd groups (high byte)<br>bits 48–55 = "n" \ 2n = factory programmable bytes/group<br>bits 56–63 = "n" such that n = user pgm'd groups (low byte)<br>bits 64–71 = "n" such that n = user pgm'd groups (high byte)<br>bits 72–79 = "n" such that 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 38: Burst Read Information**

| Offset<br>P=10Ah | Length | Description<br>(Optional flash features and commands)   | Add. | Hex<br>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: | --04        | 16 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 39: Partition and Erase Block Region Information**

| Offset <sup>(1)</sup><br>P = 10Ah |         | Description<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 40: Partition Region 1 Information (Sheet 1 of 2)

| Offset <sup>(1)</sup> |         | Description<br>(Optional flash features and commands)  | See table below |         |      |
|-----------------------|---------|--|-----------------|---------|------|
| P = 10Ah              |         |  | Len             | Address |      |
| Bottom                | Top     | 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 41: Partition Region 1 Information (Sheet 2 of 2)

| Offset <sup>(1)</sup><br>P = 10Ah                              |  | Description<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               | 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 42: Partition and Erase Block Region Information**

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

**Table 43: CFI Link Information**

| Length | Description<br>(Optional flash features and commands)   | Add.                         | Hex Code | Value |
|--------|---|------------------------------|----------|-------|
| 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: |          | --FF  |
| 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:                         |          | --FF  |

