



# Intel® Advanced+ Boot Block Flash Memory (C3)

28F800C3, 28F160C3, 28F320C3 (x16)

Datasheet

## Product Features

- **Flexible SmartVoltage Technology**
  - 2.7 V– 3.6 V read/program/erase
  - 12 V for fast production programming
- **1.65 V to 2.5 V or 2.7 V to 3.6 V I/O Option**
  - Reduces overall system power
- **High Performance**
  - 2.7 V– 3.6 V: 70 ns max access time
- **Optimized Architecture for Code Plus Data Storage**
  - Eight 4 Kword blocks, top or bottom parameter boot
  - Up to 127 x 32 Kword blocks
  - Fast program suspend capability
  - Fast erase suspend capability
- **Flexible Block Locking**
  - Lock/unlock any block
  - Full protection on power-up
  - Write Protect (WP#) pin for hardware block protection
- **Low Power Consumption**
  - 9 mA typical read
  - 7 uA typical standby with Automatic Power Savings feature
- **Extended Temperature Operation**
  - -40 °C to +85 °C
- **128-bit Protection Register**
  - 64 bit unique device identifier
  - 64 bit user programmable OTP cells
- **Extended Cycling Capability**
  - Minimum 100,000 block erase cycles
- **Software**
  - Supported by Intel's Advanced Flash File Managers -- Intel® VFM, Intel® FDI, etc.
  - Code and data storage in the same memory device
  - Robust Power Loss Recovery for Data Loss Prevention
  - Common Flash Interface
  - <http://www.intel.com/go/flashsw>
- **Standard Surface Mount Packaging**
  - 48-Ball  $\mu$ BGA\*/VFBGA
  - 64-Ball Easy BGA packages
  - 48-TSOP package
- **ETOX™ VIII (0.13  $\mu$ m) Flash Technology**
  - 8, 16, 32 Mbit
- **ETOX™ VII (0.18  $\mu$ m) Flash Technology**
  - 16, 32 Mbit
- **ETOX™ VI (0.25  $\mu$ m) Flash Technology**
  - 8, 16 and 32 Mbit

The Intel® Advanced+ Boot Block Flash Memory (C3) device, manufactured on Intel's latest 0.13  $\mu$ m and 0.18  $\mu$ m technologies, represents a feature-rich solution for low-power applications. The C3 device incorporates low-voltage capability (3 V read, program, and erase) with high-speed, low-power operation. Flexible block locking allows any block to be independently locked or unlocked. Add to this the Intel® Flash Data Integrator (Intel® FDI) software and you have a cost-effective, flexible, monolithic code plus data storage solution. Intel® Advanced+ Boot Block Flash Memory (C3) products are available in 48-lead TSOP, 48-ball CSP, and 64-ball Easy BGA packages. Additional information on this product family can be obtained from the Intel® Flash website: <http://www.intel.com/design/flash>.



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

Intel may make changes to specifications and product descriptions at any time, without notice.

Intel Corporation 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.

Contact your local Intel 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 Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

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

Copyright © 2005, Intel Corporation. All Rights Reserved.

# Contents

---

|            |   |    |
|------------|---|----|
| <b>1.0</b> | <b>Introduction</b> .....                             | 7  |
| 1.1        | Nomenclature .....                                    | 7  |
| 1.2        | Conventions.....                                      | 7  |
| <b>2.0</b> | <b>Functional Overview</b> .....                      | 8  |
| 2.1        | Product Overview .....                                | 8  |
| 2.2        | Block Diagram .....                                   | 9  |
| 2.3        | Memory Map.....                                       | 9  |
| <b>3.0</b> | <b>Package Information</b> .....                      | 12 |
| 3.1        | mBGA* and VF BGA Package.....                         | 12 |
| 3.2        | TSOP Package.....                                     | 13 |
| 3.3        | Easy BGA Package .....                                | 14 |
| <b>4.0</b> | <b>Ballout and Signal Descriptions</b> .....          | 15 |
| 4.1        | 48-Lead TSOP Package.....                             | 15 |
| 4.2        | 64-Ball Easy BGA Package .....                        | 18 |
| 4.3        | Signal Descriptions .....                             | 18 |
| <b>5.0</b> | <b>Maximum Ratings and Operating Conditions</b> ..... | 20 |
| 5.1        | Absolute Maximum Ratings .....                        | 20 |
| 5.2        | Operating Conditions .....                            | 20 |
| <b>6.0</b> | <b>Electrical Specifications</b> .....                | 22 |
| 6.1        | Current Characteristics .....                         | 22 |
| 6.2        | DC Voltage Characteristics.....                       | 24 |
| <b>7.0</b> | <b>AC Characteristics</b> .....                       | 25 |
| 7.1        | AC Read Characteristics .....                         | 25 |
| 7.2        | AC Write Characteristics.....                         | 29 |
| 7.3        | Erase and Program Timings .....                       | 33 |
| 7.4        | AC I/O Test Conditions .....                          | 33 |
| 7.5        | Device Capacitance.....                               | 34 |
| <b>8.0</b> | <b>Power and Reset Specifications</b> .....           | 35 |
| 8.1        | Active Power (Program/Erase/Read).....                | 35 |
| 8.2        | Automatic Power Savings (APS) .....                   | 35 |
| 8.3        | Standby Power .....                                   | 35 |
| 8.4        | Deep Power-Down Mode.....                             | 35 |
| 8.5        | Power and Reset Considerations .....                  | 36 |
| 8.5.1      | Power-Up/Down Characteristics .....                   | 36 |
| 8.5.2      | RP# Connected to System Reset .....                   | 36 |
| 8.5.3      | VCC, VPP and RP# Transitions .....                    | 36 |
| 8.5.4      | Reset Specifications .....                            | 37 |
| 8.6        | Power Supply Decoupling.....                          | 37 |
| <b>9.0</b> | <b>Device Operations</b> .....                        | 39 |

|             |  |           |
|-------------|--|-----------|
| 9.1         | Bus Operations .....                             | 39        |
| 9.1.1       | Read .....                                       | 39        |
| 9.1.2       | Write .....                                      | 39        |
| 9.1.3       | Output Disable .....                             | 39        |
| 9.1.4       | Standby.....                                     | 40        |
| 9.1.5       | Reset .....                                      | 40        |
| <b>10.0</b> | <b>Modes of Operation.....</b>                   | <b>41</b> |
| 10.1        | Read Mode .....                                  | 41        |
| 10.1.1      | Read Array.....                                  | 41        |
| 10.1.2      | Read Identifier .....                            | 41        |
| 10.1.3      | CFI Query .....                                  | 42        |
| 10.1.4      | Read Status Register.....                        | 42        |
| 10.1.4.1    | Clear Status Register .....                      | 43        |
| 10.2        | Program Mode.....                                | 43        |
| 10.2.1      | 12-Volt Production Programming.....              | 43        |
| 10.2.2      | Suspending and Resuming Program.....             | 44        |
| 10.3        | Erase Mode .....                                 | 44        |
| 10.3.1      | Suspending and Resuming Erase .....              | 45        |
| <b>11.0</b> | <b>Security Modes .....</b>                      | <b>49</b> |
| 11.1        | Flexible Block Locking .....                     | 49        |
| 11.1.1      | Locking Operation.....                           | 50        |
| 11.1.1.1    | Locked State .....                               | 50        |
| 11.1.1.2    | Unlocked State.....                              | 50        |
| 11.1.1.3    | Lock-Down State.....                             | 50        |
| 11.2        | Reading Block-Lock Status.....                   | 50        |
| 11.3        | Locking Operations during Erase Suspend .....    | 51        |
| 11.4        | Status Register Error Checking .....             | 51        |
| 11.5        | 128-Bit Protection Register.....                 | 51        |
| 11.5.1      | Reading the Protection Register.....             | 52        |
| 11.5.2      | Programming the Protection Register.....         | 52        |
| 11.5.3      | Locking the Protection Register.....             | 52        |
| 11.6        | V <sub>PP</sub> Program and Erase Voltages ..... | 52        |
| 11.6.1      | Program Protection.....                          | 53        |
| <b>A</b>    | <b>Write State Machine States.....</b>           | <b>54</b> |
| <b>B</b>    | <b>Flow Charts.....</b>                          | <b>56</b> |
| <b>C</b>    | <b>Common Flash Interface.....</b>               | <b>62</b> |
| C.1         | Query Structure Output.....                      | 62        |
| C.2         | Query Structure Overview .....                   | 63        |
| C.3         | Block Status Register .....                      | 64        |
| C.4         | CFI Query Identification String.....             | 65        |
| C.5         | Device Geometry Definition.....                  | 66        |
| C.6         | Intel-Specific Extended Query Table .....        | 68        |
| <b>D</b>    | <b>Additional Information .....</b>              | <b>70</b> |
| <b>E</b>    | <b>Ordering Information.....</b>                 | <b>71</b> |

# Revision History

| Date of Revision | Version | Description  |
|------------------|---------|--|
| 05/12/98         | -001    | Original version   |
| 07/21/98         | -002    | 48-Lead TSOP package diagram change<br>μBGA package diagrams change<br>32-Mbit ordering information change (Section 6)<br>CFI Query Structure Output Table Change (Table C2)<br>CFI Primary-Vendor Specific Extended Query Table Change for Optional Features and Command Support change (Table C8)<br>Protection Register Address Change<br>I <sub>PPD</sub> test conditions clarification (Section 4.3)<br>μBGA package top side mark information clarification (Section 6)        |
| 10/03/98         | -003    | Byte-Wide Protection Register Address change<br>V <sub>IH</sub> Specification change (Section 4.3)<br>V <sub>IL</sub> Maximum Specification change (Section 4.3)<br>I <sub>CCS</sub> test conditions clarification (Section 4.3)<br>Added Command Sequence Error Note (Table 7)<br>Datasheet renamed from <i>3 Volt Advanced Boot Block, 8-, 16-, 32-Mbit Flash Memory Family</i> .  |
| 12/04/98         | -004    | Added t <sub>BHWH</sub> /t <sub>BHEH</sub> and t <sub>QVBL</sub> (Section 4.6)<br>Programming the Protection Register clarification (Section 3.4.2)  |
| 12/31/98         | -005    | Removed all references to x8 configurations  |
| 02/24/99         | -006    | Removed reference to 40-Lead TSOP from front page  |
| 06/10/99         | -007    | Added Easy BGA package (Section 1.2)<br>Removed 1.8 V I/O references<br><i>Locking Operations Flowchart</i> changed (Appendix B)<br>Added t <sub>WHGL</sub> (Section 4.6)<br>CFI Primary Vendor-Specific Extended Query changed (Appendix C)   |
| 03/20/00         | -008    | Max I <sub>CCD</sub> changed to 25 μA<br>Table 10, added note indicating V <sub>CCMax</sub> = 3.3 V for 32-Mbit device   |
| 04/24/00         | -009    | Added specifications for 0.18 micron product offerings throughout document<br>Added 64-Mbit density  |
| 10/12/00         | -010    | Changed references of 32Mbit 80ns devices to 70ns devices to reflect the faster product offering.<br>Changed V <sub>ccMax</sub> =3.3V reference to indicate that the affected product is the 0.25μm 32Mbit device.<br>Minor text edits throughout document.  |
| 7/20/01          | -011    | Added 1.8v I/O operation documentation where applicable<br>Added TSOP PCN 'Pin-1' indicator information<br>Changed references in 8 x 8 BGA pinout diagrams from 'GND' to 'Vssq'<br>Added 'Vssq' to Pin Descriptions Information<br>Removed 0.4 μm references in DC characteristics table<br>Corrected 64Mb package Ordering Information from 48-μBGA to 48-VFBGA<br>Corrected 'bottom' parameter block sizes to on 8Mb device to 8 x 4KWords<br>Minor text edits throughout document |
| 10/02/01         | -012    | Added specifications for 0.13 micron product offerings throughout document   |
| 2/05/02          | -013    | Corrected I <sub>ccw</sub> / I <sub>ppw</sub> / I <sub>cces</sub> / I <sub>ppes</sub> values.<br>Added mechanicals for 16Mb and 64Mb<br>Minor text edits throughout document.  |

| Date of Revision | Version | Description  |
|------------------|---------|--|
| 4/05/02          | -014    | Updated 64Mb product offerings.<br>Updated 16Mb product offerings.<br>Revised and corrected DC Characteristics Table.<br>Added mechanicals for Easy BGA.<br>Minor text edits throughout document.  |
| 3/06/03          | -016    | Complete technical update.   |
| 10/01/03         | -017    | Corrected information in the Device Geometry Details table, address 0x34.  |
| 5/20/04          | -018    | Updated the layout of the datasheet.   |
| 9/1/04           | -019    | Fixed typo for Standby power on cover page.  |
| 9/14/04          | -020    | Added lead-free line items to <a href="#">Table 37 "Product Information Ordering Matrix" on page 72.</a>   |
| 9/27/04          | -021    | Added specification for 8Mb 0.13 micron device.<br>Added 0.13 micron to <a href="#">Table 37 "Product Information Ordering Matrix" on page 72.</a>   |
| 1/26/05          | -022    | Converted datasheet to new template. Deleted Description in Table 4. Deleted Note in Figure 5.   |
| 5/16/05          | -023    | Removed all 64M ordering information, removed VF BGA 8M ordering information.<br>Removed 64M reference in title page only. Added software verbiage in title page. Corrected Lead Width (b) measurement in Fig 2., uBGA and VF BGA Package Drawing and Dimensions, page 12. |

## 1.0 Introduction

---

This datasheet contains the specifications for the Intel® Advanced+ Boot Block Flash Memory (C3) device family, hereafter called the C3 flash memory device. These flash memories add features such as instant block locking and protection registers that can be used to enhance the security of systems.

### 1.1 Nomenclature

|                    |                          |
|--------------------|--------------------------|
| <b>0x</b>          | Hexadecimal prefix       |
| <b>0b</b>          | Binary prefix            |
| <b>Byte</b>        | 8 bits                   |
| <b>Word</b>        | 16 bits                  |
| <b>KW or Kword</b> | 1024 words               |
| <b>Mword</b>       | 1,048,576 words          |
| <b>Kb</b>          | 1024 bits                |
| <b>KB</b>          | 1024 bytes               |
| <b>Mb</b>          | 1,048,576 bits           |
| <b>MB</b>          | 1,048,576 bytes          |
| <b>APS</b>         | Automatic Power Savings  |
| <b>CSP</b>         | Chip Scale Package       |
| <b>CUI</b>         | Command User Interface   |
| <b>OTP</b>         | One Time Programmable    |
| <b>PR</b>          | Protection Register      |
| <b>PRD</b>         | Protection Register Data |
| <b>PLR</b>         | Protection Lock Register |
| <b>RFU</b>         | Reserved for Future Use  |
| <b>SR</b>          | Status Register          |
| <b>SRD</b>         | Status Register Data     |
| <b>WSM</b>         | Write State Machine      |

### 1.2 Conventions

The terms **pin** and **signal** are often used interchangeably to refer to the external signal connections on the package; for chip scale package (CSP) the term *ball* is used.

**Group Membership Brackets:** Square brackets will be used to designate group membership or to define a group of signals with similar function (i.e. A[21:1], SR[4:1])

**Set:** When referring to registers, the term set means the bit is a logical 1.

**Clear:** When referring to registers, the term clear means the bit is a logical 0.

**Block:** A group of bits (or words) that erase simultaneously with one block erase instruction.

**Main Block:** A block that contains 32 Kwords.

**Parameter Block:** A block that contains 4 Kwords.

## 2.0 Functional Overview

---

This section provides an overview of the Intel® Advanced+ Boot Block Flash Memory (C3) device features and architecture.

### 2.1 Product Overview

The C3 flash memory device provides high-performance asynchronous reads in package-compatible densities with a 16 bit data bus. Individually-erasable memory blocks are optimally sized for code and data storage. Eight 4 Kword parameter blocks are located in the boot block at either the top or bottom of the device's memory map. The rest of the memory array is grouped into 32 Kword main blocks.

The device supports read-array mode operations at various I/O voltages (1.8 V and 3 V) and erase and program operations at 3 V or 12 V VPP. With the 3 V I/O option, VCC and VPP can be tied together for a simple, ultra-low-power design. In addition to I/O voltage flexibility, the dedicated VPP input provides complete data protection when  $V_{PP} \leq V_{PPLK}$ .

The Intel® Advanced+ Boot Block Flash Memory (C3) device features a 128-bit protection register enabling security techniques and data protection schemes through a combination of factory-programmed and user-programmable OTP data registers. Zero-latency locking/unlocking on any memory block provides instant and complete protection for critical system code and data. Additional block lock-down capability provides hardware protection where software commands alone cannot change the block's protection status.

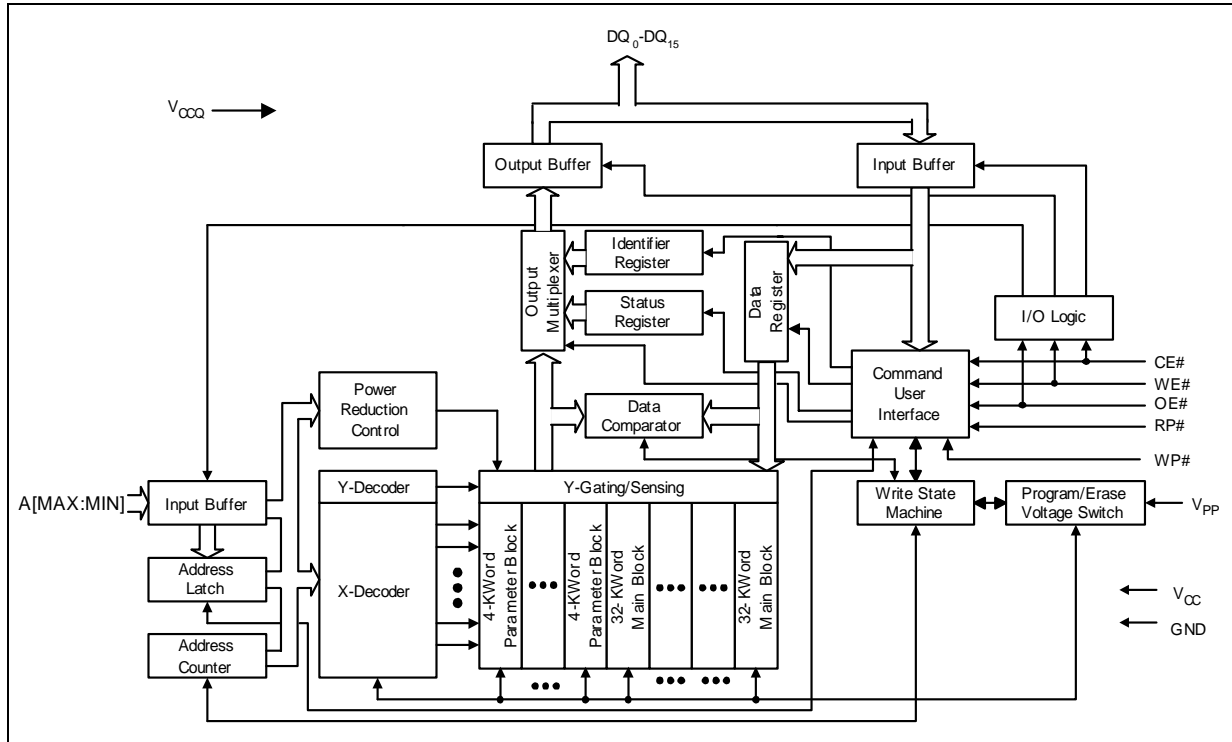
A command User Interface (CUI) serves as the interface between the system processor and internal operation of the device. A valid command sequence issued to the CUI initiates device automation. An internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for block erase, program, and lock-bit configuration operations.