## A.2 Flowcharts

Figure 32: Word Program Flowchart

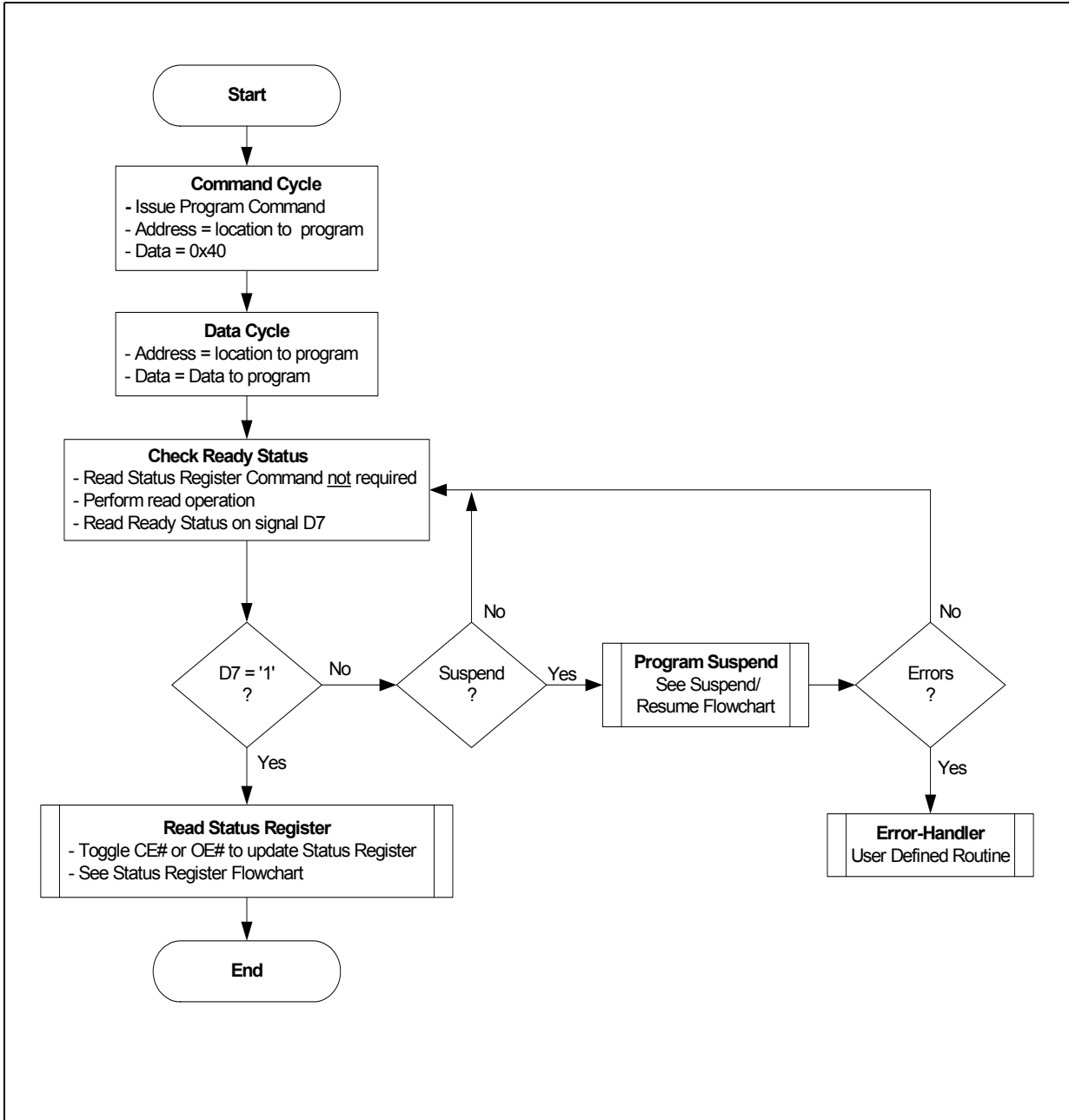


Figure 33: Program Suspend/Resume Flowchart

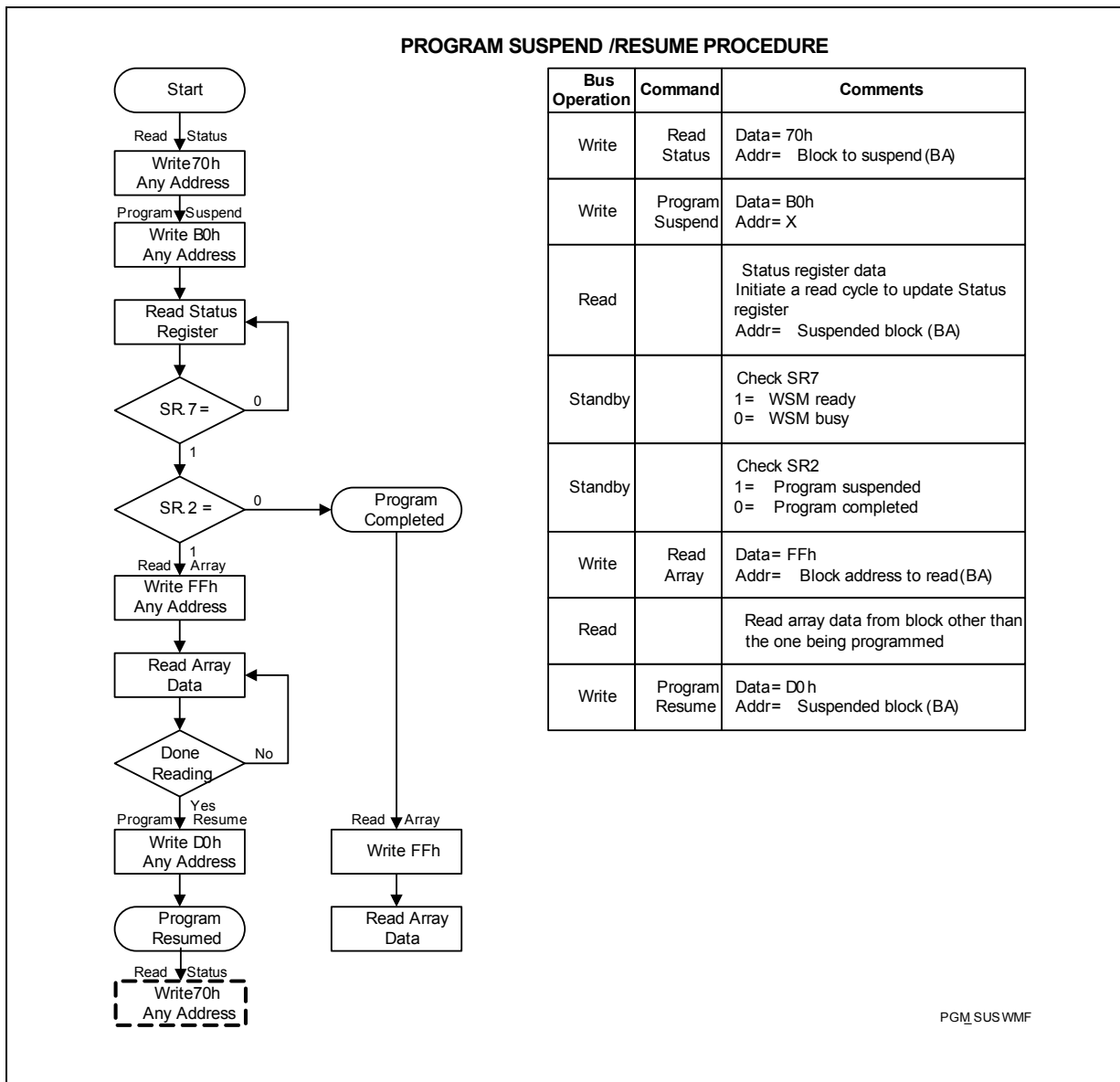


Figure 34: Erase Suspend/Resume Flowchart

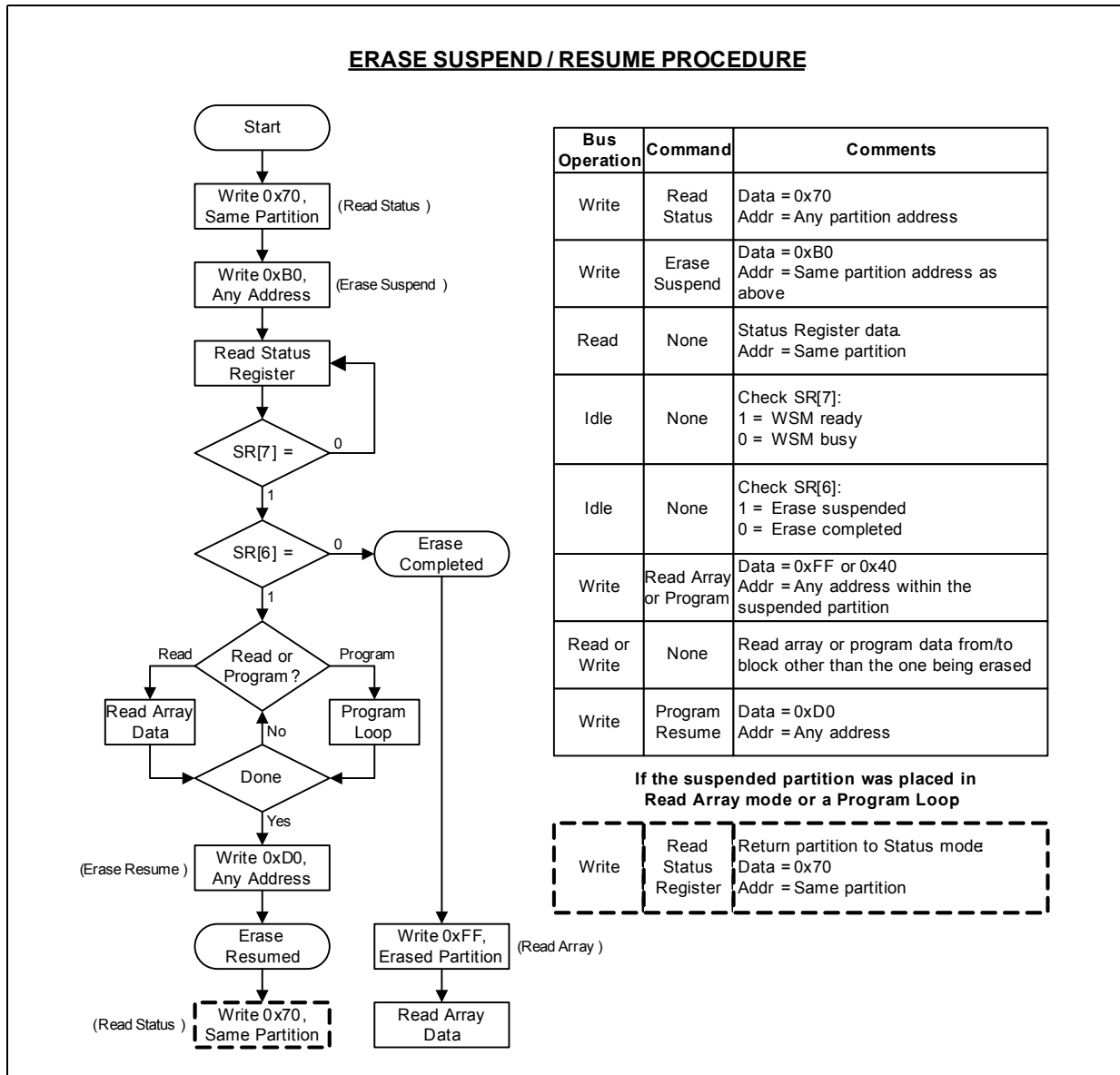
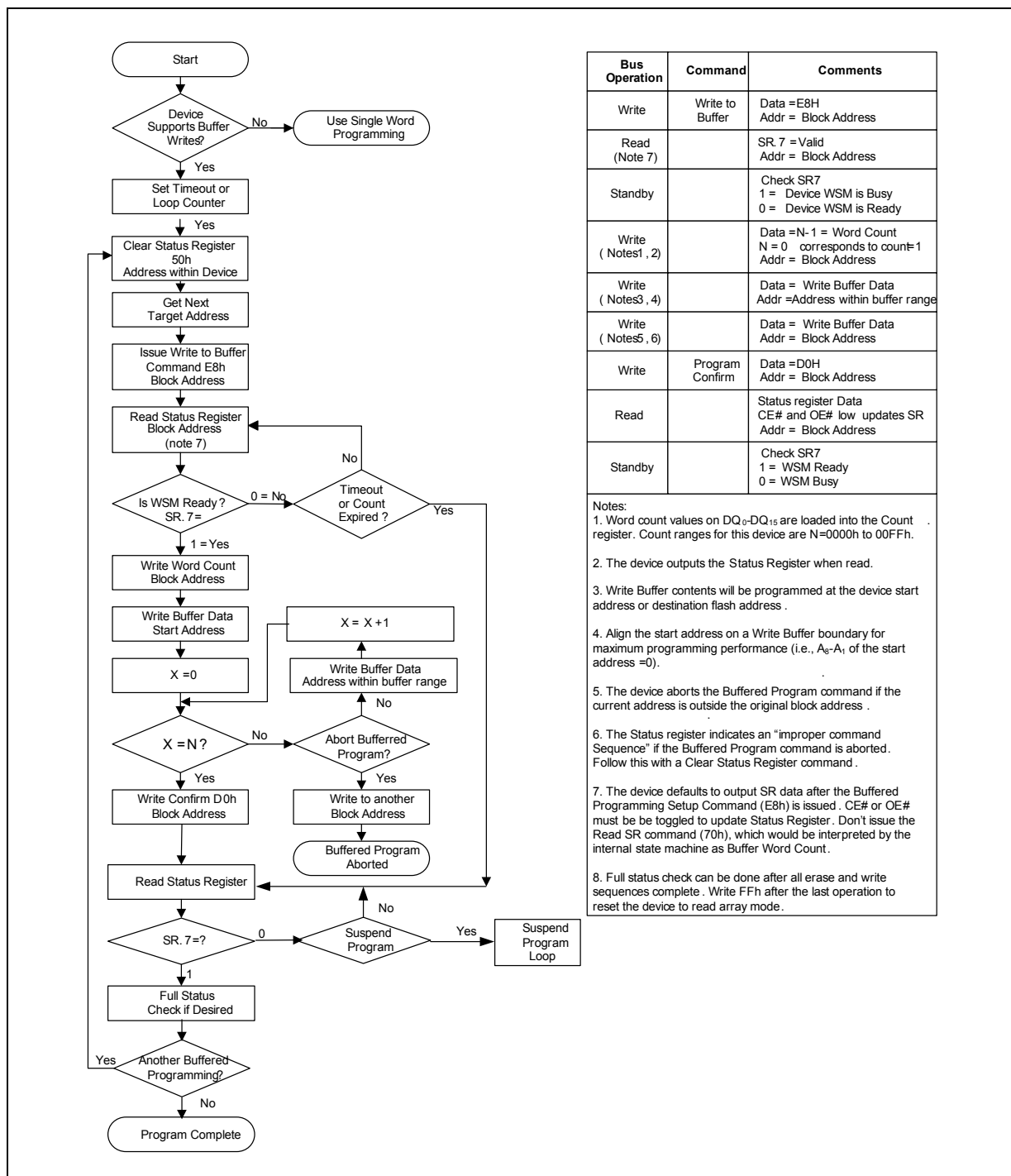


Figure 35: Buffer Program Flowchart



| Bus Operation      | Command         | Comments  |
|--------------------|-----------------|---|
| Write              | Write to Buffer | Data = E8H<br>Addr = Block Address  |
| Read (Note 7)      |                 | SR 7 = Valid<br>Addr = Block Address  |
| Standby            |                 | Check SR7<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 = Address within buffer range                  |
| 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 SR7<br>1 = WSM Ready<br>0 = WSM Busy                                      |