The device offers three low-power saving features: Automatic Power Savings (APS), standby mode, and deep power-down mode. The device automatically enters APS mode following read cycle completion. Standby mode begins when the system deselects the flash memory by deasserting Chip Enable, CE#. The deep power-down mode begins when Reset Deep Power-Down, RP# is asserted, which deselects the memory and places the outputs in a high-impedance state, producing ultra-low power savings. Combined, these three power-savings features significantly enhanced power consumption flexibility.



## 2.2 Block Diagram

Figure 1. C3 Flash Memory Device Block Diagram



## 2.3 Memory Map

The Intel® Advanced+ Boot Block Flash Memory (C3) device is asymmetrically blocked, which enables system code and data integration within a single flash device. The bulk of the array is divided into 32 Kword main blocks that can store code or data, and 4 Kword boot blocks to facilitate storage of boot code or for frequently changing small parameters. See [Table 1, “Top Boot Memory Map”](#) on page 10 and [Table 2, “Bottom Boot Memory Map”](#) on page 11 for details.

**Table 1. Top Boot Memory Map**

| Size (KW) | Blk | 8-Mbit Memory Addressing (Hex) | Size (KW) | Blk | 16-Mbit Memory Addressing (Hex) | Size (KW) | Blk | 32-Mbit Memory Addressing (Hex) | Size (KW) | Blk | 64-Mbit Memory Addressing (Hex) |
|-----------|-----|--------------------------------|-----------|-----|---------------------------------|-----------|-----|---------------------------------|-----------|-----|---------------------------------|
| 4         | 22  | 7F000-7FFFF                    | 4         | 38  | FF000-FFFFF                     | 4         | 70  | 1FF000-1FFFFF                   | 4         | 134 | 3FF000-3FFFFF                   |
| 4         | 21  | 7E000-7EFFF                    | 4         | 37  | FE000-FEFFF                     | 4         | 69  | 1FE000-1FEFFF                   | 4         | 133 | 3FE000-3FEFFF                   |
| 4         | 20  | 7D000-7DFFF                    | 4         | 36  | FD000-FDFFF                     | 4         | 68  | 1FD000-1FDFFF                   | 4         | 132 | 3FD000-3FDFFF                   |
| 4         | 19  | 7C000-7CFFF                    | 4         | 35  | FC000-FCFFF                     | 4         | 67  | 1FC000-1FCFFF                   | 4         | 131 | 3FC000-3FCFFF                   |
| 4         | 18  | 7B000-7BFFF                    | 4         | 34  | FB000-FBFFF                     | 4         | 66  | 1FB000-1FBFFF                   | 4         | 130 | 3FB000-3FBFFF                   |
| 4         | 17  | 7A000-7AFFF                    | 4         | 33  | FA000-FAFFF                     | 4         | 65  | 1FA000-1FAFFF                   | 4         | 129 | 3FA000-3FAFFF                   |
| 4         | 16  | 79000-79FFF                    | 4         | 32  | F9000-F9FFF                     | 4         | 64  | 1F9000-1F9FFF                   | 4         | 128 | 3F9000-3F9FFF                   |
| 4         | 15  | 78000-78FFF                    | 4         | 31  | F8000-F8FFF                     | 4         | 63  | 1F8000-1F8FFF                   | 4         | 127 | 3F8000-3F8FFF                   |
| 32        | 14  | 70000-77FFF                    | 32        | 30  | F0000-F7FFF                     | 32        | 62  | 1F0000-1F7FFF                   | 32        | 126 | 3F0000-3F7FFF                   |
| 32        | 13  | 68000-6FFFF                    | 32        | 29  | E8000-EFFFF                     | 32        | 61  | 1E8000-1EFFFF                   | 32        | 125 | 3E8000-3EFFFF                   |
| 32        | 12  | 60000-67FFF                    | 32        | 28  | E0000-E7FFF                     | 32        | 60  | 1E0000-1E7FFF                   | 32        | 124 | 3E0000-3E7FFF                   |
| 32        | 11  | 58000-5FFFF                    | 32        | 27  | D8000-DFFFF                     | 32        | 59  | 1D8000-1DFFFF                   | 32        | 123 | 3D8000-3DFFFF                   |
| ...       | ... | ...                            | ...       | ... | ...                             | ...       | ... | ...                             | ...       | ... | ...                             |
| 32        | 2   | 10000-17FFF                    | 32        | 2   | 10000-17FFF                     | 32        | 2   | 10000-17FFF                     | 32        | 2   | 10000-17FFF                     |
| 32        | 1   | 8000-0FFFF                     | 32        | 1   | 08000-0FFFF                     | 32        | 1   | 08000-0FFFF                     | 32        | 1   | 08000-0FFFF                     |
| 32        | 0   | 0000-07FFF                     | 32        | 0   | 00000-07FFF                     | 32        | 0   | 00000-07FFF                     | 32        | 0   | 00000-07FFF                     |



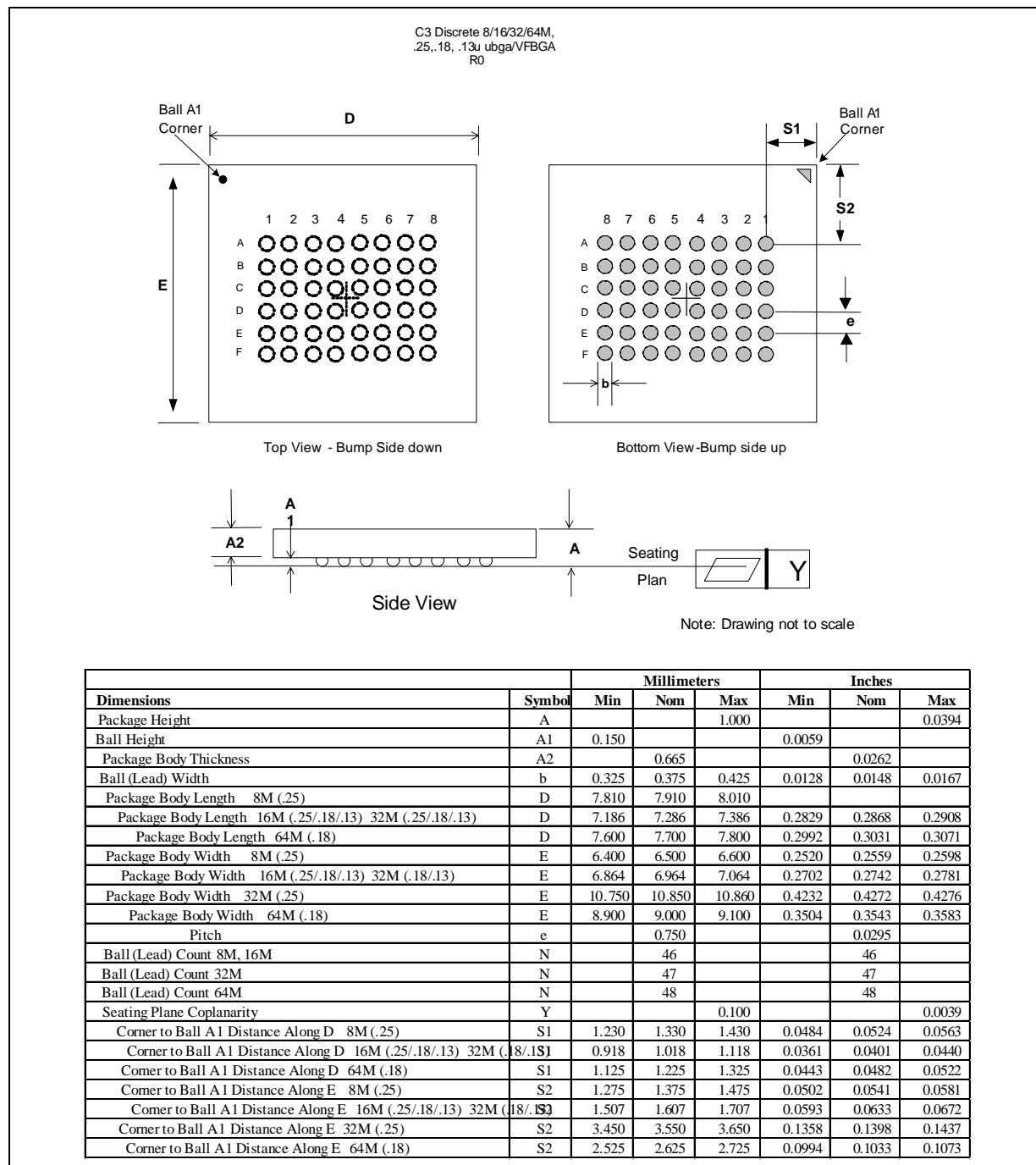
**Table 2. Bottom Boot Memory Map**

| Size (KW) | Blk | 8-Mbit Memory Addressing (Hex) | Size (KW) | Blk | 16-Mbit Memory Addressing (Hex) | Size (KW) | Blk | 32-Mbit Memory Addressing (Hex) | Size (KW) | Blk | 64-Mbit Memory Addressing (Hex) |
|-----------|-----|--------------------------------|-----------|-----|---------------------------------|-----------|-----|---------------------------------|-----------|-----|---------------------------------|
| 32        | 22  | 78000-7FFFF                    | 32        | 38  | F8000-FFFFF                     | 32        | 70  | 1F8000-1FFFFFF                  | 32        | 134 | 3F8000-3FFFFFF                  |
| 32        | 21  | 70000-77FFF                    | 32        | 37  | F0000-F7FFF                     | 32        | 69  | 1F0000-1F7FFF                   | 32        | 133 | 3F0000-3F7FFF                   |
| 32        | 20  | 68000-6FFFF                    | 32        | 36  | E8000-EFFFF                     | 32        | 68  | 1E8000-1EFFFF                   | 32        | 132 | 3E8000-3EFFFF                   |
| 32        | 19  | 60000-67FFF                    | 32        | 35  | E0000-E7FFF                     | 32        | 67  | 1E0000-1E7FFF                   | 32        | 131 | 3E0000-3E7FFF                   |
| ...       | ... | ...                            | ...       | ... | ...                             | ...       | ... | ...                             | ...       | ... | ...                             |
| 32        | 10  | 18000-1FFFF                    | 32        | 10  | 18000-1FFFF                     | 32        | 10  | 18000-1FFFF                     | 32        | 10  | 18000-1FFFF                     |
| 32        | 9   | 10000-17FFF                    | 32        | 9   | 10000-17FFF                     | 32        | 9   | 10000-17FFF                     | 32        | 9   | 10000-17FFF                     |
| 32        | 8   | 08000-0FFFF                    | 32        | 8   | 08000-0FFFF                     | 32        | 8   | 08000-0FFFF                     | 32        | 8   | 08000-0FFFF                     |
| 4         | 7   | 07000-07FFF                    | 4         | 7   | 07000-07FFF                     | 4         | 7   | 07000-07FFF                     | 4         | 7   | 07000-07FFF                     |
| 4         | 6   | 06000-06FFF                    | 4         | 6   | 06000-06FFF                     | 4         | 6   | 06000-06FFF                     | 4         | 6   | 06000-06FFF                     |
| 4         | 5   | 05000-05FFF                    | 4         | 5   | 05000-05FFF                     | 4         | 5   | 05000-05FFF                     | 4         | 5   | 05000-05FFF                     |
| 4         | 4   | 04000-04FFF                    | 4         | 4   | 04000-04FFF                     | 4         | 4   | 04000-04FFF                     | 4         | 4   | 04000-04FFF                     |
| 4         | 3   | 03000-03FFF                    | 4         | 3   | 03000-03FFF                     | 4         | 3   | 03000-03FFF                     | 4         | 3   | 03000-03FFF                     |
| 4         | 2   | 02000-02FFF                    | 4         | 2   | 02000-02FFF                     | 4         | 2   | 02000-02FFF                     | 4         | 2   | 02000-02FFF                     |
| 4         | 1   | 01000-01FFF                    | 4         | 1   | 01000-01FFF                     | 4         | 1   | 01000-01FFF                     | 4         | 1   | 01000-01FFF                     |
| 4         | 0   | 00000-00FFF                    | 4         | 0   | 00000-00FFF                     | 4         | 0   | 00000-00FFF                     | 4         | 0   | 00000-00FFF                     |

## 3.0 Package Information

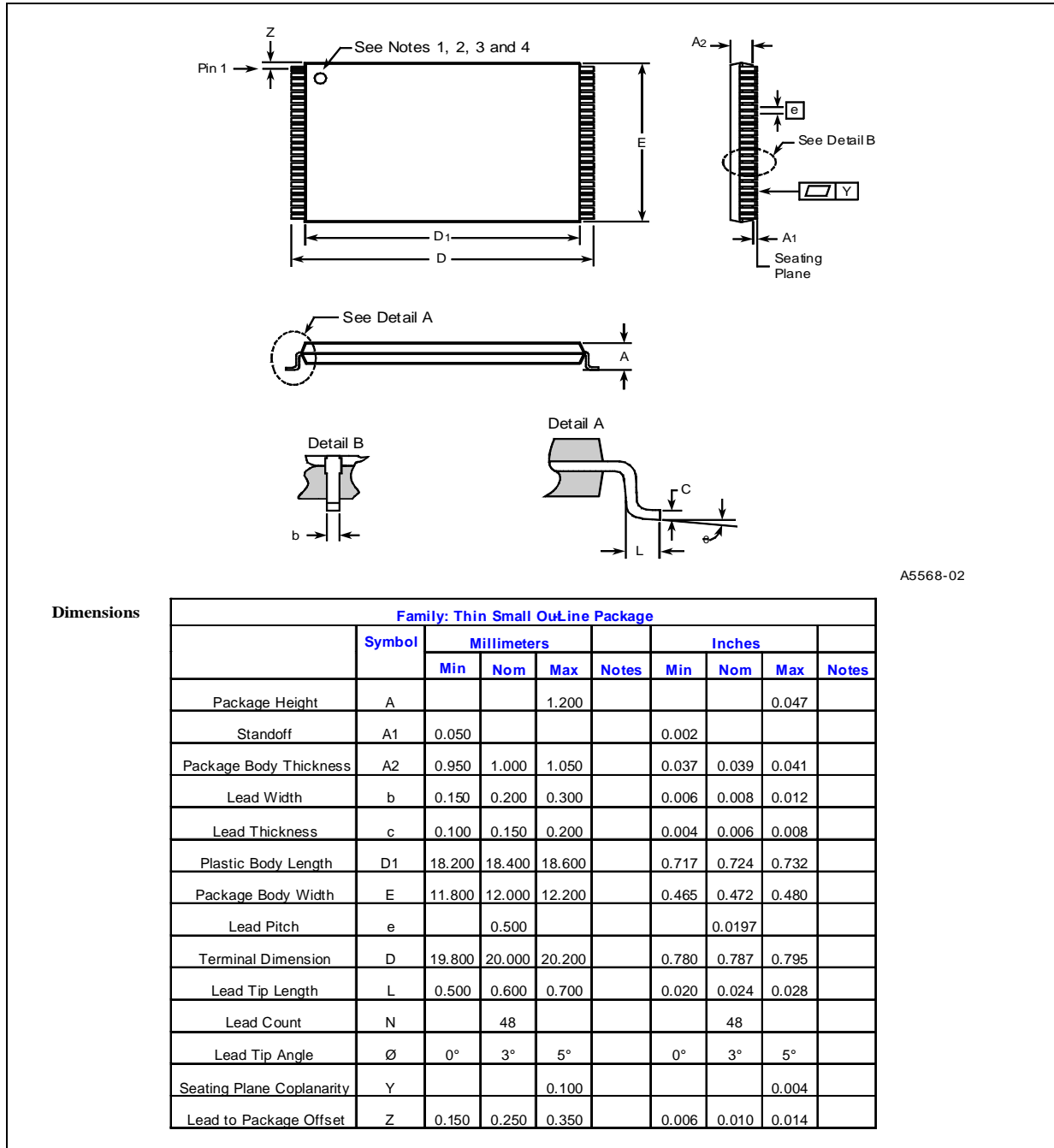
### 3.1 μBGA\* and VF BGA Package

Figure 2. μBGA\* and VF BGA Package Drawing and Dimensions



### 3.2 TSOP Package

Figure 3. TSOP Package Drawing and Dimensions

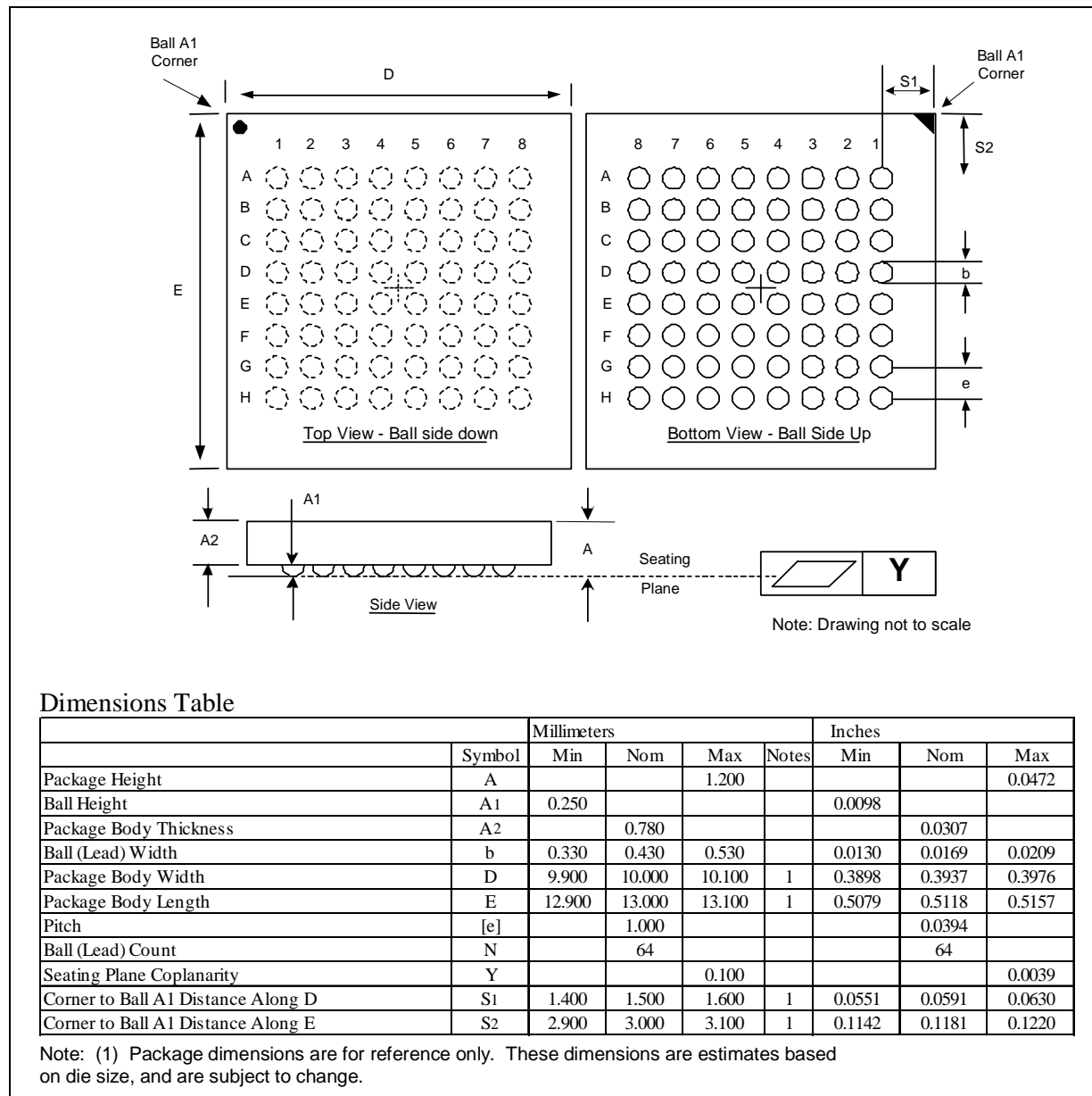


**Notes:**

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.

### 3.3 Easy BGA Package

Figure 4. Easy BGA Package Drawing and Dimension

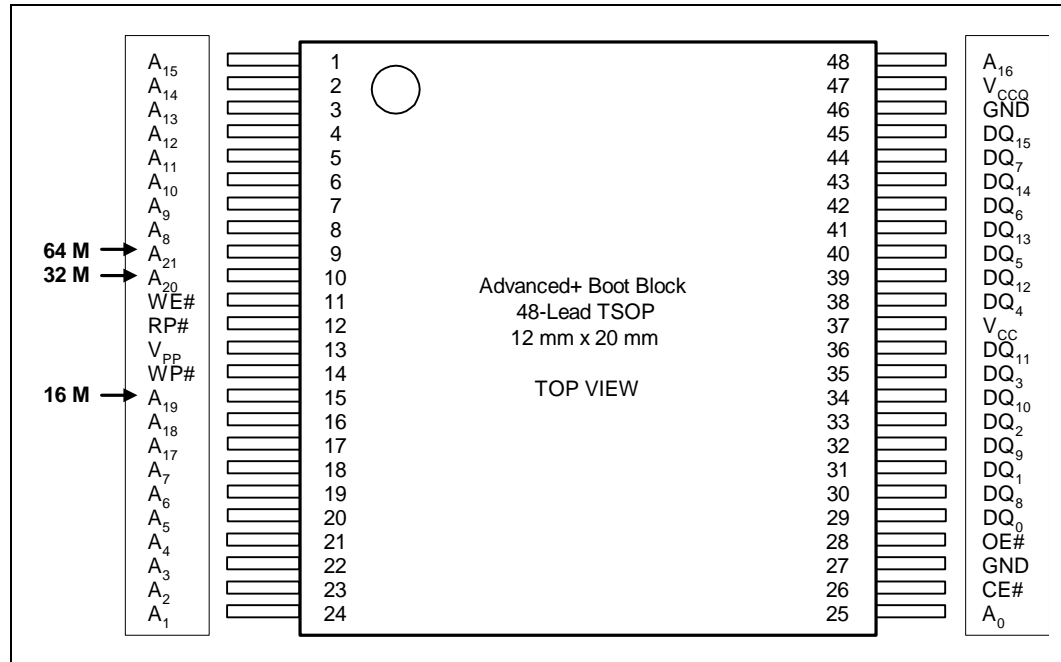


## 4.0 Ballout and Signal Descriptions

The C3 device is available in 48-lead TSOP, 48-ball VF BGA, 48-ball  $\mu$ BGA, and Easy BGA packages. See Figure 5 on page 15, Figure 7 on page 17, and Figure 8 on page 18, respectively.

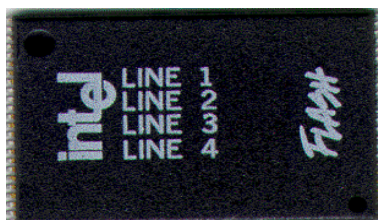
### 4.1 48-Lead TSOP Package

Figure 5. 48-Lead TSOP Package

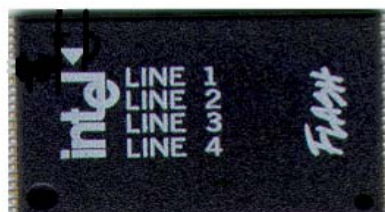


**Figure 6. Mark for Pin-1 Indicator on 48-Lead 8-Mb, 16-Mb and 32-Mb TSOP**

Current Mark:



New Mark:



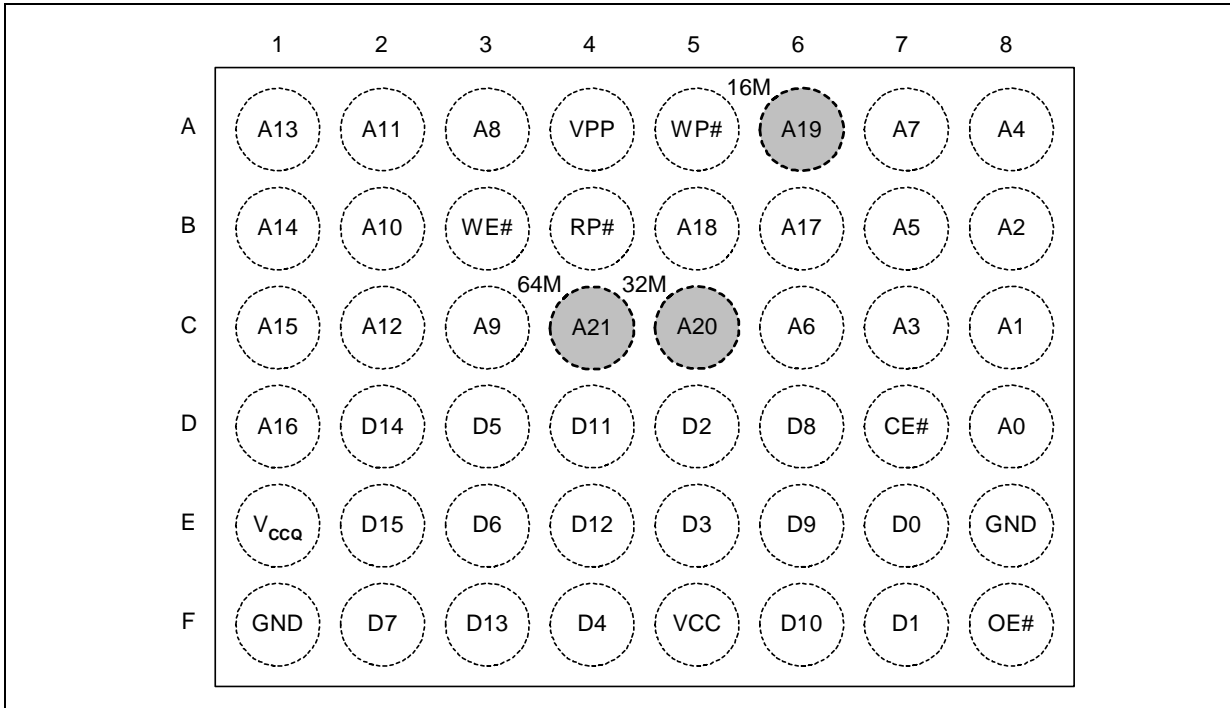
*Note:* The topside marking on 8 Mb, 16 Mb, and 32 Mb Intel® Advanced and Advanced + Boot Block 48L TSOP products will convert to a white ink triangle as a Pin 1 indicator. Products without the white triangle will continue to use a dimple as a Pin 1 indicator. There are no other changes in package size, materials, functionality, customer handling, or manufacturability. Product will continue to meet Intel stringent quality requirements. Products affected are Intel Ordering Codes shown in [Table 3](#).

**Table 3. 48-Lead TSOP**

| Extended 64 Mbit | Extended 32 Mbit                   | Extended 16 Mbit                   | Extended                           |
|------------------|------------------------------------|------------------------------------|------------------------------------|
|                  | TE28F320C3TD70<br>TE28F320C3BD70   | TE28F160C3TD70<br>TE28F160C3BD70   | TE28F800C3TA90<br>TE28F800C3BA90   |
|                  | TE28F320C3TC70<br>TE28F320C3BC70   | TE28F160C3TC80<br>TE28F160C3BC80   | TE28F800C3TA110<br>TE28F800C3BA110 |
|                  | TE28F320C3TC90<br>TE28F320C3BC90   | TE28F160C3TA90<br>TE28F160C3BA90   |                                    |
|                  | TE28F320C3TA100<br>TE28F320C3BA100 | TE28F160C3TA110<br>TE28F160C3BA110 |                                    |
|                  | TE28F320C3TA110<br>TE28F320C3BA110 |                                    |                                    |



Figure 7. 48-Ball µBGA\* and 48-Ball VF BGA Chip Scale Package (Top View, Ball Down)<sup>1,2,3</sup>

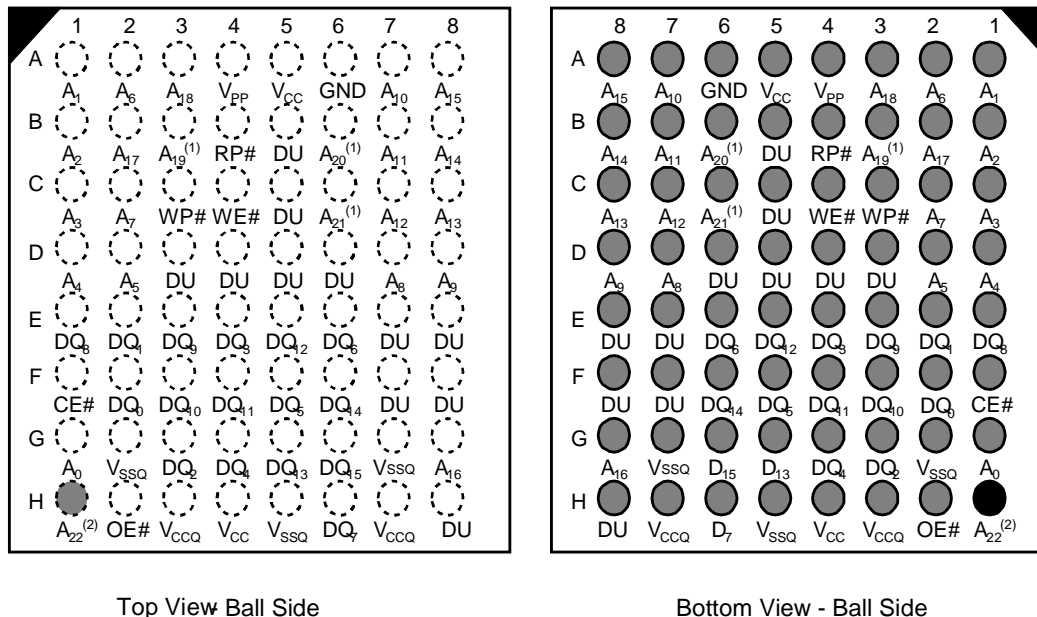


**Notes:**

1. Shaded connections indicate the upgrade address connections. Intel recommends to not use routing in this area.
2. A19 denotes 16 Mbit; A20 denotes 32 Mbit; A21 denotes 64 Mbit.
3. Unused address balls are not populated.

## 4.2 64-Ball Easy BGA Package

Figure 8. 64-Ball Easy BGA Package<sup>1,2</sup>



**Notes:**

1. A19 denotes 16 Mbit; A20 denotes 32 Mbit; A21 denotes 64 Mbit.
2. Unused address balls are not populated.

## 4.3 Signal Descriptions

Table 4. Signal Descriptions

| Symbol            | Type             | Description   |
|-------------------|------------------|---|
| A[ <b>MAX</b> :0] | Input            | <b>ADDRESS INPUTS</b> for memory addresses. Address are internally latched during a program or erase cycle.<br>8 Mbit: AMAX= A18<br>16 Mbit: AMAX = A19<br>32 Mbit: AMAX = A20<br>64 Mbit: AMAX = A21   |
| DQ[15:0]          | Input/<br>Output | <b>DATA INPUTS/OUTPUTS:</b> Inputs data and commands during a write cycle; outputs data during read cycles. Inputs commands to the Command User Interface when CE# and WE# are active. Data is internally latched. The data pins float to tri-state when the chip is de-selected or the outputs are disabled. |
| CE#               | Input            | <b>CHIP ENABLE:</b> Active-low input. Activates the internal control logic, input buffers, decoders and sense amplifiers. CE# is active low. CE# high de-selects the memory device and reduces power consumption to standby levels.   |
| OE#               | Input            | <b>OUTPUT ENABLE:</b> Active-low input. Enables the device's outputs through the data buffers during a Read operation.  |

**Table 4. Signal Descriptions**

| Symbol | Type            | Description   |
|--------|-----------------|---|
| RP#    | Input           | <b>RESET/DEEP POWER-DOWN:</b> Active-low input.<br>When RP# is at logic low, the device is in reset/deep power-down mode, which drives the outputs to High-Z, resets the Write State Machine, and minimizes current levels ( $I_{CCD}$ ).<br>When RP# is at logic high, the device is in standard operation. When RP# transitions from logic-low to logic-high, the device resets all blocks to locked and defaults to the read array mode.   |
| 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 the WE# pulse.   |
| WP#    | Input           | <b>WRITE PROTECT:</b> Active-low input.<br>When WP# is a logic low, the lock-down mechanism is enabled and blocks marked lock-down cannot be unlocked through software.<br>When WP# is logic high, the lock-down mechanism is disabled and blocks previously locked-down are now locked and can be unlocked and locked through software. After WP# goes low, any blocks previously marked lock-down revert to the lock-down state.<br>See <a href="#">Section 11.0, "Security Modes" on page 49</a> for details on block locking.   |
| VPP    | Input/<br>Power | <b>PROGRAM/ERASE Power Supply:</b> Operates as an input at logic levels to control complete device protection. Supplies power for accelerated Program and Erase operations in $12\text{ V} \pm 5\%$ range. Do not leave this pin floating.<br>Lower $VPP \leq VPPLK$ to protect all contents against Program and Erase commands.<br>Set $VPP = VCC$ for in-system Read, Program and Erase operations. In this configuration, VPP can drop as low as 1.65 V to allow for resistor or diode drop from the system supply.<br>Apply VPP to $12\text{ V} \pm 5\%$ for faster program and erase in a production environment. Applying $12\text{ V} \pm 5\%$ to VPP can only be done for a maximum of 1000 cycles on the main blocks and 2500 cycles on the boot blocks. VPP can be connected to 12 V for a total of 80 hours maximum. See <a href="#">Section 11.6</a> for details on VPP voltage configurations. |
| VCC    | Power           | <b>DEVICE CORE Power Supply:</b> Supplies power for device operations.  |
| VCCQ   | Power           | <b>OUTPUT Power Supply:</b> Output-driven source voltage. This ball can be tied directly to $V_{CC}$ if operating within $V_{CC}$ range.  |
| GND    | Power           | <b>Ground:</b> For all internal circuitry. All ground inputs must be connected.   |
| DU     | —               | <b>Do Not Use:</b> Do not use this ball. This ball must not be connected to any power supplies, signals or other balls,; it must be left floating.  |
| NC     | —               | <b>No Connect</b>   |

## 5.0 Maximum Ratings and Operating Conditions

### 5.1 Absolute Maximum Ratings

**Warning:** Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These ratings are stress ratings only. Operation beyond the “Operating Conditions” is not recommended, and extended exposure beyond the “Operating Conditions” may affect device reliability.

**NOTICE:** Specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest datasheet before finalizing a design.

| Parameter  | Maximum Rating    | Notes |
|--|-------------------|-------|
| Extended Operating Temperature   |                   |       |
| During Read  | –40 °C to +85 °C  |       |
| During Block Erase and Program   | –40 °C to +85 °C  |       |
| Temperature under Bias   | –40 °C to +85 °C  |       |
| Storage Temperature  | –65 °C to +125 °C |       |
| Voltage On Any Pin (except $V_{CC}$ and $V_{PP}$ ) with Respect to GND | –0.5 V to +3.7 V  | 1     |
| $V_{PP}$ Voltage (for Block Erase and Program) with Respect to GND     | –0.5 V to +13.5 V | 1,2,3 |
| $V_{CC}$ and $V_{CCQ}$ Supply Voltage with Respect to GND              | –0.2 V to +3.6 V  |       |
| Output Short Circuit Current   | 100 mA            | 4     |

**Notes:**

1. Minimum DC voltage is –0.5 V on input/output pins. During transitions, this level may undershoot to –2.0 V for periods <20 ns. Maximum DC voltage on input/output pins is  $V_{CC} + 0.5$  V which, during transitions, may overshoot to  $V_{CC} + 2.0$  V for periods <20 ns.
2. Maximum DC voltage on  $V_{PP}$  may overshoot to +14.0 V for periods <20 ns.
3.  $V_{PP}$  Program voltage is normally 1.65 V–3.6 V. Connection to a 11.4 V–12.6 V supply can be done for a maximum of 1000 cycles on the main blocks and 2500 cycles on the parameter blocks during program/erase.  $V_{PP}$  may be connected to 12 V for a total of 80 hours maximum.
4. Output shorted for no more than one second. No more than one output shorted at a time.

### 5.2 Operating Conditions

Table 5. Temperature and Voltage Operating Conditions

| Symbol     | Parameter               | Notes | Min  | Max | Units |
|------------|-------------------------|-------|------|-----|-------|
| $T_A$      | Operating Temperature   |       | –40  | +85 | °C    |
| $V_{CC1}$  | $V_{CC}$ Supply Voltage | 1, 2  | 2.7  | 3.6 | Volts |
| $V_{CC2}$  |                         | 1, 2  | 3.0  | 3.6 |       |
| $V_{CCQ1}$ | I/O Supply Voltage      | 1     | 2.7  | 3.6 | Volts |
| $V_{CCQ2}$ |                         |       | 1.65 | 2.5 |       |
| $V_{CCQ3}$ |                         |       | 1.8  | 2.5 |       |
| $V_{PP1}$  | Supply Voltage          | 1     | 1.65 | 3.6 | Volts |

**Table 5. Temperature and Voltage Operating Conditions**

| Symbol           | Parameter           | Notes | Min     | Max  | Units  |
|------------------|---------------------|-------|---------|------|--------|
| V <sub>PP2</sub> |                     | 1, 3  | 11.4    | 12.6 | Volts  |
| Cycling          | Block Erase Cycling | 3     | 100,000 |      | Cycles |

**Notes:**

1. V<sub>CC</sub> and V<sub>CCQ</sub> must share the same supply when they are in the V<sub>CC1</sub> range.
2. V<sub>CC</sub>Max = 3.3 V for 0.25µm 32-Mbit devices.
3. Applying V<sub>PP</sub> = 11.4 V–12.6 V during a program/erase can only be done for a maximum of 1000 cycles on the main blocks and 2500 cycles on the parameter blocks. V<sub>PP</sub> may be connected to 12 V for a total of 80 hours maximum.