- Notes:
- Word count values on DQ<sub>0</sub>-DQ<sub>15</sub> are loaded into the Count register. Count ranges for this device are N=0000h to 00FFh.
  - The device outputs the Status Register when read.
  - Write Buffer contents will be programmed at the device start address or destination flash address.
  - Align the start address on a Write Buffer boundary for maximum programming performance (i.e., A<sub>0</sub>-A<sub>1</sub> of the start address = 0).
  - The device aborts the Buffered Program command if the current address is outside the original block address.
  - The Status register indicates an "improper command Sequence" if the Buffered Program command is aborted. Follow this with a Clear Status Register command.
  - The device defaults to output SR data after the Buffered Programming Setup Command (E8h) is issued. CE# or OE# must be toggled to update Status Register. Don't issue the Read SR command (70h), which would be interpreted by the internal state machine as Buffer Word Count.
  - 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 36: Program Suspend/Resume Flowchart

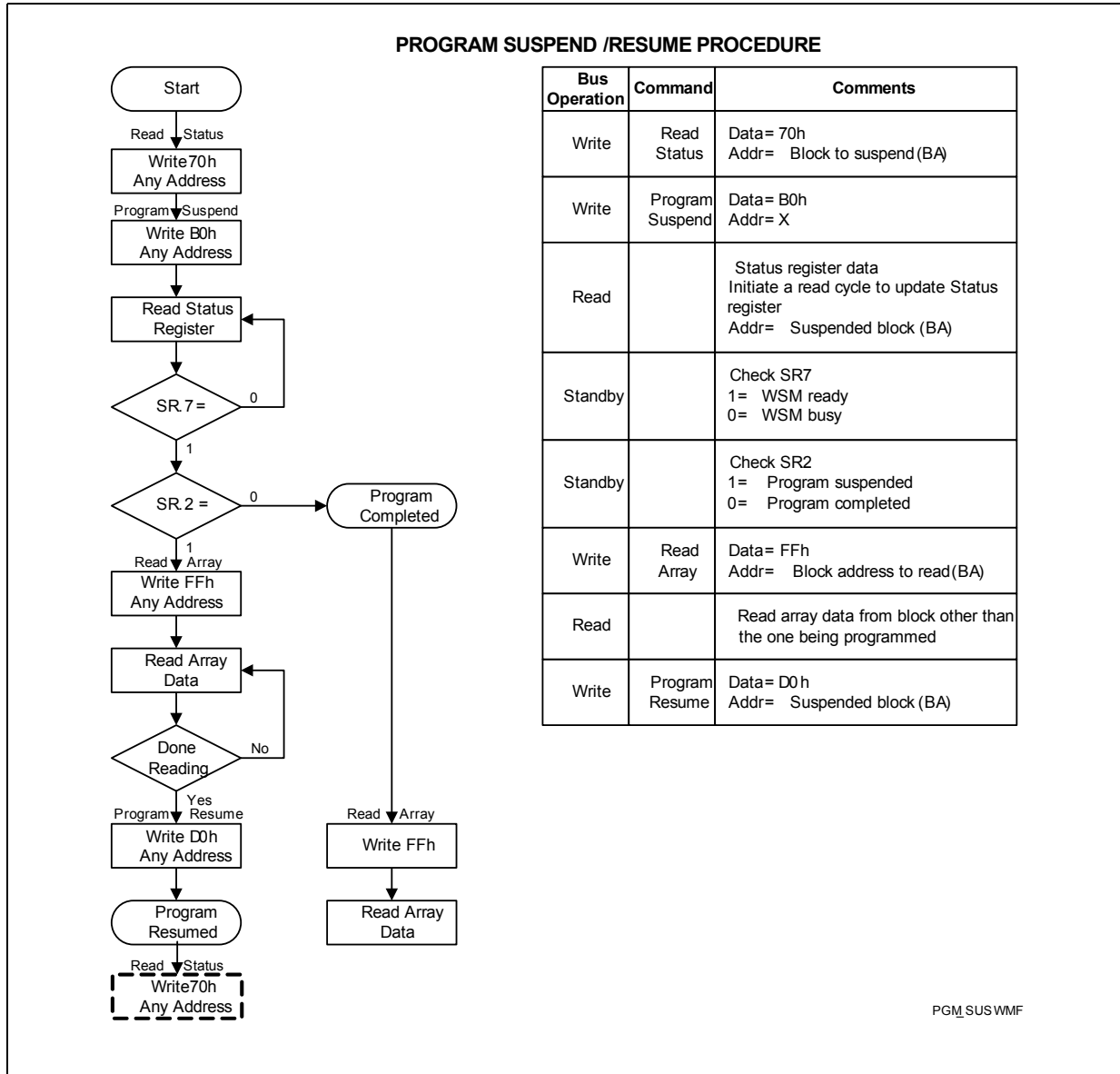


Figure 37: BEFP Flowchart

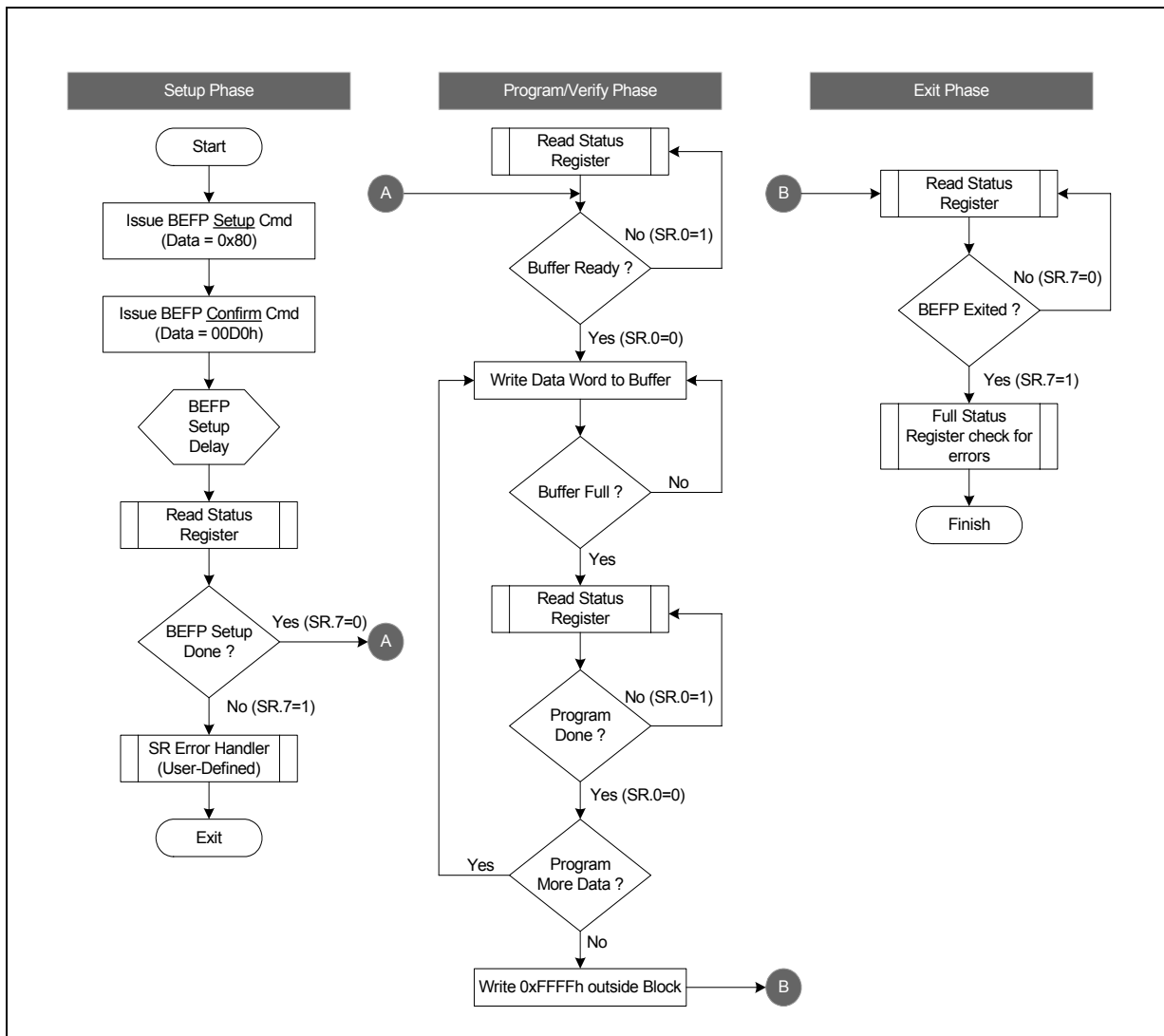


Figure 38: Block Erase Flowchart

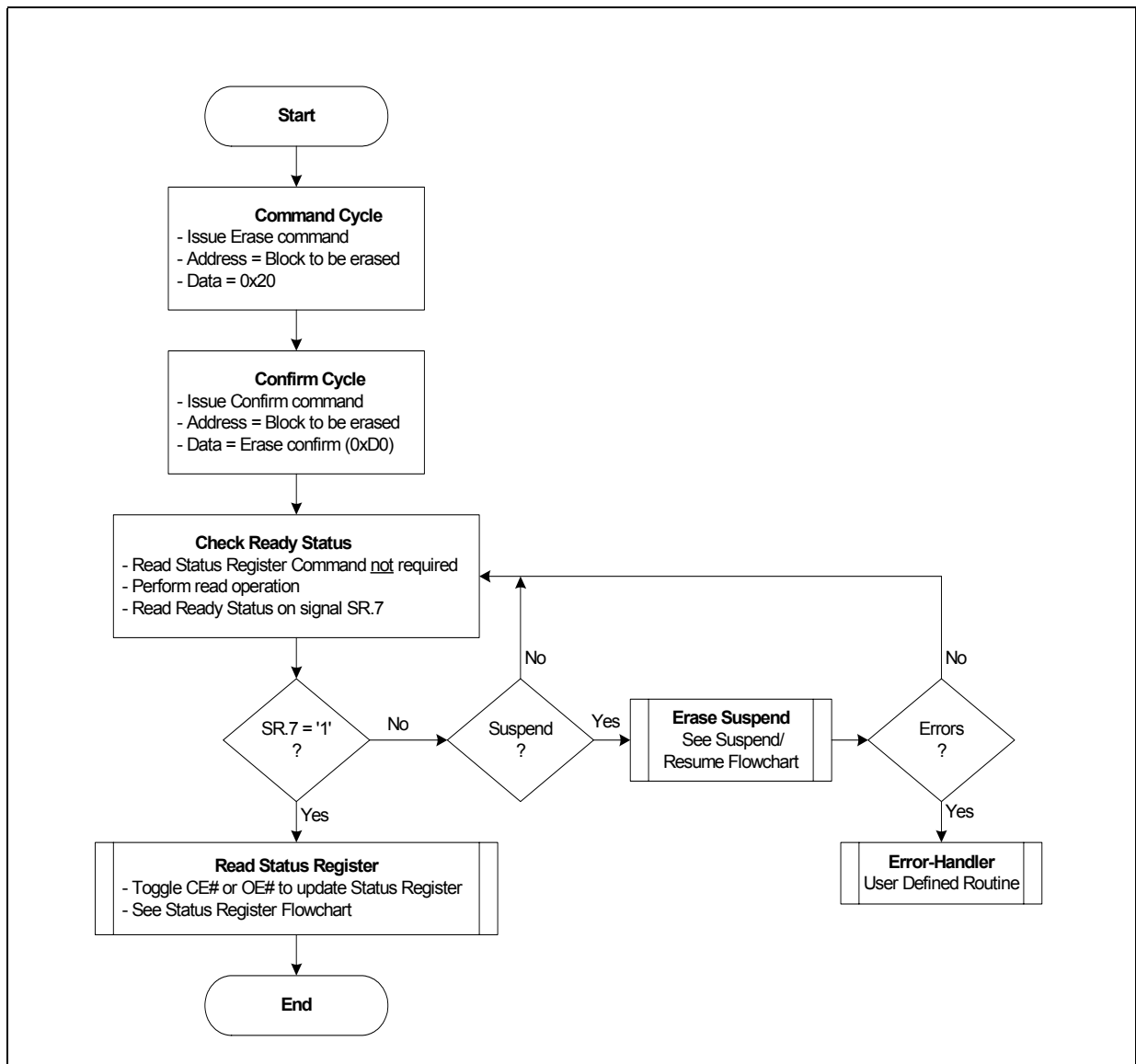


Figure 39: Block Lock Operations Flowchart

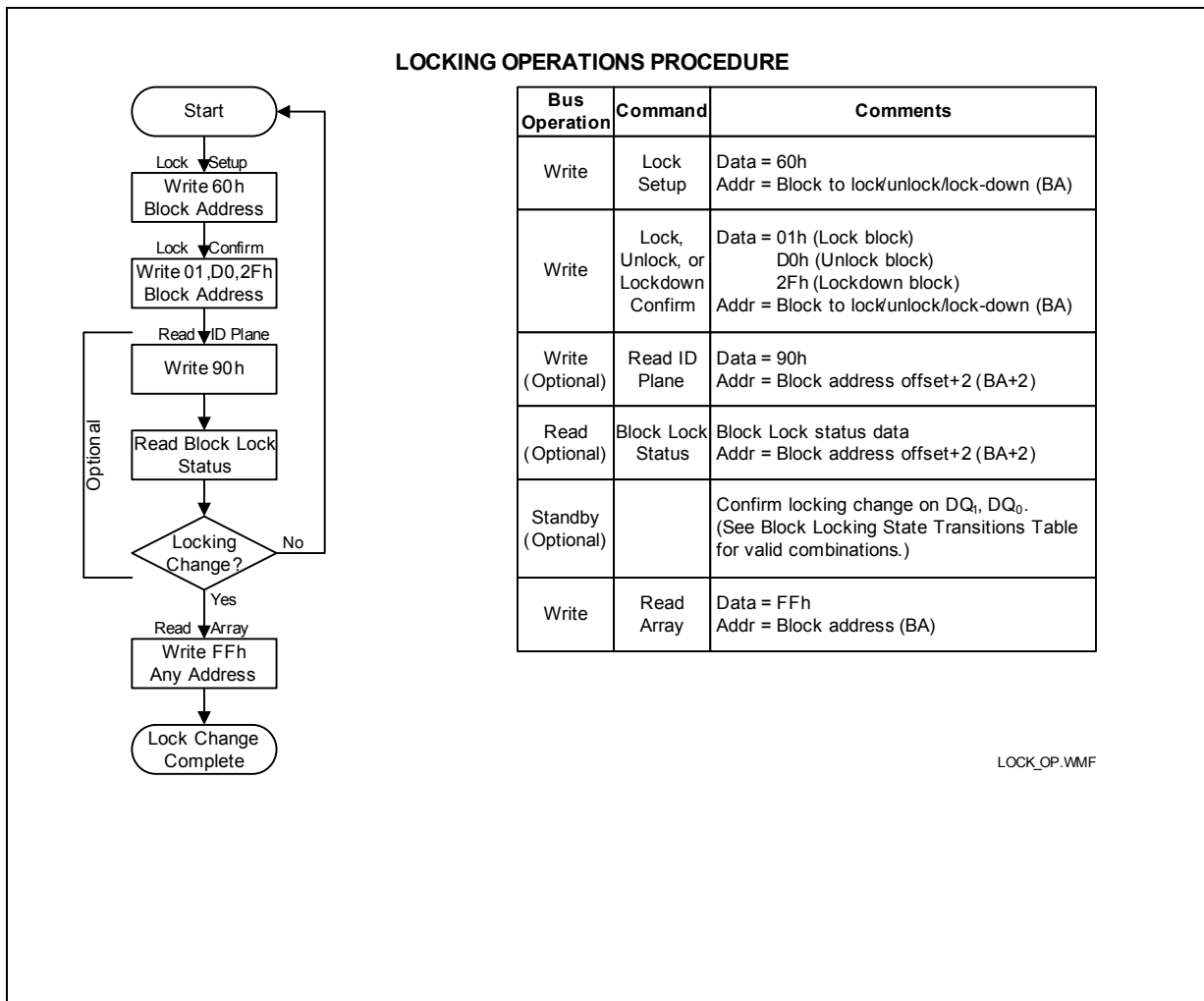


Figure 40: OTP Register Programming Flowchart

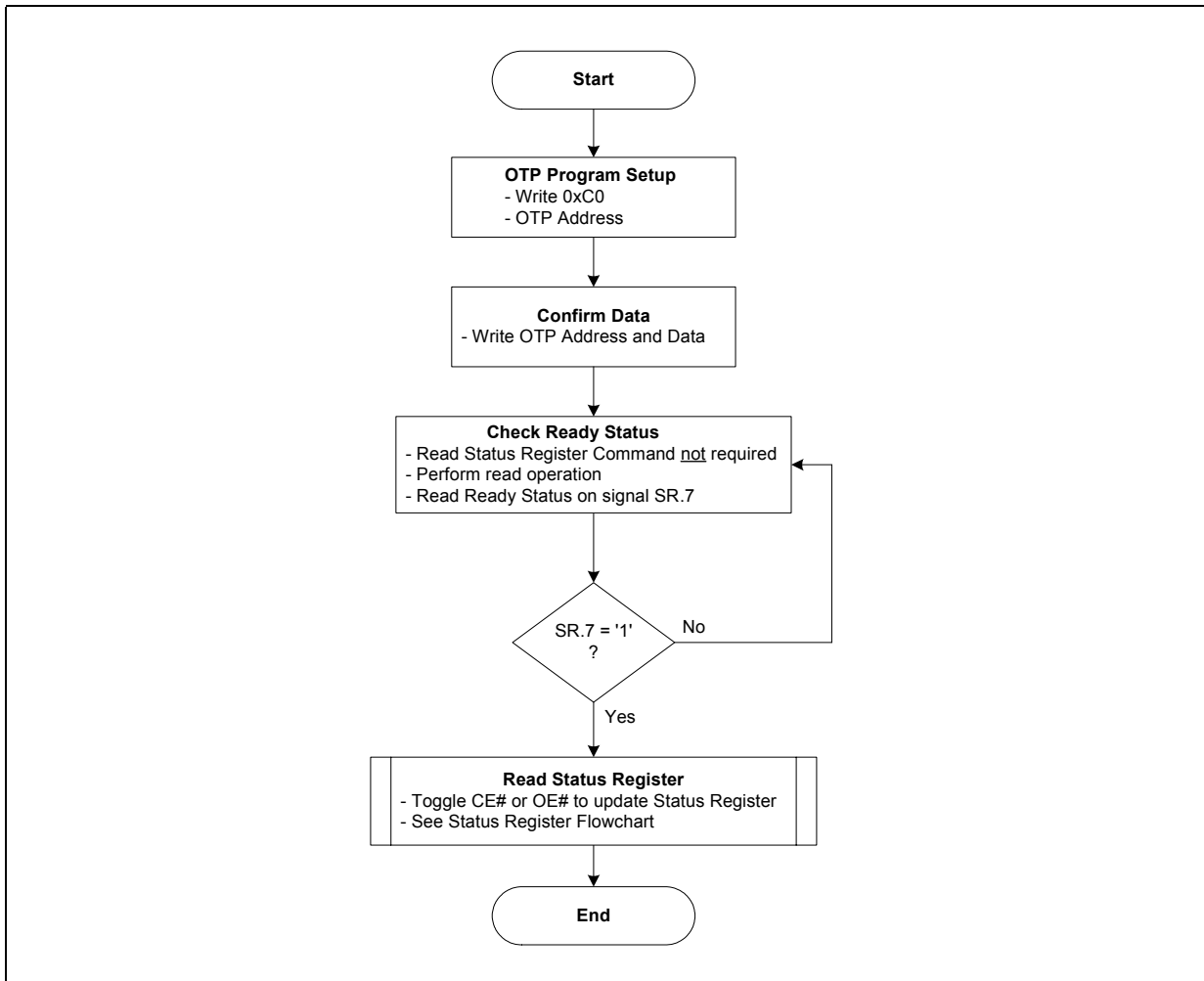
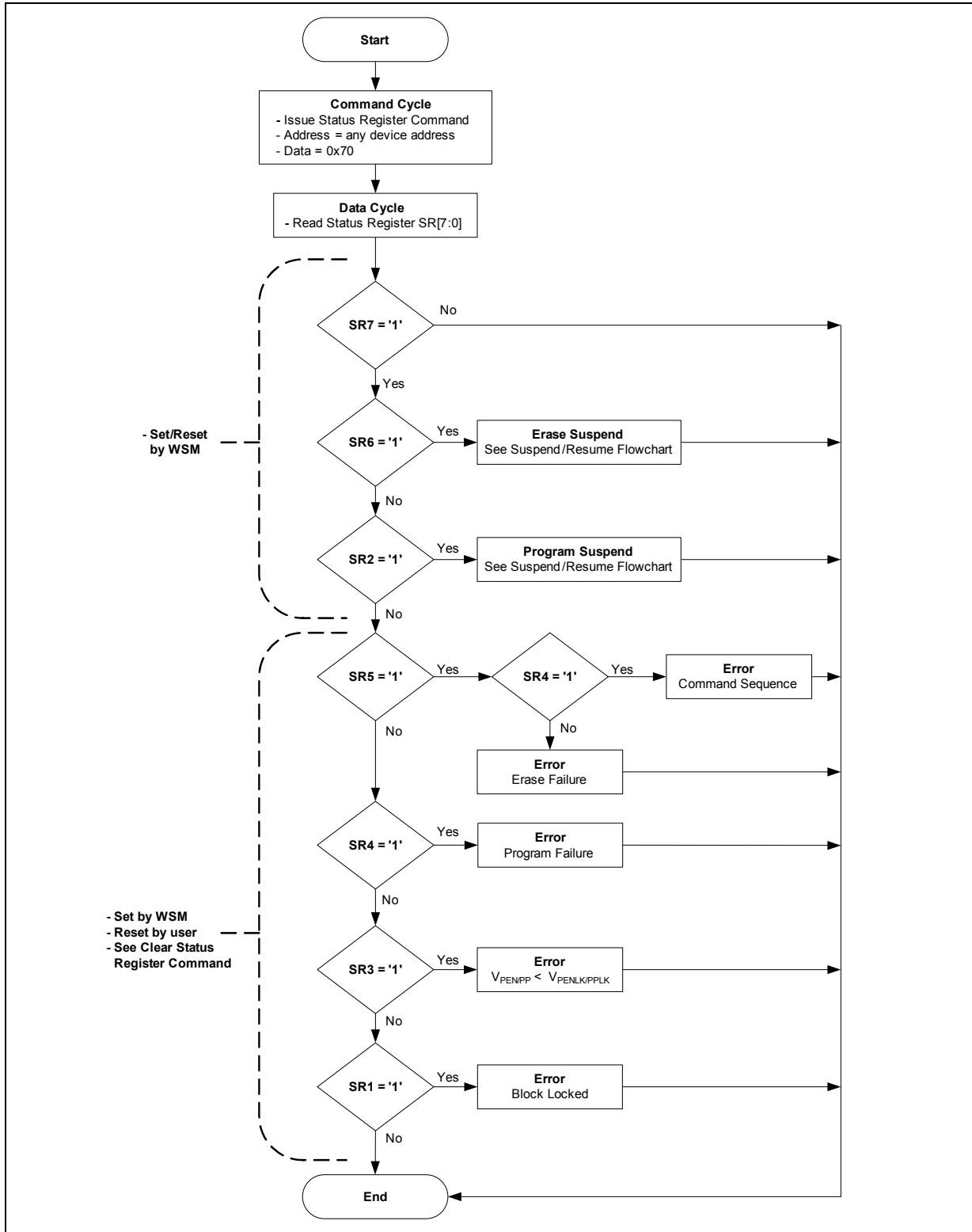


Figure 41: Status Register Flowchart



### A.3 Write State Machine

Show here are 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, Read CFI or Read Status Register) until a new command changes it. The next WSM state does not depend on the partition's output state.

Note: IS refers to Illegal State in the Next State Tables.