## 6.0 Electrical Specifications

### 6.1 Current Characteristics

Table 6. DC Current Characteristics (Sheet 1 of 2)

| Sym               | Parameter   | V <sub>CC</sub>  |     | 2.7 V–3.6 V |     | 2.7 V–2.85 V |     | 2.7 V–3.3 V |      | Unit | Test Conditions  |
|-------------------|---|------------------|-----|-------------|-----|--------------|-----|-------------|------|------|--|
|                   |   | V <sub>CCQ</sub> |     | 2.7 V–3.6 V |     | 1.65 V–2.5 V |     | 1.8 V–2.5 V |      |      |  |
|                   |   | Note             | Typ | Max         | Typ | Max          | Typ | Max         |      |      |  |
| I <sub>LI</sub>   | Input Load Current  | 1,2              |     | ± 1         |     | ± 1          |     |             | ± 1  | μA   | V <sub>CC</sub> = V <sub>CC</sub> Max<br>V <sub>CCQ</sub> = V <sub>CCQ</sub> Max<br>V <sub>IN</sub> = V <sub>CCQ</sub> or GND  |
| I <sub>LO</sub>   | Output Leakage Current  | 1,2              |     | ± 10        |     | ± 10         |     |             | ± 10 | μA   | V <sub>CC</sub> = V <sub>CC</sub> Max<br>V <sub>CCQ</sub> = V <sub>CCQ</sub> Max<br>V <sub>IN</sub> = V <sub>CCQ</sub> or GND  |
| I <sub>CCS</sub>  | V <sub>CC</sub> Standby Current for 0.13 and 0.18 Micron Product    | 1                | 7   | 15          | 20  | 50           | 150 | 250         |      | μA   | V <sub>CC</sub> = V <sub>CC</sub> Max<br>CE# = RP# = V <sub>CCQ</sub><br>or during Program/ Erase Suspend  |
|                   | V <sub>CC</sub> Standby Current for 0.25 Micron Product             | 1                | 10  | 25          | 20  | 50           | 150 | 250         |      | μA   | WP# = V <sub>CCQ</sub> or GND  |
| I <sub>CCD</sub>  | V <sub>CC</sub> Power-Down Current for 0.13 and 0.18 Micron Product | 1,2              | 7   | 15          | 7   | 20           | 7   | 20          |      | μA   | V <sub>CC</sub> = V <sub>CC</sub> Max<br>V <sub>CCQ</sub> = V <sub>CCQ</sub> Max<br>V <sub>IN</sub> = V <sub>CCQ</sub> or GND  |
|                   | V <sub>CC</sub> Power-Down Current for 0.25 Product                 | 1,2              | 7   | 25          | 7   | 25           | 7   | 25          |      | μA   | RP# = GND ± 0.2 V  |
| I <sub>CCR</sub>  | V <sub>CC</sub> Read Current for 0.13 and 0.18 Micron Product       | 1,2,3            | 9   | 18          | 8   | 15           | 9   | 15          |      | mA   | V <sub>CC</sub> = V <sub>CC</sub> Max<br>V <sub>CCQ</sub> = V <sub>CCQ</sub> Max<br>OE# = V <sub>IH</sub> , CE# = V <sub>IL</sub><br>f = 5 MHz, I <sub>OUT</sub> = 0 mA<br>Inputs = V <sub>IL</sub> or V <sub>IH</sub> |
|                   | V <sub>CC</sub> Read Current for 0.25 Micron Product                | 1,2,3            | 10  | 18          | 8   | 15           | 9   | 15          |      | mA   |  |
| I <sub>PPD</sub>  | V <sub>PP</sub> Deep Power-Down Current                             | 1                | 0.2 | 5           | 0.2 | 5            | 0.2 | 5           |      | μA   | RP# = GND ± 0.2 V<br>V <sub>PP</sub> ≤ V <sub>CC</sub>   |
| I <sub>CCW</sub>  | V <sub>CC</sub> Program Current                                     | 1,4              | 18  | 55          | 18  | 55           | 18  | 55          |      | mA   | V <sub>PP</sub> = V <sub>PP1</sub> ,<br>Program in Progress  |
|                   |   |                  | 8   | 22          | 10  | 30           | 10  | 30          |      | mA   | V <sub>PP</sub> = V <sub>PP2</sub> (12V)<br>Program in Progress  |
| I <sub>CC E</sub> | V <sub>CC</sub> Erase Current                                       | 1,4              | 16  | 45          | 21  | 45           | 21  | 45          |      | mA   | V <sub>PP</sub> = V <sub>PP1</sub> ,<br>Erase in Progress  |
|                   |   |                  | 8   | 15          | 16  | 45           | 16  | 45          |      | mA   | V <sub>PP</sub> = V <sub>PP2</sub> (12V),<br>Erase in Progress   |

**Table 6. DC Current Characteristics (Sheet 2 of 2)**

| Sym                                      | Parameter  | V <sub>CC</sub>  |      | 2.7 V–3.6 V |      | 2.7 V–2.85 V |      | 2.7 V–3.3 V |    | Unit  | Test Conditions |
|--|--|------------------|------|-------------|------|--------------|------|-------------|----|---|-----------------|
|  |  | V <sub>CCQ</sub> |      | 2.7 V–3.6 V |      | 1.65 V–2.5 V |      | 1.8 V–2.5 V |    |   |                 |
|  |  | Note             | Typ  | Max         | Typ  | Max          | Typ  | Max         |    |   |                 |
| I <sub>CCES</sub> /<br>I <sub>CCWS</sub> | V <sub>CC</sub> Erase Suspend Current for 0.13 and 0.18 Micron Product | 1,4,5            | 7    | 15          | 50   | 200          | 50   | 200         | μA | CE# = V <sub>IH</sub> , Erase Suspend in Progress                               |                 |
|  | V <sub>CC</sub> Erase Suspend Current for 0.25 Micron Product          |                  | 10   | 25          | 50   | 200          | 50   | 200         | μA |   |                 |
| I <sub>PPR</sub>                         | V <sub>PP</sub> Read Current   | 1,4              | 2    | ±15         | 2    | ±15          | 2    | ±15         | μA | V <sub>PP</sub> ≤ V <sub>CC</sub>   |                 |
|  |  |                  | 50   | 200         | 50   | 200          | 50   | 200         | μA | V <sub>PP</sub> > V <sub>CC</sub>   |                 |
| I <sub>PPW</sub>                         | V <sub>PP</sub> Program Current  | 1,4              | 0.05 | 0.1         | 0.05 | 0.1          | 0.05 | 0.1         | mA | V <sub>PP</sub> = V <sub>PP1</sub> , Program in Progress                        |                 |
|  |  |                  | 8    | 22          | 8    | 22           | 8    | 22          | mA | V <sub>PP</sub> = V <sub>PP2 (12v)</sub> , Program in Progress                  |                 |
| I <sub>PPE</sub>                         | V <sub>PP</sub> Erase Current  | 1,4              | 0.05 | 0.1         | 0.05 | 0.1          | 0.05 | 0.1         | mA | V <sub>PP</sub> = V <sub>PP1</sub> , Erase in Progress                          |                 |
|  |  |                  | 8    | 22          | 16   | 45           | 16   | 45          | mA | V <sub>PP</sub> = V <sub>PP2 (12v)</sub> , Erase in Progress                    |                 |
| I <sub>PPES</sub> /<br>I <sub>PPWS</sub> | V <sub>CC</sub> Erase Suspend Current                                  | 1,4              | 0.2  | 5           | 0.2  | 5            | 0.2  | 5           | μA | V <sub>PP</sub> = V <sub>PP1</sub> , Program or Erase Suspend in Progress       |                 |
|  |  |                  | 50   | 200         | 50   | 200          | 50   | 200         | μA | V <sub>PP</sub> = V <sub>PP2 (12v)</sub> , Program or Erase Suspend in Progress |                 |

**Notes:**

- All currents are in RMS unless otherwise noted. Typical values at nominal V<sub>CC</sub>, T<sub>A</sub> = +25 °C.
- The test conditions V<sub>CC</sub>Max, V<sub>CCQ</sub>Max, V<sub>CC</sub>Min, and V<sub>CCQ</sub>Min refer to the maximum or minimum V<sub>CC</sub> or V<sub>CCQ</sub> voltage listed at the top of each column. V<sub>CC</sub>Max = 3.3 V for 0.25μm 32-Mbit devices.
- Automatic Power Savings (APS) reduces I<sub>CCR</sub> to approximately standby levels in static operation (CMOS inputs).
- Sampled, not 100% tested.
- I<sub>CCES</sub> or I<sub>CCWS</sub> is specified with device de-selected. If device is read while in erase suspend, current draw is sum of I<sub>CCES</sub> and I<sub>CCR</sub>. If the device is read while in program suspend, current draw is the sum of I<sub>CCWS</sub> and I<sub>CCR</sub>.

## 6.2 DC Voltage Characteristics

Table 7. DC Voltage Characteristics

| Sym               | Parameter   | V <sub>CC</sub>  | 2.7 V–3.6 V               |                             | 2.7 V–2.85 V               |                           | 2.7 V–3.3 V                |                           | Unit | Test Conditions   |
|-------------------|---|------------------|---------------------------|-----------------------------|----------------------------|---------------------------|----------------------------|---------------------------|------|---|
|                   |   | V <sub>CCQ</sub> | 2.7 V–3.6 V               |                             | 1.65 V–2.5 V               |                           | 1.8 V–2.5 V                |                           |      |   |
|                   |   | Note             | Min                       | Max                         | Min                        | Max                       | Min                        | Max                       |      |   |
| V <sub>IL</sub>   | Input Low Voltage                                 |                  | –0.4                      | V <sub>CC</sub> *<br>0.22 V | –0.4                       | 0.4                       | –0.4                       | 0.4                       | V    |   |
| V <sub>IH</sub>   | Input High Voltage                                |                  | 2.0                       | V <sub>CCQ</sub><br>+0.3V   | V <sub>CCQ</sub> –<br>0.4V | V <sub>CCQ</sub><br>+0.3V | V <sub>CCQ</sub> –<br>0.4V | V <sub>CCQ</sub><br>+0.3V | V    |   |
| V <sub>OL</sub>   | Output Low Voltage                                |                  | –0.1                      | 0.1                         | –0.1                       | 0.1                       | –0.1                       | 0.1                       | V    | V <sub>CC</sub> = V <sub>CCMin</sub><br>V <sub>CCQ</sub> = V <sub>CCQMin</sub><br>I <sub>OL</sub> = 100 μA  |
| V <sub>OH</sub>   | Output High Voltage                               |                  | V <sub>CCQ</sub><br>–0.1V |                             | V <sub>CCQ</sub> –<br>0.1V |                           | V <sub>CCQ</sub> –<br>0.1V |                           | V    | V <sub>CC</sub> = V <sub>CCMin</sub><br>V <sub>CCQ</sub> = V <sub>CCQMin</sub><br>I <sub>OH</sub> = –100 μA |
| V <sub>PPLK</sub> | V <sub>PP</sub> Lock-Out Voltage                  | 1                |                           | 1.0                         |                            | 1.0                       |                            | 1.0                       | V    | Complete Write Protection   |
| V <sub>PP1</sub>  | V <sub>PP</sub> during Program / Erase Operations | 1                | 1.65                      | 3.6                         | 1.65                       | 3.6                       | 1.65                       | 3.6                       | V    |   |
| V <sub>PP2</sub>  |   | 1,2              | 11.4                      | 12.6                        | 11.4                       | 12.6                      | 11.4                       | 12.6                      | V    |   |
| V <sub>LKO</sub>  | V <sub>CC</sub> Prog/ Erase Lock Voltage          |                  | 1.5                       |                             | 1.5                        |                           | 1.5                        |                           | V    |   |
| V <sub>LKO2</sub> | V <sub>CCQ</sub> Prog/ Erase Lock Voltage         |                  | 1.2                       |                             | 1.2                        |                           | 1.2                        |                           | V    |   |

**Notes:**

1. Erase and Program are inhibited when V<sub>PP</sub> < V<sub>PPLK</sub> and not guaranteed outside the valid V<sub>PP</sub> ranges of V<sub>PP1</sub> and V<sub>PP2</sub>.
2. Applying V<sub>PP</sub> = 11.4 V–12.6 V during program/erase can only be done for a maximum of 1000 cycles on the main blocks and 2500 cycles on the parameter blocks. V<sub>PP</sub> may be connected to 12 V for a total of 80 hours maximum.



## 7.0 AC Characteristics

### 7.1 AC Read Characteristics

**Table 8. Read Operations—8-Mbit Density**

| #   | Sym               | Parameter  | Density         |          | 8 Mbit        |          |               |          |               |          |               |          |               |  |
|-----|-------------------|--|-----------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|--|
|     |                   |  | Product         |          | 70 ns         |          | 90 ns         |          |               |          | 110 ns        |          |               |  |
|     |                   |  | V <sub>CC</sub> |          | 2.7 V – 3.6 V |          | 3.0 V – 3.6 V |          | 2.7 V – 3.6 V |          | 3.0 V – 3.6 V |          | 2.7 V – 3.6 V |  |
|     |                   |  | Note            | Min (ns) | Max (ns)      | Min (ns) | Max (ns)      | Min (ns) | Max (ns)      | Min (ns) | Max (ns)      | Min (ns) | Max (ns)      |  |
| R1  | t <sub>AVAV</sub> | Read Cycle Time  | 3,4             | 70       |               | 80       |               | 90       |               | 100      |               | 110      |               |  |
| R2  | t <sub>AVQV</sub> | Address to Output Delay  | 3,4             |          | 70            |          | 80            |          | 90            |          | 100           |          | 110           |  |
| R3  | t <sub>ELQV</sub> | CE# to Output Delay  | 1,3,4           |          | 70            |          | 80            |          | 90            |          | 100           |          | 110           |  |
| R4  | t <sub>GLQV</sub> | OE# to Output Delay  | 1,3,4           |          | 20            |          | 30            |          | 30            |          | 30            |          | 30            |  |
| R5  | t <sub>PHQV</sub> | RP# to Output Delay  | 3,4             |          | 150           |          | 150           |          | 150           |          | 150           |          | 150           |  |
| R6  | t <sub>ELQX</sub> | CE# to Output in Low Z   | 2,3,4           | 0        |               | 0        |               | 0        |               | 0        |               | 0        |               |  |
| R7  | t <sub>GLQX</sub> | OE# to Output in Low Z   | 2,3,4           | 0        |               | 0        |               | 0        |               | 0        |               | 0        |               |  |
| R8  | t <sub>EHQZ</sub> | CE# to Output in High Z  | 2,3,4           |          | 20            |          | 20            |          | 20            |          | 20            |          | 20            |  |
| R9  | t <sub>GHQZ</sub> | OE# to Output in High Z  | 2,3,4           |          | 20            |          | 20            |          | 20            |          | 20            |          | 20            |  |
| R10 | t <sub>OH</sub>   | Output Hold from Address, CE#, or OE# Change, Whichever Occurs First | 2,3,4           | 0        |               | 0        |               | 0        |               | 0        |               | 0        |               |  |

**Notes:**

1. OE# may be delayed up to t<sub>ELQV</sub>–t<sub>GLQV</sub> after the falling edge of CE# without impact on t<sub>ELQV</sub>.
2. Sampled, but not 100% tested.
3. See [Figure 9, “Read Operation Waveform”](#) on page 28.
4. See [Figure 11, “AC Input/Output Reference Waveform”](#) on page 33 for timing measurements and maximum allowable input slew rate.

Table 9. Read Operations—16-Mbit Density

| #   | Sym               | Parameter  | Density         | 16 Mbit     |          |             |          |             |          |             |          |             |          |             |          | Notes |
|-----|-------------------|--|-----------------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------|
|     |                   |  | Product         | 70 ns       |          | 80 ns       |          | 90 ns       |          |             |          | 110 ns      |          |             |          |       |
|     |                   |  | V <sub>CC</sub> | 2.7 V–3.6 V |          | 2.7 V–3.6 V |          | 3.0 V–3.6 V |          | 2.7 V–3.6 V |          | 3.0 V–3.6 V |          | 2.7 V–3.6 V |          |       |
|     |                   |  |                 | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) |       |
| R1  | t <sub>AVAV</sub> | Read Cycle Time  | 70              |             | 80       |             | 80       |             | 90       |             | 100      |             | 110      |             | 3,4      |       |
| R2  | t <sub>AVQV</sub> | Address to Output Delay  |                 | 70          |          | 80          |          | 80          |          | 90          |          | 100         |          | 110         | 3,4      |       |
| R3  | t <sub>ELQV</sub> | CE# to Output Delay  |                 | 70          |          | 80          |          | 80          |          | 90          |          | 100         |          | 110         | 1,3,4    |       |
| R4  | t <sub>GLQV</sub> | OE# to Output Delay  |                 | 20          |          | 20          |          | 30          |          | 30          |          | 30          |          | 30          | 1,3,4    |       |
| R5  | t <sub>PHQV</sub> | RP# to Output Delay  |                 | 150         |          | 150         |          | 150         |          | 150         |          | 150         |          | 150         | 3,4      |       |
| R6  | t <sub>ELQX</sub> | CE# to Output in Low Z   | 0               |             | 0        |             | 0        |             | 0        |             | 0        |             | 0        |             | 2,3,4    |       |
| R7  | t <sub>GLQX</sub> | OE# to Output in Low Z   | 0               |             | 0        |             | 0        |             | 0        |             | 0        |             | 0        |             | 2,3,4    |       |
| R8  | t <sub>EHQZ</sub> | CE# to Output in High Z  |                 | 20          |          | 20          |          | 20          |          | 20          |          | 20          |          | 20          | 2,3,4    |       |
| R9  | t <sub>GHQZ</sub> | OE# to Output in High Z  |                 | 20          |          | 20          |          | 20          |          | 20          |          | 20          |          | 20          | 2,3,4    |       |
| R10 | t <sub>OH</sub>   | Output Hold from Address, CE#, or OE# Change, Whichever Occurs First | 0               |             | 0        |             | 0        |             | 0        |             | 0        |             | 0        |             | 2,3,4    |       |

**Notes:**

- 1.OE# may be delayed up to t<sub>ELQV</sub>-t<sub>GLQV</sub> after the falling edge of CE# without impact on t<sub>ELQV</sub>.
- 2.Sampled, but not 100% tested.
- 3.See [Figure 9, "Read Operation Waveform"](#) on page 28.
4. See [Figure 11, "AC Input/Output Reference Waveform"](#) on page 33 for timing measurements and maximum allowable input slew rate.

**Table 10. Read Operations—32-Mbit Density**

| #   | Sym               | Parameter  | Density         | 32 Mbit     |          |             |          |             |          |             |          |             |          |             |          | Notes |
|-----|-------------------|--|-----------------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------|
|     |                   |  | Product         | 70 ns       |          | 90 ns       |          | 100 ns      |          |             |          | 110 ns      |          |             |          |       |
|     |                   |  | V <sub>CC</sub> | 2.7 V–3.6 V |          | 2.7 V–3.6 V |          | 3.0 V–3.3 V |          | 2.7 V–3.3 V |          | 3.0 V–3.3 V |          | 2.7 V–3.3 V |          |       |
|     |                   |  |                 | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) | Min (ns)    | Max (ns) |       |
| R1  | t <sub>AVAV</sub> | Read Cycle Time  | 70              |             | 90       |             | 90       |             | 100      |             | 100      |             | 110      |             | 3,4      |       |
| R2  | t <sub>AVQV</sub> | Address to Output Delay  |                 | 70          |          | 90          |          | 90          |          | 100         |          | 100         |          | 110         | 3,4      |       |
| R3  | t <sub>ELQV</sub> | CE# to Output Delay  |                 | 70          |          | 90          |          | 90          |          | 100         |          | 100         |          | 110         | 1,3,4    |       |
| R4  | t <sub>GLQV</sub> | OE# to Output Delay  |                 | 20          |          | 20          |          | 30          |          | 30          |          | 30          |          | 30          | 1,3,4    |       |
| R5  | t <sub>PHQV</sub> | RP# to Output Delay  |                 | 150         |          | 150         |          | 150         |          | 150         |          | 150         |          | 150         | 3,4      |       |
| R6  | t <sub>ELQX</sub> | CE# to Output in Low Z   | 0               |             | 0        |             | 0        |             | 0        |             | 0        |             | 0        |             | 2,3,4    |       |
| R7  | t <sub>GLQX</sub> | OE# to Output in Low Z   | 0               |             | 0        |             | 0        |             | 0        |             | 0        |             | 0        |             | 2,3,4    |       |
| R8  | t <sub>EHQZ</sub> | CE# to Output in High Z  |                 | 20          |          | 20          |          | 20          |          | 20          |          | 20          |          | 20          | 2,3,4    |       |
| R9  | t <sub>GHQZ</sub> | OE# to Output in High Z  |                 | 20          |          | 20          |          | 20          |          | 20          |          | 20          |          | 20          | 2,3,4    |       |
| R10 | t <sub>OH</sub>   | Output Hold from Address, CE#, or OE# Change, Whichever Occurs First | 0               |             | 0        |             | 0        |             | 0        |             | 0        |             | 0        |             | 2,3,4    |       |

**Notes:**

1. OE# may be delayed up to t<sub>ELQV</sub>–t<sub>GLQV</sub> after the falling edge of CE# without impact on t<sub>ELQV</sub>.
2. Sampled, but not 100% tested.
3. See [Figure 9, “Read Operation Waveform” on page 28](#).
4. See [Figure 11, “AC Input/Output Reference Waveform” on page 33](#) for timing measurements and maximum allowable input slew rate.

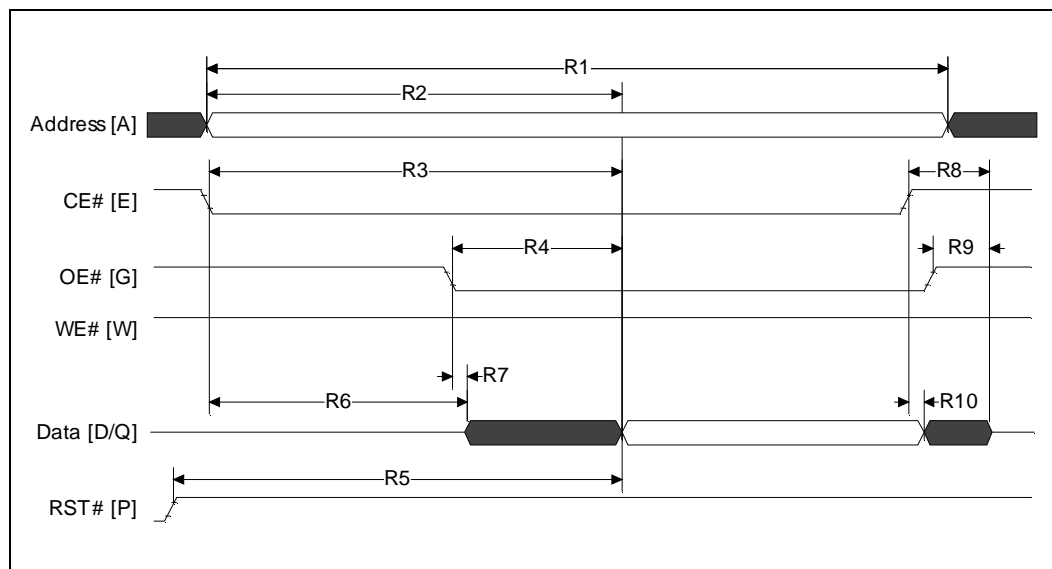
Table 11. Read Operations — 64-Mbit Density

| #   | Sym               | Parameter  | Density         |     | 64 Mbit     |     |             |     | Unit |
|-----|-------------------|--|-----------------|-----|-------------|-----|-------------|-----|------|
|     |                   |  | Product         |     | 70 ns       |     | 80 ns       |     |      |
|     |                   |  | V <sub>CC</sub> |     | 2.7 V–3.6 V |     | 2.7 V–3.6 V |     |      |
|     |                   |  | Note            | Min | Max         | Min | Max         |     |      |
| R1  | t <sub>AVAV</sub> | Read Cycle Time  | 3,4             |     | 70          |     | 80          |     | ns   |
| R2  | t <sub>AVQV</sub> | Address to Output Delay  | 3,4             |     |             | 70  |             | 80  | ns   |
| R3  | t <sub>ELQV</sub> | CE# to Output Delay  | 1,3,4           |     |             | 70  |             | 80  | ns   |
| R4  | t <sub>GLQV</sub> | OE# to Output Delay  | 1,3,4           |     |             | 20  |             | 20  | ns   |
| R5  | t <sub>PHQV</sub> | RP# to Output Delay  | 3,4             |     |             | 150 |             | 150 | ns   |
| R6  | t <sub>ELQX</sub> | CE# to Output in Low Z   | 2,3,4           |     | 0           |     | 0           |     | ns   |
| R7  | t <sub>GLQX</sub> | OE# to Output in Low Z   | 2,3,4           |     | 0           |     | 0           |     | ns   |
| R8  | t <sub>EHQZ</sub> | CE# to Output in High Z  | 2,3,4           |     |             | 20  |             | 20  | ns   |
| R9  | t <sub>GHQZ</sub> | OE# to Output in High Z  | 2,3,4           |     |             | 20  |             | 20  | ns   |
| R10 | t <sub>OH</sub>   | Output Hold from Address, CE#, or OE# Change, Whichever Occurs First | 2,3,4           |     | 0           |     | 0           |     | ns   |

**Notes:**

- 1.OE# may be delayed up to t<sub>ELQV</sub>-t<sub>GLQV</sub> after the falling edge of CE# without impact on t<sub>ELQV</sub>.
- 2.Sampled, but not 100% tested.
- 3.See Figure 9, “Read Operation Waveform” on page 28.
- 4.See Figure 11, “AC Input/Output Reference Waveform” on page 33 for timing measurements and maximum allowable input slew rate.

Figure 9. Read Operation Waveform



## 7.2 AC Write Characteristics

Table 12. Write Operations—8-Mbit Density

| #   | Sym                                   | Parameter                                     | Density         |               | 8 Mbit   |          |          |        |     |
|-----|---------------------------------------|---|-----------------|---------------|----------|----------|----------|--------|-----|
|     |                                       |   | Product         |               | 70ns     | 90 ns    |          | 110 ns |     |
|     |                                       |   | V <sub>CC</sub> | 3.0 V – 3.6 V |          | 80       |          | 100    |     |
|     |                                       |   |                 | 2.7 V – 3.6 V | 70       |          | 90       |        | 110 |
|     |                                       | Note  | Min (ns)        | Min (ns)      | Min (ns) | Min (ns) | Min (ns) |        |     |
| W1  | t <sub>PHWL</sub> / t <sub>PHL</sub>  | RP# High Recovery to WE# (CE#) Going Low      | 4,5             | 150           | 150      | 150      | 150      | 150    |     |
| W2  | t <sub>ELWL</sub> / t <sub>WLEL</sub> | CE# (WE#) Setup to WE# (CE#) Going Low        | 4,5             | 0             | 0        | 0        | 0        | 0      |     |
| W3  | t <sub>WLWH</sub> / t <sub>ELEH</sub> | WE# (CE#) Pulse Width                         | 4,5             | 45            | 50       | 60       | 70       | 70     |     |
| W4  | t <sub>DVWH</sub> / t <sub>DVEH</sub> | Data Setup to WE# (CE#) Going High            | 2,4,5           | 40            | 50       | 50       | 60       | 60     |     |
| W5  | t <sub>AVWH</sub> / t <sub>AVEH</sub> | Address Setup to WE# (CE#) Going High         | 2,4,5           | 50            | 50       | 60       | 70       | 70     |     |
| W6  | t <sub>WHEH</sub> / t <sub>EHWH</sub> | CE# (WE#) Hold Time from WE# (CE#) High       | 4,5             | 0             | 0        | 0        | 0        | 0      |     |
| W7  | t <sub>WHDX</sub> / t <sub>EHDX</sub> | Data Hold Time from WE# (CE#) High            | 2,4,5           | 0             | 0        | 0        | 0        | 0      |     |
| W8  | t <sub>WHAX</sub> / t <sub>EHAX</sub> | Address Hold Time from WE# (CE#) High         | 2,4,5           | 0             | 0        | 0        | 0        | 0      |     |
| W9  | t <sub>WHWL</sub> / t <sub>EHEL</sub> | WE# (CE#) Pulse Width High                    | 2,4,5           | 25            | 30       | 30       | 30       | 30     |     |
| W10 | t <sub>VPWH</sub> / t <sub>VPEH</sub> | V <sub>PP</sub> Setup to WE# (CE#) Going High | 3,4,5           | 200           | 200      | 200      | 200      | 200    |     |
| W11 | t <sub>QVVL</sub>                     | V <sub>PP</sub> Hold from Valid SRD           | 3,4             | 0             | 0        | 0        | 0        | 0      |     |
| W12 | t <sub>BHWH</sub> / t <sub>BHEH</sub> | WP# Setup to WE# (CE#) Going High             | 3,4             | 0             | 0        | 0        | 0        | 0      |     |
| W13 | t <sub>QVBL</sub>                     | WP# Hold from Valid SRD                       | 3,4             | 0             | 0        | 0        | 0        | 0      |     |
| W14 | t <sub>WHGL</sub>                     | WE# High to OE# Going Low                     | 3,4             | 30            | 30       | 30       | 30       | 30     |     |

**Notes:**

- Write pulse width (t<sub>WP</sub>) is defined from CE# or WE# going low (whichever goes low last) to CE# or WE# going high (whichever goes high first). Hence, t<sub>WP</sub> = t<sub>WLWH</sub> = t<sub>ELEH</sub> = t<sub>WLEH</sub> = t<sub>ELWH</sub>. Similarly, write pulse width high (t<sub>WPH</sub>) is defined from CE# or WE# going high (whichever goes high first) to CE# or WE# going low (whichever goes low last). Hence, t<sub>WPH</sub> = t<sub>WHWL</sub> = t<sub>EHEL</sub> = t<sub>WHEL</sub> = t<sub>EHWL</sub>.
- Refer to Table 22, "Command Bus Operations" on page 46 for valid A<sub>IN</sub> or D<sub>IN</sub>.
- Sampled, but not 100% tested.
- See Figure 11, "AC Input/Output Reference Waveform" on page 33 for timing measurements and maximum allowable input slew rate.
- See Figure 10, "Write Operations Waveform" on page 32.

Table 13. Write Operations—16-Mbit Density

| #   | Sym                                   | Parameter                                     | Density         |               | 16 Mbit |       |       |     |        |     | Unit |
|-----|---------------------------------------|---|-----------------|---------------|---------|-------|-------|-----|--------|-----|------|
|     |                                       |   | Product         |               | 70 ns   | 80 ns | 90 ns |     | 110 ns |     |      |
|     |                                       |   | V <sub>CC</sub> | 3.0 V – 3.6 V |         |       | 80    |     | 100    |     |      |
|     |                                       |   |                 | 2.7 V – 3.6 V | 70      | 80    |       | 90  |        | 110 |      |
|     |                                       | Note  | Min             | Min           | Min     | Min   | Min   | Min |        |     |      |
| W1  | t <sub>PHWL</sub> / t <sub>PHEL</sub> | RP# High Recovery to WE# (CE#) Going Low      | 4,5             | 150           | 150     | 150   | 150   | 150 | 150    | ns  |      |
| W2  | t <sub>ELWL</sub> / t <sub>WLEL</sub> | CE# (WE#) Setup to WE# (CE#) Going Low        | 4,5             | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W3  | t <sub>WLWH</sub> / t <sub>ELEH</sub> | WE# (CE#) Pulse Width                         | 1,4,5           | 45            | 50      | 50    | 60    | 70  | 70     | ns  |      |
| W4  | t <sub>DVWH</sub> / t <sub>DVEH</sub> | Data Setup to WE# (CE#) Going High            | 2,4,5           | 40            | 40      | 50    | 50    | 60  | 60     | ns  |      |
| W5  | t <sub>AVWH</sub> / t <sub>AVEH</sub> | Address Setup to WE# (CE#) Going High         | 2,4,5           | 50            | 50      | 50    | 60    | 70  | 70     | ns  |      |
| W6  | t <sub>WHEH</sub> / t <sub>EHWH</sub> | CE# (WE#) Hold Time from WE# (CE#) High       | 4,5             | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W7  | t <sub>WHDX</sub> / t <sub>EHDX</sub> | Data Hold Time from WE# (CE#) High            | 2,4,5           | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W8  | t <sub>WHAX</sub> / t <sub>EHAX</sub> | Address Hold Time from WE# (CE#) High         | 2,4,5           | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W9  | t <sub>WHWL</sub> / t <sub>EHEL</sub> | WE# (CE#) Pulse Width High                    | 1,4,5           | 25            | 30      | 30    | 30    | 30  | 30     | ns  |      |
| W10 | t <sub>VPWH</sub> / t <sub>VPEH</sub> | V <sub>PP</sub> Setup to WE# (CE#) Going High | 3,4,5           | 200           | 200     | 200   | 200   | 200 | 200    | ns  |      |
| W11 | t <sub>QVVL</sub>                     | V <sub>PP</sub> Hold from Valid SRD           | 3,4             | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W12 | t <sub>BHWH</sub> / t <sub>BHEH</sub> | WP# Setup to WE# (CE#) Going High             | 3,4             | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W13 | t <sub>QVBL</sub>                     | WP# Hold from Valid SRD                       | 3,4             | 0             | 0       | 0     | 0     | 0   | 0      | ns  |      |
| W14 | t <sub>WHGL</sub>                     | WE# High to OE# Going Low                     | 3,4             | 30            | 30      | 30    | 30    | 30  | 30     | ns  |      |

**Notes:**

- Write pulse width (t<sub>WP</sub>) is defined from CE# or WE# going low (whichever goes low last) to CE# or WE# going high (whichever goes high first). Hence, t<sub>WP</sub> = t<sub>WLWH</sub> = t<sub>ELEH</sub> = t<sub>WLEH</sub> = t<sub>ELWH</sub>. Similarly, write pulse width high (t<sub>WPH</sub>) is defined from CE# or WE# going high (whichever goes high first) to CE# or WE# going low (whichever goes low last). Hence, t<sub>WPH</sub> = t<sub>WHWL</sub> = t<sub>EHEL</sub> = t<sub>WHEL</sub> = t<sub>EHWL</sub>.
- Refer to Table 22, "Command Bus Operations" on page 46 for valid A<sub>IN</sub> or D<sub>IN</sub>.
- Sampled, but not 100% tested.
- See Figure 11, "AC Input/Output Reference Waveform" on page 33 for timing measurements and maximum allowable input slew rate.
- See Figure 10, "Write Operations Waveform" on page 32.

**Table 14. Write Operations—32-Mbit Density**

| #    | Sym                                   | Parameter                                     | Density         |                            | 32 Mbit |       |        |     |        |     | Unit |
|------|---------------------------------------|---|-----------------|----------------------------|---------|-------|--------|-----|--------|-----|------|
|      |                                       |   | Product         |                            | 70 ns   | 90 ns | 100 ns |     | 110 ns |     |      |
|      |                                       |   | V <sub>CC</sub> | 3.0 V – 3.6 V <sup>6</sup> |         |       | 90     |     | 100    |     |      |
|      |                                       |   |                 | 2.7 V – 3.6 V              | 70      | 90    |        | 100 |        | 110 |      |
| Note |                                       | Min   | Min             | Min                        | Min     | Min   | Min    |     |        |     |      |
| W1   | t <sub>PHWL</sub> / t <sub>PHEL</sub> | RP# High Recovery to WE# (CE#) Going Low      | 4,5             | 150                        | 150     | 150   | 150    | 150 | 150    | ns  |      |
| W2   | t <sub>ELWL</sub> / t <sub>WLEL</sub> | CE# (WE#) Setup to WE# (CE#) Going Low        | 4,5             | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W3   | t <sub>WLWH</sub> / t <sub>ELEH</sub> | WE# (CE#) Pulse Width                         | 1,4,5           | 45                         | 60      | 60    | 70     | 70  | 70     | ns  |      |
| W4   | t <sub>DVWH</sub> / t <sub>DVEH</sub> | Data Setup to WE# (CE#) Going High            | 2,4,5           | 40                         | 40      | 50    | 60     | 60  | 60     | ns  |      |
| W5   | t <sub>AVWH</sub> / t <sub>AVEH</sub> | Address Setup to WE# (CE#) Going High         | 2,4,5           | 50                         | 60      | 60    | 70     | 70  | 70     | ns  |      |
| W6   | t <sub>WHEH</sub> / t <sub>EHWH</sub> | CE# (WE#) Hold Time from WE# (CE#) High       | 4,5             | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W7   | t <sub>WHDX</sub> / t <sub>EHDX</sub> | Data Hold Time from WE# (CE#) High            | 2,4,5           | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W8   | t <sub>WHAX</sub> / t <sub>EHAX</sub> | Address Hold Time from WE# (CE#) High         | 2,4,5           | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W9   | t <sub>WHWL</sub> / t <sub>EHEL</sub> | WE# (CE#) Pulse Width High                    | 1,4,5           | 25                         | 30      | 30    | 30     | 30  | 30     | ns  |      |
| W10  | t <sub>VPWH</sub> / t <sub>VPEH</sub> | V <sub>PP</sub> Setup to WE# (CE#) Going High | 3,4,5           | 200                        | 200     | 200   | 200    | 200 | 200    | ns  |      |
| W11  | t <sub>QVVL</sub>                     | V <sub>PP</sub> Hold from Valid SRD           | 3,4             | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W12  | t <sub>BHWH</sub> / t <sub>BHEH</sub> | WP# Setup to WE# (CE#) Going High             | 3,4             | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W13  | t <sub>QVBL</sub>                     | WP# Hold from Valid SRD                       | 3,4             | 0                          | 0       | 0     | 0      | 0   | 0      | ns  |      |
| W14  | t <sub>WHGL</sub>                     | WE# High to OE# Going Low                     | 3,4             | 30                         | 30      | 30    | 30     | 30  | 30     | ns  |      |

**Notes:**

- Write pulse width (t<sub>WP</sub>) is defined from CE# or WE# going low (whichever goes low last) to CE# or WE# going high (whichever goes high first). Hence, t<sub>WP</sub> = t<sub>WLWH</sub> = t<sub>ELEH</sub> = t<sub>WLEH</sub> = t<sub>ELWH</sub>. Similarly, write pulse width high (t<sub>WPH</sub>) is defined from CE# or WE# going high (whichever goes high first) to CE# or WE# going low (whichever goes low last). Hence, t<sub>WPH</sub> = t<sub>WHWL</sub> = t<sub>EHEL</sub> = t<sub>WHEL</sub> = t<sub>EHWL</sub>.
- Refer to [Table 22, "Command Bus Operations"](#) on page 46 for valid A<sub>IN</sub> or D<sub>IN</sub>.
- Sampled, but not 100% tested.
- See [Figure 11, "AC Input/Output Reference Waveform"](#) on page 33 for timing measurements and maximum allowable input slew rate.
- See [Figure 10, "Write Operations Waveform"](#) on page 32.
- V<sub>CC</sub>Max = 3.3 V for 32-Mbit 0.25 Micron product.