**Table 44: Next State Table for P3x-65nm (Sheet 1 of 3)**

| Current Chip State |                            | Command Input and Resulting Chip Next State <sup>(1)</sup>                                |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | WSM Operation Completes |                            |                           |                           |                      |
|--------------------|----------------------------|---|----------------------|--------------|----------------------------|----------------------|----------------------------|---------------------------|---------------------|------------------------------|----------------------|----------------------------|--------------------|-------------------------|----------------|-------------------------|----------------------------|---------------------------|---------------------------|----------------------|
|                    |                            | Array Read (3)  | Word Pgm Setup (4,9) | BP Setup (8) | EFI Command Setup          | Erase Setup (4,9)    | BEFP Setup (6)             | Confirm (7)               | Pgm/Ers Suspend     | Read Status                  | Clear SR (5)         | Read ID/Query              | Lock/RCR/ECR Setup | Blank Check             | OTP Setup      |                         | Lock Blk Confirm (7)       | Lock-down Blk Confirm (7) | Write ECR/RCR Confirm (7) | Block Address Change |
|                    |                            | (FFh)   | (40h)                | (E8h)        | (EBh)                      | (20h)                | (80h)                      | (D0h)                     | (B0)                | (70h)                        | (50h)                | (90h, 98h)                 | (60h)              | (BCh)                   | (C0h)          | (01h)                   | (2Fh)                      | (03h, 04h)                |                           | other                |
| Ready              | Ready                      | Program Setup   | BP Setup             | EFI Setup    | Erase Setup                | BEFP Setup           | Ready                      |                           |                     |                              | Lock/RCR/ECR Setup   | BC Setup                   | OTP Setup          | Ready                   |                |                         | N/A                        | Ready                     | N/A                       |                      |
| Lock/RCR/ECR Setup | Ready (Lock Error [Botch]) |   |                      |              |                            | Ready (Unlock Block) | Ready (Lock Error [Botch]) |                           |                     |                              |                      | Ready (Lock Error [Botch]) | Ready (Lock Block) | Ready (Lock down Block) | Ready (Set CR) | N/A                     | Ready (Lock Error [Botch]) | N/A                       |                           |                      |
| OTP                | Setup                      | OTP Busy  |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     | OTP Busy                   | N/A                       |                           |                      |
|                    | Busy                       | OTP Busy  | IS in OTP Busy       | OTP Busy     | IS in OTP Busy             | OTP Busy             |                            |                           |                     | Illegal State in OTP Busy    | OTP Busy             |                            |                    | N/A                     | OTP Busy       | Ready                   |                            |                           |                           |                      |
|                    | IS in OTP Busy             | OTP Busy  |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     | OTP Busy                   | Ready                     |                           |                      |
| Word Program       | Setup                      | Word Program Busy   |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     | Pgm Busy                   | N/A                       |                           |                      |
|                    | Busy                       | Pgm Busy  | IS in Pgm Busy       | Pgm Busy     | IS in Pgm Busy             | Pgm Busy             | Pgm Susp                   | Word Pgm Busy             | IS in Word Pgm Busy | Word Pgm Busy                | N/A                  | Pgm Busy                   | Ready              |                         |                |                         |                            |                           |                           |                      |
|                    | IS in Pgm Busy             | Word Pgm Busy   |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     | Pgm Busy                   | Ready                     |                           |                      |
|                    | Suspend                    | Pgm Susp  | IS in Pgm Susp       | Pgm Suspend  | IS in Pgm Susp             | Pgm Busy             | Pgm Susp                   | Pgm Susp (Er bits clear)  | Word Pgm Susp       | Illegal State in Pgm Suspend | Word Program Suspend | N/A                        | Word Pgm Susp      | N/A                     |                |                         |                            |                           |                           |                      |
|                    | IS in Pgm Suspend          | Word Program Suspend  |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     | Word Pgm Susp              | N/A                       |                           |                      |
| EFI                | EFI Setup                  | Sub-function Setup  |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |
|                    | Sub-function Setup         | Sub-op-code Load 1  |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |
|                    | Sub-op-code Load 1         | Sub-function Load 2 if word count >0, else Sub-function confirm                           |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |
|                    | Sub-function Load 2        | Sub-function Confirm if data load in program buffer is complete, ELSE Sub-function Load 2 |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |
|                    | Sub-function Confirm       | Ready (Error [Botch])   |                      |              |                            |                      | S-fn Busy                  | Ready (Error [Botch])     |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |
|                    | Sub-function Busy          | S-fn Busy   | IS in S-fn Busy      | S-fn Busy    | Illegal State in S-fn Busy | S-fn Busy            | S-fn Susp                  | S-fn Busy                 | IS in S-fn Busy     | S-fn Busy                    | S-fn Busy            | S-fn Busy                  | S-fn Busy          | S-fn Busy               | S-fn Busy      | S-fn Busy               | Ready                      |                           |                           |                      |
|                    | IS in Sub-function Busy    | Sub-function Busy   |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |
|                    | Sub-function Susp          | S-fn Susp   | IS in S-fn Susp      | Sub-function | Illegal State in S-fn Busy | S-fn Busy            | S-fn Suspend               | S-fn Susp (Er bits clear) | S-fn Susp           | IS in S-fn Susp              | S-fn Suspend         | N/A                        | S-fn Susp          | N/A                     |                |                         |                            |                           |                           |                      |
|                    | IS in S-fn Susp            | Sub-function Suspend  |                      |              |                            |                      |                            |                           |                     |                              |                      |                            |                    |                         |                | N/A                     |                            | N/A                       |                           |                      |

Table 44: Next State Table for P3x-65nm (Sheet 2 of 3)

| Current Chip State        |  | Command Input and Resulting Chip Next State <sup>(1)</sup>  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             | WSM Operation Completes     |                             |   |                                |                             |                             |                             |
|---------------------------|--|---|------------------------------|-----------------------------|--|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------------|-----------------------------|-----------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|---|--------------------------------|-----------------------------|-----------------------------|-----------------------------|
|                           |  | Array Read (3)  | Word Pgm Setup (4,9)         | BP Setup (8)                | EFI Command Setup                      | Erase Setup (4,9)           | BEFP Setup (6)              | Confirm (7)                 | Pgm/Ers Suspend             | Read Status                   | Clear SR (5)                      | Read ID/Query               | Lock/RCR/ECR Setup                | Blank Check                 | OTP Setup                   | Lock Blk Confirm (7)        | Lock-down Blk Confirm (7)   |                             | Write ECR/RCR Confirm (7)   | Block Address Change  | Other Commands (2)             |                             |                             |                             |
|                           |  | (FFh)   | (40h)                        | (E8h)                       | (EBh)                                  | (20h)                       | (80h)                       | (D0h)                       | (B0)                        | (70h)                         | (50h)                             | (90h, 98h)                  | (60h)                             | (BCh)                       | (C0h)                       | (01h)                       | (2Fh)                       | (03h, 04h)                  |                             | other   |                                |                             |                             |                             |
| Buffer Pgm (BP)           | Setup  | BP Load 1   |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | BP Load 1 <sup>(6)</sup>                               | BP Load 2 if word count >0, else BP confirm   |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | BP Load 2 <sup>(8)</sup>                               | BP Confirm if data load in program buffer is complete, ELSE BP load 2                                   |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             | Ready (Error [Botch])       | BP Confirm if data load in program buffer is complete, else BP load 2 | N/A                            |                             |                             |                             |
|                           | BP Confirm   | Ready (Error [Botch])   |                              |                             |  |                             |                             |                             | BP Busy                     | Ready (Error [Botch])         |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | BP Busy  | BP Busy   | IS in BP Busy                | BP Busy                     | Illegal State in BP Busy               | BP Busy                     | BP Susp                     | BP Busy                     | BP Susp                     | BP Busy                       | IS in BP Busy                     | BP Busy                     | BP Busy                           | IS in BP Busy               | BP Busy                     | BP Busy                     | BP Busy                     | BP Busy                     | BP Busy                     | BP Busy   | Ready                          |                             |                             |                             |
|                           | IS in BP Busy  | BP Busy   |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
| Erase                     | BP Susp  | BP Susp   | IS in BP Susp                | BP Suspend                  | Illegal State in BP Busy               | BP Busy                     | BP Suspend                  | BP Susp                     | BP Susp                     | IS in BP Susp                 | BP Suspend                        | BP Susp                     | IS in BP Susp                     | BP Suspend                  | N/A                         | BP Susp                     | N/A                         | BP Susp                     | N/A                         | N/A   | BP Susp                        | N/A                         |                             |                             |
|                           | IS in BP Susp  | BP Suspend  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | Setup  | Ready (Error [Botch])   |                              |                             |  |                             |                             |                             | Erase Busy                  | Ready (Error [Botch])         |                                   |                             |                                   |                             |                             |                             | N/A                         | Ready (Err Botch0)          | N/A                         |   |                                |                             |                             |                             |
|                           | Busy   | Erase Busy  | IS in Erase Busy             | Erase Busy                  | IS in Erase Busy                       | Erase Busy                  | Erase Susp                  | Erase Busy                  | Erase Susp                  | Erase Busy                    | IS in Erase Busy                  | Erase Busy                  | Erase Susp                        | IS in Erase Busy            | Erase Busy                  | Erase Susp                  | Erase Busy                  | Erase Susp                  | Erase Busy                  | Erase Susp  | Erase Busy                     | Erase Susp                  | Erase Busy                  | Erase Susp                  |
| Word Pgm in Erase Suspend | IS in Erase Busy                                       | Erase Busy  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   | Ready                          |                             |                             |                             |
|                           | Suspend  | Erase Susp  | Word Pgm Setup in Erase Susp | BP Setup in Erase Susp      | EFI Setup in Erase Susp                | IS in Erase Suspend         | Erase Busy                  | Erase Suspend               | Erase Susp                  | Erase Susp                    | IS in Word Pgm busy in Erase Susp | Erase Susp                  | IS in Word Pgm busy in Erase Susp | Erase Susp                  | IS in Erase Susp            | Erase Susp                  | Erase Susp                  | Erase Susp                  | Erase Susp                  | Erase Susp  | Erase Susp                     | Erase Susp                  | Erase Susp                  |                             |
|                           | IS in Erase Susp                                       | Erase Suspend   |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | Setup  | Word Pgm busy in Erase Suspend  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   | N/A                            |                             |                             |                             |
| BP in Erase Suspend       | Busy   | Word Pgm busy in Erase Susp   | IS in Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | IS in Word Pgm busy in Erase Susp      | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp   | Word Pgm busy in Erase Susp       | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp       | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp   | Word Pgm busy in Erase Susp    | Word Pgm busy in Erase Susp | Word Pgm busy in Erase Susp |                             |
|                           | Illegal state (IS) in Pgm busy in Erase Suspend        | Word Pgm busy in Erase Suspend  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   | Word Pgm Busy in Erase Suspend | IS in Erase Susp            |                             |                             |
|                           | Suspend  | Word Pgm susp in Erase Susp   | IS in pgm susp in Erase Susp | Word Pgm susp in Erase Susp | IS in pgm susp in Erase Susp           | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp   | Word Pgm susp in Erase Susp       | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp       | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp   | Word Pgm susp in Erase Susp    | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp | Word Pgm susp in Erase Susp |
|                           | Illegal State in Word Program Suspend in Erase Suspend | Word Pgm busy in Erase Suspend  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
| BP in Erase Suspend       | Setup  | BP Load 1 in Erase Suspend  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | BP Load 1 <sup>(6)</sup>                               | BP Load 2 in Erase Suspend if word count >0, else BP confirm  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | BP Load 2 <sup>(8)</sup>                               | BP Confirming Erase Suspend if data load in program buffer is complete, ELSE BP load 2 in Erase Suspend |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             | Ers Susp (Error [Botch])    | BP Confirm in Erase Suspend when count=0, ELSE BP load 2              | N/A                            |                             |                             |                             |
|                           | BP Confirm   | Erase Suspend (Error [BotchBP])   |                              |                             |  |                             |                             |                             | BP Busy in Erase Susp       | Erase Susp (Error [Botch BP]) |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |
|                           | BP Busy  | BP Busy in Erase Susp   | IS in BP Busy in Erase Susp  | BP Busy in Erase Susp       | Illegal State in BP Busy in Erase Susp | BP Busy in Erase Susp       | BP Susp in Erase Susp       | BP Busy in Erase Susp       | BP Susp in Erase Susp       | BP Busy in Erase Susp         | IS in BP Busy in Erase Susp       | BP Busy in Erase Susp       | BP Susp in Erase Susp             | IS in BP Busy in Erase Susp | BP Busy in Erase Susp       | BP Busy in Erase Susp       | BP Busy in Erase Susp       | BP Busy in Erase Susp       | BP Busy in Erase Susp       | BP Busy in Erase Susp   | BP Busy in Erase Susp          | BP Busy in Erase Susp       | BP Busy in Erase Susp       |                             |
|                           | IS in BP Busy  | BP Busy in Erase Suspend  |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   | IS in Erase Susp               |                             |                             |                             |
| BP in Erase Suspend       | BP Susp  | BP Susp in Erase Susp   | IS in BP Susp in Erase Susp  | BP Suspend in Erase Suspend | Illegal State in BP Busy in Erase Susp | BP Busy in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp         | IS in BP Busy in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp             | IS in BP Busy in Erase Susp | BP Susp in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp       | BP Susp in Erase Susp   | BP Susp in Erase Susp          | BP Susp in Erase Susp       | BP Susp in Erase Susp       |                             |
|                           | IS in BP Susp  | BP Suspend in Erase Suspend   |                              |                             |  |                             |                             |                             |                             |                               |                                   |                             |                                   |                             |                             |                             |                             |                             |                             |   |                                |                             |                             |                             |