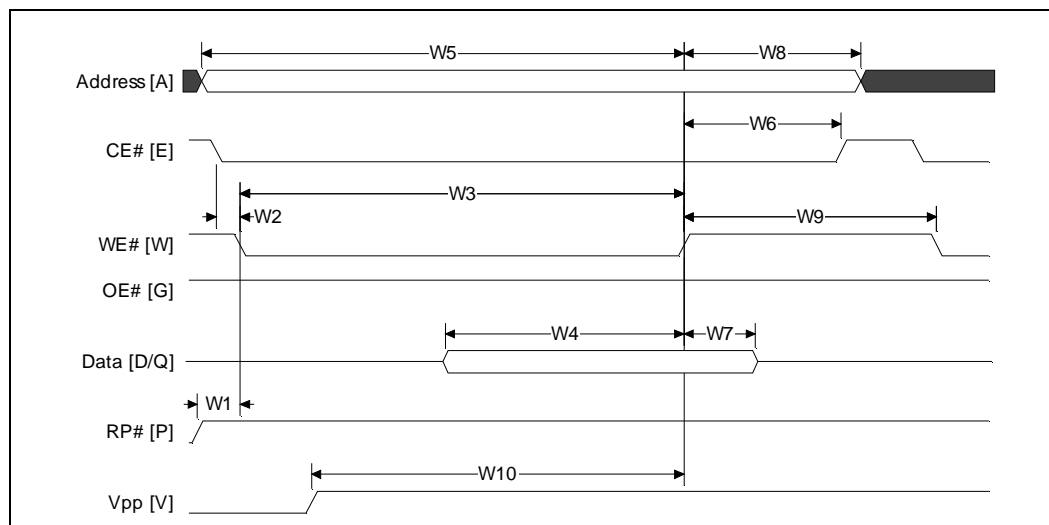
**Table 15. Write Operations—64-Mbit Density**

| #   | Symbol                                | Parameter                                     | Density         |               |       | 64 Mbit | Unit |
|-----|---------------------------------------|---|-----------------|---------------|-------|---------|------|
|     |                                       |   | Product         |               |       | 80 ns   |      |
|     |                                       |   | V <sub>CC</sub> | 2.7 V – 3.6 V | Note  | Min     |      |
| W1  | t <sub>PHWL</sub> / t <sub>PHEL</sub> | RP# High Recovery to WE# (CE#) Going Low      |                 |               | 4,5   | 150     | ns   |
| W2  | t <sub>ELWL</sub> / t <sub>WLEL</sub> | CE# (WE#) Setup to WE# (CE#) Going Low        |                 |               | 4,5   | 0       | ns   |
| W3  | t <sub>WLWH</sub> / t <sub>ELEH</sub> | WE# (CE#) Pulse Width                         |                 |               | 1,4,5 | 60      | ns   |
| W4  | t <sub>DVWH</sub> / t <sub>DVEH</sub> | Data Setup to WE# (CE#) Going High            |                 |               | 2,4,5 | 40      | ns   |
| W5  | t <sub>AVWH</sub> / t <sub>AVEH</sub> | Address Setup to WE# (CE#) Going High         |                 |               | 2,4,5 | 60      | ns   |
| W6  | t <sub>WHEH</sub> / t <sub>EHWH</sub> | CE# (WE#) Hold Time from WE# (CE#) High       |                 |               | 4,5   | 0       | ns   |
| W7  | t <sub>WHDX</sub> / t <sub>EHDH</sub> | Data Hold Time from WE# (CE#) High            |                 |               | 2,4,5 | 0       | ns   |
| W8  | t <sub>WHAX</sub> / t <sub>EHAH</sub> | Address Hold Time from WE# (CE#) High         |                 |               | 2,4,5 | 0       | ns   |
| W9  | t <sub>WHWL</sub> / t <sub>EHEL</sub> | WE# (CE#) Pulse Width High                    |                 |               | 1,4,5 | 30      | ns   |
| W10 | t <sub>VPWH</sub> / t <sub>VPEH</sub> | V <sub>PP</sub> Setup to WE# (CE#) Going High |                 |               | 3,4,5 | 200     | ns   |
| W11 | t <sub>QVVL</sub>                     | V <sub>PP</sub> Hold from Valid SRD           |                 |               | 3,4   | 0       | ns   |
| W12 | t <sub>BHWH</sub> / t <sub>BHEH</sub> | WP# Setup to WE# (CE#) Going High             |                 |               | 3,4   | 0       | ns   |
| W13 | t <sub>QVBL</sub>                     | WP# Hold from Valid SRD                       |                 |               | 3,4   | 0       | ns   |
| W14 | t <sub>WHGL</sub>                     | WE# High to OE# Going Low                     |                 |               | 3,4   | 30      | ns   |

**Notes:**

1. Write pulse width (t<sub>WP</sub>) is defined from CE# or WE# going low (whichever goes low last) to CE# or WE# going high (whichever goes high first). Hence, t<sub>WP</sub> = t<sub>WLWH</sub> = t<sub>ELEH</sub> = t<sub>WLEH</sub> = t<sub>ELWH</sub>. Similarly, write pulse width high (t<sub>WPH</sub>) is defined from CE# or WE# going high (whichever goes high first) to CE# or WE# going low (whichever goes low last). Hence, t<sub>WPH</sub> = t<sub>WHWL</sub> = t<sub>EHEL</sub> = t<sub>WHEL</sub> = t<sub>EHWL</sub>.
2. Refer to Table 22, “Command Bus Operations” on page 46 for valid A<sub>IN</sub> or D<sub>IN</sub>.
3. Sampled, but not 100% tested.
4. See Figure 11, “AC Input/Output Reference Waveform” on page 33 for timing measurements and maximum allowable input slew rate.
5. See Figure 10, “Write Operations Waveform” on page 32.

**Figure 10. Write Operations Waveform**





### 7.3 Erase and Program Timings

Table 16. Erase and Program Timings

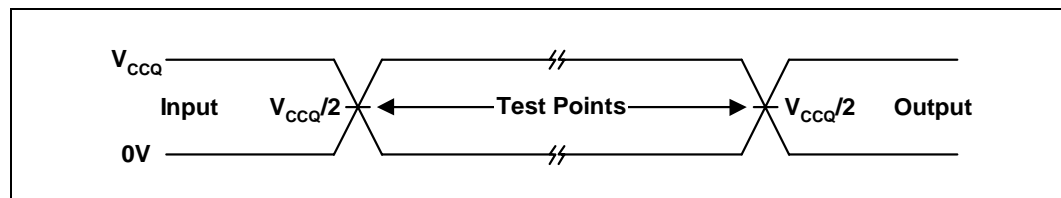
| Symbol                                  | Parameter  | V <sub>PP</sub> | 1.65 V–3.6 V |      | 11.4 V–12.6 V |      | Unit |
|---|--|-----------------|--------------|------|---------------|------|------|
|   |  | Note            | Typ          | Max  | Typ           | Max  |      |
| t <sub>BWPB</sub>                       | 4-KW Parameter Block Word Program Time             | 1, 2, 3         | 0.10         | 0.30 | 0.03          | 0.12 | s    |
| t <sub>BWMB</sub>                       | 32-KW Main Block Word Program Time                 | 1, 2, 3         | 0.8          | 2.4  | 0.24          | 1    | s    |
| t <sub>WHQV1</sub> / t <sub>EHQV1</sub> | Word Program Time for 0.13 and 0.18 Micron Product | 1, 2, 3         | 12           | 200  | 8             | 185  | μs   |
|   | Word Program Time for 0.25 Micron Product          | 1, 2, 3         | 22           | 200  | 8             | 185  | μs   |
| t <sub>WHQV2</sub> / t <sub>EHQV2</sub> | 4-KW Parameter Block Erase Time                    | 1, 2, 3         | 0.5          | 4    | 0.4           | 4    | s    |
| t <sub>WHQV3</sub> / t <sub>EHQV3</sub> | 32-KW Main Block Erase Time                        | 1, 2, 3         | 1            | 5    | 0.6           | 5    | s    |
| t <sub>WHRH1</sub> / t <sub>EHRH1</sub> | Program Suspend Latency                            | 1,3             | 5            | 10   | 5             | 10   | μs   |
| t <sub>WHRH2</sub> / t <sub>EHRH2</sub> | Erase Suspend Latency                              | 1,3             | 5            | 20   | 5             | 20   | μs   |

**Notes:**

1. Typical values measured at T<sub>A</sub> = +25 °C and nominal voltages.
2. Excludes external system-level overhead.
3. Sampled, but not 100% tested.

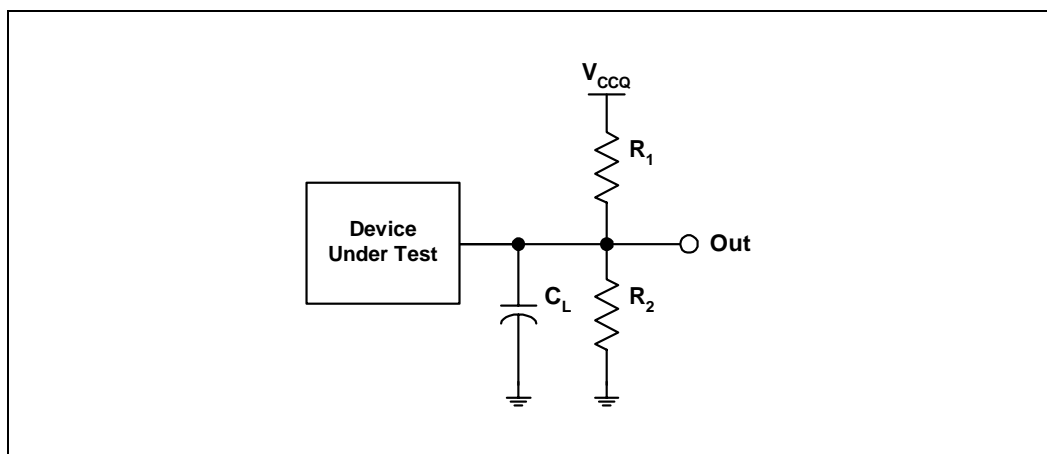
### 7.4 AC I/O Test Conditions

Figure 11. AC Input/Output Reference Waveform



**Note:** Input timing begins, and output timing ends, at V<sub>CCQ</sub>/2. Input rise and fall times (10% to 90%) < 5 ns. Worst-case speed conditions are when V<sub>CC</sub> = V<sub>CCMin</sub>.

Figure 12. Transient Equivalent Testing Load Circuit



**Note:** See Table 17 for component values.

Table 17. Test Configuration Component Values for Worst-Case Speed Conditions

| Test Configuration          | $C_L$ (pF) | $R_1$ (k $\Omega$ ) | $R_2$ (k $\Omega$ ) |
|-----------------------------|------------|---------------------|---------------------|
| $V_{CCQ}$ Min Standard Test | 50         | 25                  | 25                  |

**Note:**  $C_L$  includes jig capacitance.

## 7.5 Device Capacitance

$T_A = 25^\circ\text{C}$ ,  $f = 1\text{ MHz}$

Table 18. Device Capacitance

| Symbol    | Parameter <sup>§</sup> | Typ | Max | Unit | Condition                |
|-----------|------------------------|-----|-----|------|--------------------------|
| $C_{IN}$  | Input Capacitance      | 6   | 8   | pF   | $V_{IN} = 0.0\text{ V}$  |
| $C_{OUT}$ | Output Capacitance     | 8   | 12  | pF   | $V_{OUT} = 0.0\text{ V}$ |

<sup>§</sup>Sampled, not 100% tested.

## 8.0 Power and Reset Specifications

---

Intel® flash devices have a tiered approach to power savings that can significantly reduce overall system power consumption. The Automatic Power Savings (APS) feature reduces power consumption when the device is selected but idle. If CE# is deasserted, the flash enters its standby mode, where current consumption is even lower. If RP# is deasserted, the flash enters deep power-down mode for ultra-low current consumption. The combination of these features can minimize memory power consumption, and therefore, overall system power consumption.

### 8.1 Active Power (Program/Erase/Read)

With CE# at a logic-low level and RP# at a logic-high level, the device is in the active mode. Refer to the DC Characteristic tables for  $I_{CC}$  current values. Active power is the largest contributor to overall system power consumption. Minimizing the active current could have a profound effect on system power consumption, especially for battery-operated devices.

### 8.2 Automatic Power Savings (APS)

Automatic Power Savings provides low-power operation during read mode. After data is read from the memory array and the address lines are idle, APS circuitry places the device in a mode where typical current is comparable to  $I_{CCS}$ . The flash stays in this static state with outputs valid until a new location is read.

### 8.3 Standby Power

When CE# is at a logic-high level ( $V_{IH}$ ), the flash memory is in standby mode, which disables much of the device's circuitry and substantially reduces power consumption. Outputs are placed in a high-impedance state independent of the status of the OE# signal. If CE# transitions to a logic-high level during Erase or Program operations, the device will continue to perform the operation and consume corresponding active power until the operation is completed.

System engineers should analyze the breakdown of standby time versus active time and quantify the respective power consumption in each mode for their specific application. This approach will provide a more accurate measure of application-specific power and energy requirements.

### 8.4 Deep Power-Down Mode

The deep power-down mode is activated when  $RP\# = V_{IL}$ . During read modes, RP# going low de-selects the memory and places the outputs in a high-impedance state. Recovery from deep power-down requires a minimum time of  $t_{PHQV}$  for read operations, and  $t_{PHWL}/t_{PHEL}$  for write operations.

During program or erase modes, RP# transitioning low aborts the in-progress operation. The memory contents of the address being programmed or the block being erased are no longer valid as the data integrity has been compromised by the abort. During deep power-down, all internal circuits are switched to a low-power savings mode (RP# transitioning to  $V_{IL}$  or turning off power to the device clears the Status Register).

## 8.5 Power and Reset Considerations

### 8.5.1 Power-Up/Down Characteristics

To prevent any condition that may result in a spurious write or erase operation, Intel recommends to power-up VCC and VCCQ together. Conversely, VCC and VCCQ must power-down together.

Intel also recommends that you power-up VPP with or after VCC has reached  $V_{CC_{min}}$ . Conversely, VPP must powerdown with or slightly before VCC.

If VCCQ and/or VPP are not connected to the VCC supply, then VCC must attain  $V_{CC_{min}}$  before applying VCCQ and VPP. Device inputs must not be driven before supply voltage reaches  $V_{CC_{min}}$ .

Power supply transitions must only occur when RP# is low.

### 8.5.2 RP# Connected to System Reset

The use of RP# during system reset is important with automated program/erase devices since the system reads from the flash memory when it comes out of reset. If a CPU reset occurs without a flash memory reset, proper CPU initialization will not occur because the flash memory may be providing status information instead of array data. Intel recommends connecting RP# to the system CPU RESET# signal to allow proper CPU/flash initialization following system reset.

System designers must guard against spurious writes when  $V_{CC}$  voltages are above  $V_{LKO}$ . Because both WE# and CE# must be low for a command write, driving either signal to  $V_{IH}$  will inhibit writes to the device. The CUI architecture provides additional protection since alteration of memory contents can only occur after successful completion of the two-step command sequences. The device is also disabled until RP# is brought to  $V_{IH}$ , regardless of the state of its control inputs. By holding the device in reset during power-up/down, invalid bus conditions during power-up can be masked, providing yet another level of memory protection.

### 8.5.3 $V_{CC}$ , $V_{PP}$ and RP# Transitions

The CUI latches commands as issued by system software and is not altered by  $V_{PP}$  or CE# transitions or WSM actions. Its default state upon power-up, after exit from reset mode or after  $V_{CC}$  transitions above  $V_{LKO}$  (Lockout voltage), is read-array mode.

After any program or Block-Erase operation is complete (even after  $V_{PP}$  transitions down to  $V_{PPLK}$ ), the CUI must be reset to read-array mode by the Read Array command if access to the flash-memory array is desired.

### 8.5.4 Reset Specifications

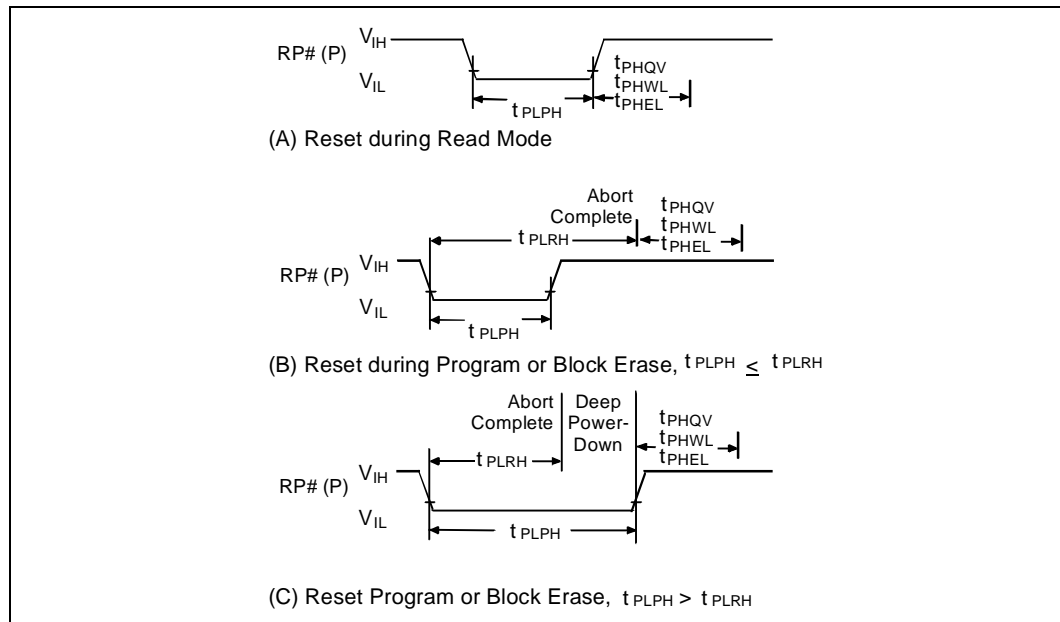
Table 19. Reset Specifications

| Symbol             | Parameter  | V <sub>CC</sub> 2.7 V – 3.6 V |     | Unit | Notes |
|--------------------|--|-------------------------------|-----|------|-------|
|                    |  | Min                           | Max |      |       |
| t <sub>PLPH</sub>  | RP# Low to Reset during Read<br>(If RP# is tied to V <sub>CC</sub> , this specification is not applicable) | 100                           |     | ns   | 1, 2  |
| t <sub>PLRH1</sub> | RP# Low to Reset during Block Erase  |                               | 22  | μs   | 3     |
| t <sub>PLRH2</sub> | RP# Low to Reset during Program  |                               | 12  | μs   | 3     |

**Notes:**

- 1.If t<sub>PLPH</sub> is < 100 ns the device may still reset but this is not guaranteed.
- 2.If RP# is asserted while a Block Erase or Word Program operation is not executing, the reset will complete within 100 ns.
- 3.Sampled, but not 100% tested.

Figure 13. Reset Operations Waveforms



### 8.6 Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers should consider the following three supply current issues:

- Standby current levels (I<sub>CCS</sub>)
- Read current levels (I<sub>CCR</sub>)
- Transient peaks produced by falling and rising edges of CE#.



Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-line control and proper decoupling capacitor selection will suppress these transient voltage peaks. Each flash device should have a 0.1  $\mu$ F ceramic capacitor connected between each  $V_{CC}$  and GND, and between its  $V_{PP}$  and VSS. These high-frequency, inherently low-inductance capacitors should be placed as close as possible to the package leads.

## 9.0 Device Operations

The Intel® Advanced+ Boot Block Flash Memory (C3) device uses a CUI and automated algorithms to simplify Program and Erase operations. The CUI allows for 100% CMOS-level control inputs and fixed power supplies during erasure and programming.

The internal WSM completely automates Program and Erase operations while the CUI signals the start of an operation and the Status Register reports device status. The CUI handles the WE# interface to the data and address latches as well as system status requests during WSM operation.

### 9.1 Bus Operations

The Intel® Advanced+ Boot Block Flash Memory (C3) device performs read, program, and erase operations in-system through the local CPU or microcontroller. Four control pins (CE#, OE#, WE#, and RP#) manage the data flow in and out of the flash device. [Table 20 on page 39](#) summarizes these bus operations.

**Table 20. Bus Operations**

| Mode           | RP#             | CE#             | OE#             | WE#             | DQ[15:0]         |
|----------------|-----------------|-----------------|-----------------|-----------------|------------------|
| Read           | V <sub>IH</sub> | V <sub>IL</sub> | V <sub>IL</sub> | V <sub>IH</sub> | D <sub>OUT</sub> |
| Write          | V <sub>IH</sub> | V <sub>IL</sub> | V <sub>IH</sub> | V <sub>IL</sub> | D <sub>IN</sub>  |
| Output Disable | V <sub>IH</sub> | V <sub>IL</sub> | V <sub>IH</sub> | V <sub>IH</sub> | High-Z           |
| Standby        | V <sub>IH</sub> | V <sub>IH</sub> | X               | X               | High-Z           |
| Reset          | V <sub>IL</sub> | X               | X               | X               | High-Z           |

**Note:** X = Don't Care (V<sub>IL</sub> or V<sub>IH</sub>)

#### 9.1.1 Read

When performing a read cycle, CE# and OE# must be asserted; WE# and RP# must be deasserted. CE# is the device selection control; when active low, it enables the flash memory device. OE# is the data output control; when low, data is output on DQ[15:0]. See [Figure 9, “Read Operation Waveform” on page 28](#).

#### 9.1.2 Write

A write cycle occurs when both CE# and WE# are low; RP# and OE# are high. Commands are issued to the Command User Interface (CUI). The CUI does not occupy an addressable memory location. Address and data are latched on the rising edge of the WE# or CE# pulse, whichever occurs first. See [Figure 10, “Write Operations Waveform” on page 32](#).

#### 9.1.3 Output Disable

With OE# at a logic-high level (V<sub>IH</sub>), the device outputs are disabled. DQ[15:0] are placed in a high-impedance state.

### 9.1.4 Standby

Deselecting the device by bringing CE# to a logic-high level ( $V_{IH}$ ) places the device in standby mode, which substantially reduces device power consumption without any latency for subsequent read accesses. In standby, outputs are placed in a high-impedance state independent of OE#. If deselected during a Program or Erase operation, the device continues to consume active power until the Program or Erase operation is complete.

### 9.1.5 Reset

From read mode, RP# at  $V_{IL}$  for time  $t_{PLPH}$  deselects the memory, places output drivers in a high-impedance state, and turns off all internal circuits. After return from reset, a time  $t_{PHQV}$  is required until the initial read-access outputs are valid. A delay ( $t_{PHWL}$  or  $t_{PHEL}$ ) is required after return from reset before a write cycle can be initiated. After this wake-up interval, normal operation is restored. The CUI resets to read-array mode, the Status Register is set to 0x80, and all blocks are locked. See Figure 13, “Reset Operations Waveforms” on page 37.

If RP# is taken low for time  $t_{PLPH}$  during a Program or Erase operation, the operation will be aborted; the memory contents at the aborted location (for a program) or block (for an erase) are no longer valid, since the data may be partially erased or written. The abort process goes through the following sequence:

1. When RP# goes low, the device shuts down the operation in progress, a process which takes time  $t_{PLRH}$  to complete.
2. After time  $t_{PLRH}$ , the part will either reset to read-array mode (if RP# is asserted during  $t_{PLRH}$ ) or enter reset mode (if RP# is deasserted after  $t_{PLRH}$ ). See Figure 13, “Reset Operations Waveforms” on page 37.

In both cases, after returning from an aborted operation, the relevant time  $t_{PHQV}$  or  $t_{PHWL}/t_{PHEL}$  must be observed before a Read or Write operation is initiated, as discussed in the previous paragraph. However, in this case, these delays are referenced to the end of  $t_{PLRH}$  rather than when RP# goes high.

As with any automated device, it is important to assert RP# during a system reset. When the system comes out of reset, the processor reads from the flash memory. Automated flash memories provide status information when read during Program or Block-Erase operations. If a CPU reset occurs with no flash memory reset, proper CPU initialization may not occur because the flash memory may be providing status information instead of array data. Intel® flash memories allow proper CPU initialization following a system reset through the use of the RP# input. In this application, RP# is controlled by the same RESET# signal that resets the system CPU.



## 10.0 Modes of Operation

---

### 10.1 Read Mode

The flash memory has four read modes (read array, read identifier, read status, and CFI query) and two write modes (program and erase). Three additional modes (erase suspend to program, erase suspend to read, and program suspend to read) are available only during suspended operations. [Table 22, “Command Bus Operations” on page 46](#) and [Table 23, “Command Codes and Descriptions” on page 47](#) summarize the commands used for these modes.

[Appendix A, “Write State Machine States” on page 54](#) is a comprehensive chart showing the state transitions.

#### 10.1.1 Read Array

When RP# transitions from  $V_{IL}$  (reset) to  $V_{IH}$ , the device defaults to read-array mode and will respond to the read-control inputs (CE#, address inputs, and OE#) without any additional CUI commands.

When the device is in read array mode, four control signals control data output.

- WE# must be logic high ( $V_{IH}$ )
- CE# must be logic low ( $V_{IL}$ )
- OE# must be logic low ( $V_{IL}$ )
- RP# must be logic high ( $V_{IH}$ )

In addition, the address of the desired location must be applied to the address pins. If the device is not in read-array mode, as would be the case after a Program or Erase operation, the Read Array command (0xFF) must be issued to the CUI before array reads can occur.

#### 10.1.2 Read Identifier

The read-identifier mode outputs three types of information: the manufacturer/device identifier, the block locking status, and the protection register. The device is switched to this mode by issuing the Read Identifier command (0x90). Once in this mode, read cycles from addresses shown in [Table 21](#) retrieve the specified information. To return to read-array mode, issue the Read Array command (0xFF).

Table 21. Device Identification Codes

| Item                                | Address <sup>1</sup> |             | Data          | Description   |
|-------------------------------------|----------------------|-------------|---------------|---|
|                                     | Base                 | Offset      |               |   |
| Manufacturer ID                     | Block                | 0x00        | 0x0089        |   |
| Device ID                           | Block                | 0x01        | 0x88C0        | 8-Mbit Top Boot Device  |
|                                     |                      |             | 0x88C1        | 8-Mbit Bottom Boot Device   |
|                                     |                      |             | 0x88C2        | 16-Mbit Top Boot Device   |
|                                     |                      |             | 0x88C3        | 16-Mbit Bottom Boot Device  |
|                                     |                      |             | 0x88C4        | 32-Mbit Top Boot Device   |
|                                     |                      |             | 0x88C5        | 32-Mbit Bottom Boot Device  |
|                                     |                      |             | 0x88CC        | 64-Mbit Top Boot Device   |
|                                     |                      |             | 0x88CD        | 64-Mbit Bottom Boot Device  |
| Block Lock Status <sup>2</sup>      | Block                | 0x02        | DQ0 = 0b0     | Block is unlocked   |
|                                     |                      |             | DQ0 = 0b1     | Block is locked   |
| Block Lock-Down Status <sup>2</sup> | Block                | 0x02        | DQ1 = 0b0     | Block is not locked-down  |
|                                     |                      |             | DQ1 = 0b1     | Block is locked down  |
| Protection Register Lock Status     | Block                | 0x80        | Lock Data     |   |
| Protection Register                 | Block                | 0x81 - 0x88 | Register Data | Multiple reads required to read the entire 128-bit Protection Register. |

**Notes:**

1. The address is constructed from a base address plus an offset. For example, to read the Block Lock Status for block number 38 in a bottom boot device, set the address to 0x0F8000 plus the *offset* (0x02), i.e. 0x0F8002. Then examine DQ0 of the data to determine if the block is locked.
2. See [Section 11.2, "Reading Block-Lock Status"](#) on page 50 for valid lock status.

### 10.1.3 CFI Query

The CFI query mode outputs Common Flash Interface (CFI) data after issuing the Read Query Command (0x98). The CFI data structure contains information such as block size, density, command set, and electrical specifications. Once in this mode, read cycles from addresses shown in [Appendix C, "Common Flash Interface,"](#) retrieve the specified information. To return to read-array mode, issue the Read Array command (0xFF).

### 10.1.4 Read Status Register

The Status Register indicates the status of device operations and the success/failure of that operation. The Read Status Register (0x70) command causes subsequent reads to output data from the Status Register until another command is issued. To return to reading from the array, issue a Read Array (0xFF) command.

The Status Register bits are output on DQ[7:0]. The upper byte, DQ[15:8], outputs 0x00 when a Read Status Register command is issued.

The contents of the Status Register are latched on the falling edge of OE# or CE# (whichever occurs last) which prevents possible bus errors that might occur if Status Register contents change while being read. CE# or OE# must be toggled with each subsequent status read, or the Status Register will not indicate completion of a Program or Erase operation.

When the WSM is active, SR[7] will indicate the status of the WSM; the remaining bits in the Status Register indicate whether the WSM was successful in performing the preferred operation. See Table 24, “Status Register Bit Definition” on page 48.

#### 10.1.4.1 Clear Status Register

The WSM can set Status Register bits 1 through 7 and can clear bits 2, 6, and 7, but the WSM cannot clear Status Register bits 1, 3, 4 or 5. Because bits 1, 3, 4, and 5 indicate various error conditions, these bits can be cleared only through the Clear Status Register (0x50) command. By allowing the system software to control the resetting of these bits, several operations may be performed (such as cumulatively programming several addresses or erasing multiple blocks in sequence) before reading the Status Register to determine if an error occurred during that series. Clear the Status Register before beginning another command or sequence. The Read Array command must be issued before data can be read from the memory array. Resetting the device also clears the Status Register.

## 10.2 Program Mode

Programming is executed using a two-write cycle sequence. The Program Setup command (0x40) is issued to the CUI, followed by a second write that specifies the address and data to be programmed. The WSM will execute a sequence of internally timed events to program preferred bits of the addressed location, then verify the bits are sufficiently programmed. Programming the memory results in specific bits within an address location being changed to a “0.” If users attempt to program “1”s, the memory cell contents do not change and no error occurs.

The Status Register indicates programming status. While the program sequence executes, status bit 7 is “0.” The Status Register can be polled by toggling either CE# or OE#. While programming, the only valid commands are Read Status Register, Program Suspend, and Program Resume.

When programming is complete, the program-status bits must be checked. If the programming operation was unsuccessful, SR[4] is set to indicate a program failure. If SR[3] is set, then  $V_{PP}$  was not within acceptable limits, and the WSM did not execute the program command. If SR[1] is set, a program operation was attempted on a locked block and the operation was aborted.

The Status Register should be cleared before attempting the next operation. Any CUI instruction can follow after programming is completed; however, to prevent inadvertent Status Register reads, be sure to reset the CUI to read-array mode.

### 10.2.1 12-Volt Production Programming

When  $V_{PP}$  is between 1.65 V and 3.6 V, all program and erase current is drawn through the VCC pin.

*Note:* If  $V_{PP}$  is driven by a logic signal,  $V_{IH\ min} = 1.65\ V$ . That is,  $V_{PP}$  must remain above 1.65 V to perform in-system flash modifications.

When  $V_{PP}$  is connected to a 12 V power supply, the device draws program and erase current directly from the VPP pin. This eliminates the need for an external switching transistor to control  $V_{PP}$ . [Figure 16 on page 53](#) shows examples of how the flash power supplies can be configured for various usage models.

The 12 V  $V_{PP}$  mode enhances programming performance during the short period of time typically found in manufacturing processes; however, it is not intended for extended use. You can apply 12 V to VPP during Program and Erase operations for a maximum of 1000 cycles on the main blocks and 2500 cycles on the parameter blocks. VPP may be connected to 12 V for a total of 80 hours maximum. Stressing the device beyond these limits may cause permanent damage.

## 10.2.2 Suspending and Resuming Program

The Program Suspend command halts an in-progress program operation so that data can be read from other locations of memory. Once the programming process starts, issuing the Program Suspend command to the CUI requests that the WSM suspend the program sequence at predetermined points in the program algorithm. The device continues to output Status Register data after the Program Suspend command is issued. Polling SR[7] and SR[2] will determine when the program operation has been suspended (both will be set to “1”). The program-suspend latency is specified with  $t_{WHRH1}/t_{EHRH1}$ .

A Read-Array command can now be issued to the CUI to read data from blocks other than that which is suspended. The only other valid commands while program is suspended are Read Status Register, Read Identifier, CFI Query, and Program Resume.

After the Program Resume command is issued to the flash memory, the WSM will continue with the programming process and SR[2] and SR[7] will automatically be cleared. The device automatically outputs Status Register data when read (see [Figure 18, “Program Suspend / Resume Flowchart” on page 57](#)) after the Program Resume command is issued.  $V_{PP}$  must remain at the same  $V_{PP}$  level used for program while in program-suspend mode. RP# must also remain at  $V_{IH}$ .

## 10.3 Erase Mode

To erase a block, issue the Erase Set-up and Erase Confirm commands to the CUI, along with an address identifying the block to be erased. This address is latched internally when the Erase Confirm command is issued. Block erasure results in all bits within the block being set to “1.” Only one block can be erased at a time. The WSM will execute a sequence of internally timed events to program all bits within the block to “0,” erase all bits within the block to “1,” then verify that all bits within the block are sufficiently erased. While the erase executes, status bit 7 is a “0.”

When the Status Register indicates that erasure is complete, check the erase-status bit to verify that the Erase operation was successful. If the Erase operation was unsuccessful, SR[5] of the Status Register will be set to a “1,” indicating an erase failure. If  $V_{PP}$  is not within acceptable limits after the Erase Confirm command was issued, the WSM will not execute the erase sequence; instead, SR[5] of the Status Register is set to indicate an erase error, and SR[3] is set to a “1” to identify that  $V_{PP}$  supply voltage is not within acceptable limits.

After an Erase operation, clear the Status Register (0x50) before attempting the next operation. Any CUI instruction can follow after erasure is completed; however, to prevent inadvertent status-register reads, Intel recommends that you place the flash in read-array mode after the erase is complete.

### 10.3.1 Suspending and Resuming Erase

Since an Erase operation requires on the order of seconds to complete, an Erase Suspend command is provided to allow erase-sequence interruption to read data from—or program data to— another block in memory. Once the erase sequence is started, issuing the Erase Suspend command to the CUI suspends the erase sequence at a predetermined point in the erase algorithm. The Status Register indicates if/when the Erase operation has been suspended. Erase-suspend latency is specified by  $t_{WHRH2}/t_{EHRH2}$ .

A Read Array or Program command can now be issued to the CUI to read/program data from/to blocks other than that which is suspended. This nested Program command can subsequently be suspended to read yet another location. The only valid commands while Erase is suspended are Read Status Register, Read Identifier, CFI Query, Program Setup, Program Resume, Erase Resume, Lock Block, Unlock Block, and Lock-Down Block. During erase-suspend mode, the device can be placed in a pseudo-standby mode by taking  $CE\#$  to  $V_{IH}$ , which reduces active current consumption.

Erase Resume continues the erase sequence when  $CE\# = V_{IL}$ . Similar to the end of a standard Erase operation, the Status Register must be read and cleared before the next instruction is issued.

Table 22. Command Bus Operations

| Command               | Notes | First Bus Cycle |      |               | Second Bus Cycle |      |      |
|-----------------------|-------|-----------------|------|---------------|------------------|------|------|
|                       |       | Oper            | Addr | Data          | Oper             | Addr | Data |
| Read Array            | 1,3   | Write           | X    | 0xFF          |                  |      |      |
| Read Identifier       | 1,3   | Write           | X    | 0x90          | Read             | IA   | ID   |
| CFI Query             | 1,3   | Write           | X    | 0x98          | Read             | QA   | QD   |
| Read Status Register  | 1,3   | Write           | X    | 0x70          | Read             | X    | SRD  |
| Clear Status Register | 1,3   | Write           | X    | 0x50          |                  |      |      |
| Program               | 2,3   | Write           | X    | 0x40/<br>0x10 | Write            | PA   | PD   |
| Block Erase/Confirm   | 1,3   | Write           | X    | 0x20          | Write            | BA   | DOH  |
| Program/Erase Suspend | 1,3   | Write           | X    | 0xB0          |                  |      |      |
| Program/Erase Resume  | 1,3   | Write           | X    | 0xD0          |                  |      |      |
| Lock Block            | 1,3   | Write           | X    | 0x60          | Write            | BA   | 0x01 |
| Unlock Block          | 1,3   | Write           | X    | 0x60          | Write            | BA   | 0xD0 |
| Lock-Down Block       | 1,3   | Write           | X    | 0x60          | Write            | BA   | 0x2F |
| Protection Program    | 1,3   | Write           | X    | 0xC0          | Write            | PA   | PD   |

X = "Don't Care"      PA = Prog Addr      BA = Block Addr      IA = Identifier Addr.      QA = Query Addr.  
 SRD = Status Reg. Data      PD = Prog Data      ID = Identifier Data      QD = Query Data

**Notes:**

- Following the Read Identifier or CFI Query commands, read operations output device identification data or CFI query information, respectively. See [Section 10.1.2](#) and [Section 10.1.3](#).
- Either 0x40 or 0x10 command is valid, but the Intel standard is 0x40.
- When writing commands, the upper data bus [DQ8-DQ15] should be either  $V_{IL}$  or  $V_{IH}$ , to minimize current draw.