Table 44: Next State Table for P3x-65nm (Sheet 3 of 3)

|  |                                       | Command Input and Resulting Chip Next State <sup>(1)</sup>   |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         |                       |                |
|--|---------------------------------------|--|-----------------------------|--------------------------|--|--------------------------|-------------------------------|---------------------------------------|-------------------------------|-----------------------------|--------------------------|-----------------------------|-----------------------|----------------------|-----------------------|----------------------|---------------------------|---------------------------|--------------------------|---|-------------------------|-----------------------|----------------|
| Current Chip State                                 |                                       | Array Read (3)   | Word Pgm Setup (4,9)        | BP Setup (8)             | EFI Command Setup                      | Erase Setup (4,9)        | BEFP Setup (6)                | Confirm (7)                           | Pgm/Ers Suspend               | Read Status                 | Clear SR (5)             | Read ID/Query               | Lock/RCR/ECR Setup    | Blank Check          | OTP Setup             | Lock Blk Confirm (7) | Lock-down Blk Confirm (7) | Write ECR/RCR Confirm (7) | Block Address Change     | Other Commands (2)  | WSM Operation Completes |                       |                |
|  |                                       | (Ffh)  | (40h)                       | (E8h)                    | (EBh)                                  | (20h)                    | (80h)                         | (D0h)                                 | (B0)                          | (70h)                       | (50h)                    | (90h, 98h)                  | (60h)                 | (BCh)                | (C0h)                 | (01h)                | (2Fh)                     | (03h, 04h)                |                          | other   |                         |                       |                |
| EFI in Erase Suspend                               | EFI Setup                             | Sub-function Setup in Erase Suspend  |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         | N/A                   |                |
|  | Sub-function Setup                    | Sub-op-code Load 1 in Erase Suspend  |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         |                       |                |
|  | Sub-op-code Load 1                    | Sub-function Load 2 in Erase Suspend if word count >0, else Sub-function confirm in Erase Suspend                                |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         |                       |                |
|  | Sub-function Load 2                   | Sub-function Confirm in Erase Suspend if data load in program buffer is complete, ELSE Sub-function Load 2                       |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           | Ers Susp (Error [Botch]) | Sub-function Confirm if data load in program buffer is complete, ELSE Sub-function Load 2 |                         |                       |                |
|  | Sub-function Confirm                  | Erase Suspend (Error [Botch])  |                             |                          |  |                          |                               | S-fn Busy in Ers Susp                 | Erase Suspend (Error [Botch]) |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         |                       |                |
|  | Sub-function Busy                     | S-fn Busy in Ers Susp  | IS in S-fn Busy in Ers Susp | S-fn Busy in Ers Suspend | Illegal State in S-fn Busy in Ers Susp | S-fn Busy in Ers Susp    | S-fn Susp in Ers Susp         | S-fn Busy in Ers Susp                 | S-fn Busy in Ers Susp         | IS in S-fn Busy in Ers Susp | S-fn Busy in Ers Susp    | IS in S-fn Busy in Ers Susp | S-fn Busy in Ers Susp | N/A                  | S-fn Busy in Ers Susp | Erase Susp           |                           |                           |                          |   |                         |                       |                |
|  | IS in Sub-function Busy               | Sub-function Busy in Ers Susp  |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         |                       | IS in Ers Susp |
|  | Sub-function Susp                     | S-fn Susp in Ers Susp  | IS in S-fn Susp in Ers Susp | S-fn Suspend in Ers Susp | Illegal State in S-fn Busy in Ers Susp | S-fn Busy in Ers Susp    | S-fn Suspend in Ers Susp      | S-fn Susp in Ers Susp (Er bits clear) | S-fn Susp in Ers Susp         | IS in S-fn Susp in Ers Susp | S-fn Suspend in Ers Susp | N/A                         | S-fn Susp in Ers Susp | Erase Susp           |                       |                      |                           |                           |                          |   |                         |                       |                |
| IS in Phase-1 Susp                                 | Sub-Function Suspend in Erase Suspend |  |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   | N/A                     |                       |                |
| Lock/RCR/ECR/Lock EFA Block Setup in Erase Suspend | Erase Suspend (Lock Error [Botch])    |  |                             |                          |  | Ers Susp (Un-lock Block) | Ers Susp (Lock Error [Botch]) |                                       |                               |                             |                          | Ers Susp (Error [Botch])    | Ers Susp Blk Lock     | Ers Susp Blk Lk-Down | Ers Susp CR Set       | N/A                  | Ers Susp (Error [Botch])  | N/A                       |                          |   |                         |                       |                |
| Blank Check  | Setup                                 | Ready (Error [Botch])  |                             |                          |  |                          |                               | BC Busy                               | Ready (Error [Botch])         |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         | Ready (Error [Botch]) | N/A            |
|  | Blank Check Busy                      | BC Busy  | IS in BC Busy               | BC Busy                  | IS in BC Busy                          | Blank Check Busy         |                               |                                       | IS in BC Busy                 | BC Busy                     | N/A                      |                             |                       |                      |                       |                      |                           |                           |                          |   | BC Busy                 | Ready                 |                |
|  | IS in Blank Check Busy                | BP Busy  |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         |                       |                |
| BEFP   | Setup                                 | Ready (Error [Botch])  |                             |                          |  |                          |                               | BEFP Load Data                        | Ready (Error [Botch])         |                             |                          |                             |                       |                      |                       |                      |                           |                           |                          |   |                         | N/A                   |                |
|  | BEFP Busy                             | BEFP Program and Verify Busy (if Block Address given matches address given on BEFP Setup command). Commands treated as data. (7) |                             |                          |  |                          |                               |                                       |                               |                             |                          |                             |                       |                      |                       |                      |                           |                           | Ready                    | BEFP Busy   | Ready                   |                       |                |

**Table 45: Output Next State Table for P3x-65nm**

|  |                            | Command Input to Chip and Resulting Output MUX Next State <sup>(1)</sup> |                           |                   |                            |                |             |                 |             |              |               |                    |             |             |                      |                           |                           |                      |                    |                            |
|--|----------------------------|--|---------------------------|-------------------|----------------------------|----------------|-------------|-----------------|-------------|--------------|---------------|--------------------|-------------|-------------|----------------------|---------------------------|---------------------------|----------------------|--------------------|----------------------------|
| Current Chip State   | Array Read (3)             | Word Pgm Setup (4,9)   | BP Setup (8)              | EFI Command Setup | Erase Setup (4,9)          | BEFP Setup (6) | Confirm (7) | Pgm/Ers Suspend | Read Status | Clear SR (5) | Read ID/Query | Lock/RCR/ECR Setup | Blank Check | OTP Setup   | Lock Blk Confirm (7) | Lock-down Blk Confirm (7) | Write ECR/RCR Confirm (7) | Block Address Change | Other Commands (2) | WSM Operation Completes    |
|  | (FFh)                      | (40h)  | (E8h)                     | (EBh)             | (20h)                      | (80h)          | (D0h)       | (B0)            | (70h)       | (50h)        | (90h, 98h)    | (60h)              | (BCh)       | (C0h)       | (01h)                | (2Fh)                     | (03h, 04h)                |                      | other              |                            |
| BEFP Setup, BEFP Pgm & Verify Busy, Erase Setup, OTP Setup, BP Setup, Load 1, Load 2 BP Setup, Load1, Load 2 - in Erase Susp. BP Confirm EFI Sub-function Confirm WordPgmSetup, Word Pgm Setup in Erase Susp, BP Confirm in Erase Suspend, EFI S-fn Confirm in Ers Susp, Blank Check Setup, Blank Check Busy | Status Read                |  |                           |                   |                            |                |             |                 |             |              |               |                    |             |             |                      |                           |                           |                      |                    |                            |
| Lock/RCR/ECR Setup, Lock/RCR/ECR Setup in Erase Susp   | Status Read                |  |                           |                   |                            |                |             |                 |             |              |               |                    |             |             |                      |                           |                           | Array Read           |                    |                            |
| EFI S-fn Setup, Ld 1, Ld 2 EFI S-fn Setup, Ld1, Ld 2 - in Erase Susp.  | Output MUX will not change |  |                           |                   |                            |                |             |                 |             |              |               |                    |             |             |                      |                           |                           |                      |                    |                            |
| BP Busy BP Busy in Erase Suspend EFI Sub-function Busy EFI Sub-fn Busy in Ers Susp Word Program Busy, Word Pgm Busy in Erase Suspend, OTP Busy Erase Busy  | Status Read                | Status Read  | Status Read               | Status Read       | Output MUX Does not Change | Status Read    | Array Read  | Status Read     | Status Read | Status Read  | Status Read   | Status Read        | Status Read | Status Read | Status Read          | Status Read               | Status Read               | Status Read          | Status Read        | Output MUX does not Change |
| Ready, Word Pgm Suspend, BP Suspend, Phase-1 BP Suspend, Erase Suspend, BP Suspend in Erase Suspend Phase-1 BP Susp in Ers Susp  | Array Read                 | Array Read   | Output MUX doesn't Change | Array Read        | Array Read                 | Array Read     | Array Read  | Array Read      | Array Read  | Array Read   | Array Read    | Array Read         | Array Read  | Array Read  | Array Read           | Array Read                | Array Read                | Array Read           | Array Read         | Array Read                 |

**Notes:**

- IS refers to *Illegal State* in the Next State Table.
- "Illegal commands" include commands outside of the allowed command set.
- The device defaults to "Read Array" on powerup.
- If a "Read Array" is attempted when the device is busy, the result will be "garbage" data (we should not tell the user that it will actually be Status Register data). The key point is that the output mux will be pointing to the "array", but garbage data will be output. "Read ID" and "Read Query" commands do the exact same thing in the device. The ID and Query data are located at different locations in the address map.
- The Clear Status command only clears the error bits in the Status Register if the device is not in the following modes: 1. WSM running (Pgm Busy, Erase Busy, Pgm Busy In Erase Suspend, OTP Busy, BEFP modes) 2. Suspend states (Erase Suspend, Pgm Suspend, Pgm Suspend In Erase Suspend).
- BEFP writes are only allowed when the Status Register bit #0 = 0 or else the data is ignored.
- Confirm commands (Lock Block, Unlock Block, Lock-Down Block, Configuration Register and Blank Check) perform the operation and then move to the Ready State.
- Buffered programming will botch when a different block address (as compared to the address given on the first data write cycle) is written during the BP Load1 and BP Load2 states.
- All two cycle commands will be considered as a contiguous whole during device suspend states. Individual commands will not be parsed separately. (I.e. If an erase set-up command is issued followed by a D0h command, the D0h command will not resume the program operation. Issuing the erase set-up places the CUI in an "illegal state". A subsequent command will clear the "illegal state", but the command will be otherwise ignored.

## Appendix B Conventions - Additional Documentation

### B.1 Acronyms

|        |                                     |
|--------|-------------------------------------|
| BEFP : | Buffer Enhanced Factory Programming |
| CUI :  | Command User Interface              |
| CFI :  | Common Flash Interface              |
| EFI :  | Extended Function Interface         |
| SBC :  | Single Bit per Cell                 |
| OTP :  | One-Time Programmable               |
| PLR :  | one-time programmable Lock Register |
| PR :   | one-time programmable Register      |
| RCR :  | Read Configuration Register         |
| RFU :  | Reserved for Future Use             |
| SR :   | Status Register                     |
| SRD :  | Status Register Data                |
| WSM :  | Write State Machine                 |

### B.2 Definitions and Terms

|                   |  |
|-------------------|--|
| VCC :             | Signal or voltage connection   |
| V <sub>CC</sub> : | Signal or voltage level  |
| h :               | Hexadecimal number suffix  |
| 0b :              | Binary number prefix   |
| 0x :              | Exadecimal number prefix   |
| SR.4 :            | Denotes an individual register bit.  |
| SR[3,1] :         | Denotes a group individual register bits.  |
| SR[3:1] :         | Denotes a group continuous register bits.  |
| 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 :             | Single 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  |
| K :               | 1,000  |

- M : 1,000,000
- Block : A group of bits, bytes, or words within the flash memory array that erase simultaneously.
- Array block : An array block that is usually used to store code and/or data.

## Appendix C Revision History

| Date     | Revision | Description   |
|----------|----------|---|
| Jun 2009 | 01       | Initial release   |
| Apr 2010 | 02       | <p>Update the buffered program performance, suspend latency, BEFP performance in <a href="#">Table 27, "Program and Erase Specifications" on page 58</a>.</p> <p>Update the 40Mhz spec for TSOP package in <a href="#">Table 25, "AC Read Specifications" on page 50</a>.</p> <p>Add <math>t_{DVWH}</math> timing comments in <a href="#">Table 26, "AC Write Specifications" on page 54</a>.</p> <p>Reflect the program performance in CFI in <a href="#">Table 34, "System Interface Information" on page 64</a>.</p> <p>Update the URL for part number lookup.</p> |



## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- ⊖ [View JS28F640P30BF75A on WIN SOURCE](#)
- ⊖ [Micron Technology Inc. Information](#)

## Optimize Your Supply Chain with WIN SOURCE Solutions

- ✓ Global Sourcing Solution
- ✓ Obsolete Management
- ✓ Cost Control Management
- ✓ Shortage Management
- ✓ Alternative Solution
- ✓ Excess Inventory Management