Bus operations are defined in [Table 20, "Bus Operations"](#) on page 39.

**Table 23. Command Codes and Descriptions**

| Code (HEX) | Device Mode   | Command Description   |
|------------|---|---|
| FF         | Read Array  | This command places the device in read-array mode, which outputs array data on the data pins.   |
| 40         | Program Set-Up  | This is a two-cycle command. The first cycle prepares the CUI for a program operation. The second cycle latches addresses and data information and initiates the WSM to execute the Program algorithm. The flash outputs Status Register data when CE# or OE# is toggled. A Read Array command is required after programming to read array data. See <a href="#">Section 10.2, "Program Mode" on page 43</a> .  |
| 20         | Erase Set-Up  | This is a two-cycle command. It prepares the CUI for the Erase Confirm command. If the next command is not an Erase Confirm command, then the CUI will (a) set both SR.4 and SR.5 to "1," (b) place the device into the read-Status Register mode, and (c) wait for another command. See <a href="#">Section 10.3, "Erase Mode" on page 44</a> .  |
| D0         | Erase Confirm<br>Program/Erase Resume<br>Unlock Block     | If the previous command was an Erase Set-Up command, then the CUI will close the address and data latches and begin erasing the block indicated on the address pins. During program/erase, the device will respond only to the Read Status Register, Program Suspend and Erase Suspend commands, and will output Status Register data when CE# or OE# is toggled.<br>If a Program or Erase operation was previously suspended, this command will resume that operation.<br>If the previous command was Block Unlock Set-Up, the CUI will latch the address and unlock the block indicated on the address pins. If the block had been previously set to Lock-Down, this operation will have no effect. (See <a href="#">Section 11.1</a> ) |
| B0         | Program Suspend<br>Erase Suspend                          | Issuing this command will begin to suspend the currently executing Program/Erase operation. The Status Register will indicate when the operation has been successfully suspended by setting either the program-suspend SR[2] or erase-suspend SR[6] and the WSM status bit SR[7] to a "1" (ready). The WSM will continue to idle in the SUSPEND state, regardless of the state of all input-control pins except RP#, which will immediately shut down the WSM and the remainder of the chip if RP# is driven to V <sub>IL</sub> . See Sections 3.2.5.1 and 3.2.6.1.   |
| 70         | Read Status Register                                      | This command places the device into read-Status Register mode. Reading the device will output the contents of the Status Register, regardless of the address presented to the device. The device automatically enters this mode after a Program or Erase operation has been initiated. See <a href="#">Section 10.1.4, "Read Status Register" on page 42</a> .  |
| 50         | Clear Status Register                                     | The WSM can set the block-lock status SR[1], V <sub>PP</sub> Status SR[3], program status SR[4], and erase-status SR[5] bits in the Status Register to "1," but it cannot clear them to "0." Issuing this command clears those bits to "0."   |
| 90         | Read Identifier   | This command puts the device into the read-identifier mode so that reading the device will output the manufacturer/device codes or block-lock status. See <a href="#">Section 10.1.2, "Read Identifier" on page 41</a> .  |
| 60         | Block Lock,<br>Block Unlock,<br>Block Lock-Down<br>Set-Up | This command prepares the CUI for block-locking changes. If the next command is not Block Unlock, Block Lock, or Block Lock-Down, then the CUI will set both the program and erase-Status Register bits to indicate a command-sequence error. See <a href="#">Section 11.0, "Security Modes" on page 49</a> .   |
| 01         | Lock-Block  | If the previous command was Lock Set-Up, the CUI will latch the address and lock the block indicated on the address pins. (See <a href="#">Section 11.1</a> )   |
| 2F         | Lock-Down   | If the previous command was a Lock-Down Set-Up command, the CUI will latch the address and lock-down the block indicated on the address pins. (See <a href="#">Section 11.1</a> )   |
| 98         | CFI Query   | This command puts the device into the CFI-Query mode so that reading the device will output Common Flash Interface information. See <a href="#">Section 10.1.3</a> and <a href="#">Appendix C, "Common Flash Interface"</a> .   |
| C0         | Protection<br>Program<br>Set-Up                           | This is a two-cycle command. The first cycle prepares the CUI for a program operation to the protection register. The second cycle latches addresses and data information and initiates the WSM to execute the Protection Program algorithm to the protection register. The flash outputs Status Register data when CE# or OE# is toggled. A Read Array command is required after programming to read array data. See <a href="#">Section 11.5</a> .  |
| 10         | Alt. Prog Set-Up  | Operates the same as Program Set-up command. (See 0x40/Program Set-Up)  |
| 00         | Invalid/<br>Reserved                                      | Unassigned commands should not be used. Intel reserves the right to redefine these codes for future functions.  |

**Note:** See [Appendix A, "Write State Machine States"](#) for mode transition information.

Table 24. Status Register Bit Definition

| WSMS   | ESS | ES | PS | VPPS   | PSS | BLS | R |
|--|-----|----|----|--|-----|-----|---|
| 7  | 6   | 5  | 4  | 3  | 2   | 1   | 0 |
|  |     |    |    | <b>NOTES:</b>  |     |     |   |
| SR[7] WRITE STATE MACHINE STATUS (WSMS)<br>1 = Ready<br>0 = Busy   |     |    |    | Before checking program or erase- status bits, check the Write State Machine bit first to determine Word Program or Block Erase completion.  |     |     |   |
| SR[6] = ERASE-SUSPEND STATUS (ESS)<br>1 = Erase Suspended<br>0 = Erase In Progress/Completed                                     |     |    |    | When Erase Suspend is issued, WSM halts execution and sets both WSMS and ESS bits to "1." ESS bit remains set to "1" until an Erase Resume command is issued.  |     |     |   |
| SR[5] = ERASE STATUS (ES)<br>1 = Error In Block Erase<br>0 = Successful Block Erase  |     |    |    | When this bit is set to "1," WSM has applied the maximum number of erase pulses to the block and is still unable to verify successful block erasure.   |     |     |   |
| SR[4] = PROGRAM STATUS (PS)<br>1 = Error in Programming<br>0 = Successful Programming  |     |    |    | When this bit is set to "1," WSM has attempted but failed to program a word/byte.  |     |     |   |
| SR[3] = V <sub>PP</sub> STATUS (VPPS)<br>1 = V <sub>PP</sub> Low Detect, Operation Abort<br>0 = V <sub>PP</sub> OK               |     |    |    | The V <sub>PP</sub> status bit does not provide continuous indication of V <sub>PP</sub> level. The WSM interrogates V <sub>PP</sub> level only after the Program or Erase command sequences have been entered and informs the system if V <sub>PP</sub> has not been switched on. The V <sub>PP</sub> is also checked before the operation is verified by the WSM. The V <sub>PP</sub> status bit is not guaranteed to report accurate feedback between V <sub>PP</sub> and V <sub>PP1Min</sub> . |     |     |   |
| SR[2] = PROGRAM SUSPEND STATUS (PSS)<br>1 = Program Suspended<br>0 = Program in Progress/Completed                               |     |    |    | When Program Suspend is issued, WSM halts execution and sets both WSMS and PSS bits to "1." PSS bit remains set to "1" until a Program Resume command is issued.   |     |     |   |
| SR[1] = BLOCK LOCK STATUS<br>1 = Prog/Erase attempted on a locked block; Operation aborted.<br>0 = No operation to locked blocks |     |    |    | If a Program or Erase operation is attempted to one of the locked blocks, this bit is set by the WSM. The operation specified is aborted and the device is returned to read status mode.   |     |     |   |
| SR[0] = RESERVED FOR FUTURE ENHANCEMENTS (R)   |     |    |    | This bit is reserved for future use and should be masked out when polling the Status Register.   |     |     |   |

**Note:** A Command-Sequence Error is indicated when SR[4], SR[5], and SR[7] are set.



## 11.0 Security Modes

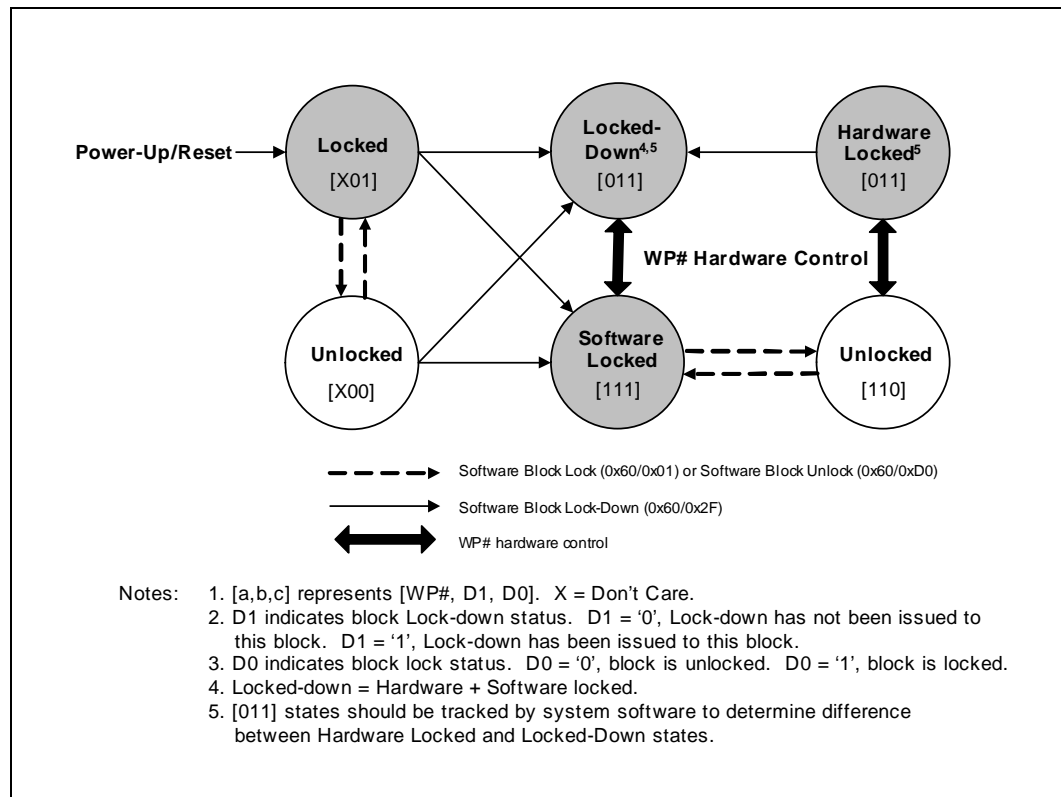
### 11.1 Flexible Block Locking

The Intel® Advanced+ Boot Block Flash Memory (C3) device offers an instant, individual block-locking scheme that allows any block to be locked or unlocked with no latency, enabling instant code and data protection.

This locking scheme offers two levels of protection. The first level allows software-only control of block locking (useful for data blocks that change frequently), while the second level requires hardware interaction before locking can be changed (useful for code blocks that change infrequently).

The following sections will discuss the operation of the locking system. The term “state [abc]” will be used to specify locking states; for example, “state [001],” where a = value of WP#, b = bit D1 of the Block Lock Status Register, and c = bit D0 of the Block Lock Status Register. [Figure 14, “Block Locking State Diagram” on page 49](#) displays all of the possible locking states.

Figure 14. Block Locking State Diagram



## 11.1.1 Locking Operation

The locking status of each block can be set to Locked, Unlocked, or Lock-Down, each of which will be described in the following sections. See [Figure 14, “Block Locking State Diagram” on page 49](#) and [Figure 21, “Locking Operations Flowchart” on page 60](#).

The following paragraph concisely summarizes the locking functionality.

### 11.1.1.1 Locked State

The default state of all blocks upon power-up or reset is locked (states [001] or [101]). Locked blocks are fully protected from alteration. Any Program or Erase operations attempted on a locked block will return an error on bit SR[1]. The state of a locked block can be changed to Unlocked or Lock Down using the appropriate software commands. An Unlocked block can be locked by writing the Lock command sequence, 0x60 followed by 0x01.

### 11.1.1.2 Unlocked State

Unlocked blocks (states [000], [100], [110]) can be programmed or erased. All unlocked blocks return to the Locked state when the device is reset or powered down. The status of an unlocked block can be changed to Locked or Locked Down using the appropriate software commands. A Locked block can be unlocked by writing the Unlock command sequence, 0x60 followed by 0xD0.

### 11.1.1.3 Lock-Down State

Blocks that are Locked-Down (state [011]) are protected from Program and Erase operations (just like Locked blocks), but their protection status cannot be changed using software commands alone. A Locked or Unlocked block can be Locked Down by writing the Lock-Down command sequence, 0x60 followed by 0x2F. Locked-Down blocks revert to the Locked state when the device is reset or powered down.

The Lock-Down function depends on the WP# input pin. When WP# = 0, blocks in Lock Down [011] are protected from program, erase, and lock status changes. When WP# = 1, the Lock-Down function is disabled ([111]), and Locked-Down blocks can be individually unlocked by software command to the [110] state, where they can be erased and programmed. These blocks can then be relocked [111] and unlocked [110] as required while WP# remains high. When WP# goes low, blocks that were previously Locked Down return to the Lock-Down state [011], regardless of any changes made while WP# was high. Device reset or power-down resets all blocks, including those in Lock-Down, to Locked state.

## 11.2 Reading Block-Lock Status

The Lock status of each block can be read in read-identifier mode of the device by issuing the read-identifier command (0x90). Subsequent reads at Block Address + 0x00002 will output the Lock status of that block. The Lock status is represented by DQ0 and DQ1:

- DQ0 indicates the Block Lock/Unlock status and is set by the Lock command and cleared by the Unlock command. It is also automatically set when entering Lock Down.
- DQ1 indicates Lock-Down status and is set by the Lock-Down command. It cannot be cleared by software—only by device reset or power-down.

See [Table 21, “Device Identification Codes” on page 42](#) for block-status information.

### 11.3 Locking Operations during Erase Suspend

Changes to block-lock status can be performed during an erase-suspend by using the standard locking command sequences to Unlock, Lock, or Lock Down a block. This operation is useful in the case when another block needs to be updated while an Erase operation is in progress.

To change block locking during an Erase operation, first issue the Erase Suspend command (0xB0), and then check the Status Register until it indicates that the Erase operation has been suspended. Next, write the preferred Lock command sequence to a block and the Lock status will be changed. After completing any preferred Lock, Read, or Program operations, resume the Erase operation with the Erase Resume command (0xD0).

If a block is Locked or Locked Down during a Suspended Erase of the same block, the locking status bits will be changed immediately. But when the Erase is resumed, the Erase operation will complete.

Locking operations cannot be performed during a Program Suspend. Refer to [Appendix A, “Write State Machine States” on page 54](#) for detailed information on which commands are valid during Erase Suspend.

### 11.4 Status Register Error Checking

Using nested-locking or program-command sequences during Erase Suspend can introduce ambiguity into Status Register results.

Since locking changes are performed using a two-cycle command sequence, for example, 0x60 followed by 0x01 to lock a block. Following the Block Lock, Block Unlock, or Block Lock-Down Setup command (0x60) with an invalid command will produce a Lock-Command error (SR[4] and SR[5] will be set to 1) in the Status Register. If a Lock-Command error occurs during an Erase Suspend, SR[4] and SR[5] will be set to 1 and will remain at 1 after the Erase is resumed. When Erase is complete, any possible error during the Erase cannot be detected by the Status Register because of the previous Lock-Command error.

A similar situation happens if an error occurs during a Program-Operation error nested within an Erase Suspend.

### 11.5 128-Bit Protection Register

The C3 device architecture includes a 128-bit protection register that can be used to increase the security of a system design. For example, the number contained in the protection register can be used to “match” the flash component with other system components, such as the CPU or ASIC, preventing device substitution. Application note, *AP-657 Designing with the Advanced+ Boot Block Flash Memory Architecture*, contains additional application information.

The 128 bits of the protection register are divided into two 64-bit segments. One of the segments is programmed at the Intel factory with a unique 64-bit number, which is unchangeable. The other segment is left blank for customer designs to program, as preferred. Once the customer segment is programmed, it can be locked to prevent further programming.

### 11.5.1 Reading the Protection Register

The protection register is read in the Read-Identifier mode. The device is switched to this mode by issuing the Read Identifier command (0x90). Once in this mode, read cycles from addresses shown in Figure 15, “Protection Register Mapping” retrieve the specified information. To return to Read-Array mode, issue the Read Array command (0xFF).

### 11.5.2 Programming the Protection Register

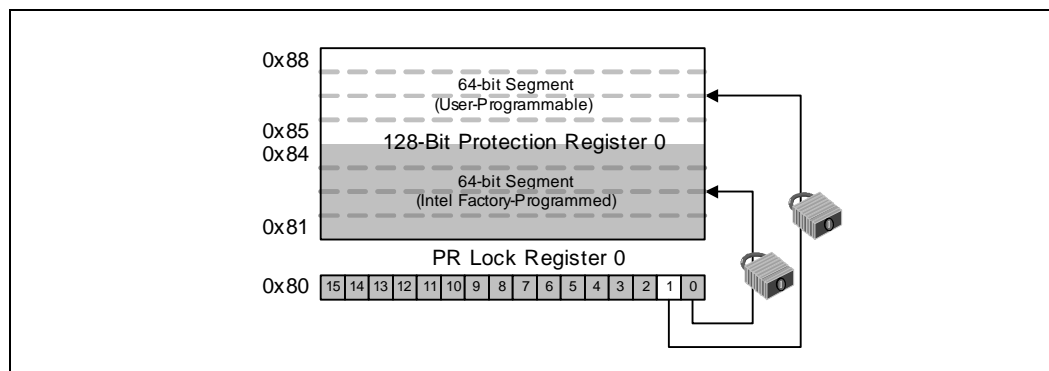
The protection register bits are programmed using the two-cycle Protection Program command. The 64-bit number is programmed 16 bits at a time. First, issue the Protection Program Setup command, 0xC0. The next write to the device will latch in address and data and program the specified location. The allowable addresses are listed in Table 21, “Device Identification Codes” on page 42. See Figure 22, “Protection Register Programming Flowchart” on page 61. Attempting to program to a previously locked protection register segment will result in a Status Register error (Program Error bit SR[4] and Lock Error bit SR[1] will be set to 1).

*Note:* Do not attempt to address Protection Program commands outside the defined protection register address space; status register can be indeterminate.

### 11.5.3 Locking the Protection Register

The user-programmable segment of the protection register is lockable by programming bit 1 of the PR-LOCK location to 0. See Figure 15, “Protection Register Mapping” on page 52. Bit 0 of this location is programmed to 0 at the Intel factory to protect the unique device number. This bit is set using the Protection Program command to program 0xFFFFD to the PR-LOCK location. After these bits have been programmed, no further changes can be made to the values stored in the protection register. Protection Program commands to a locked section will result in a Status Register error (Program Error bit SR[4] and Lock Error bit SR[1] will be set to 1). Protection register lockout state is not reversible.

Figure 15. Protection Register Mapping



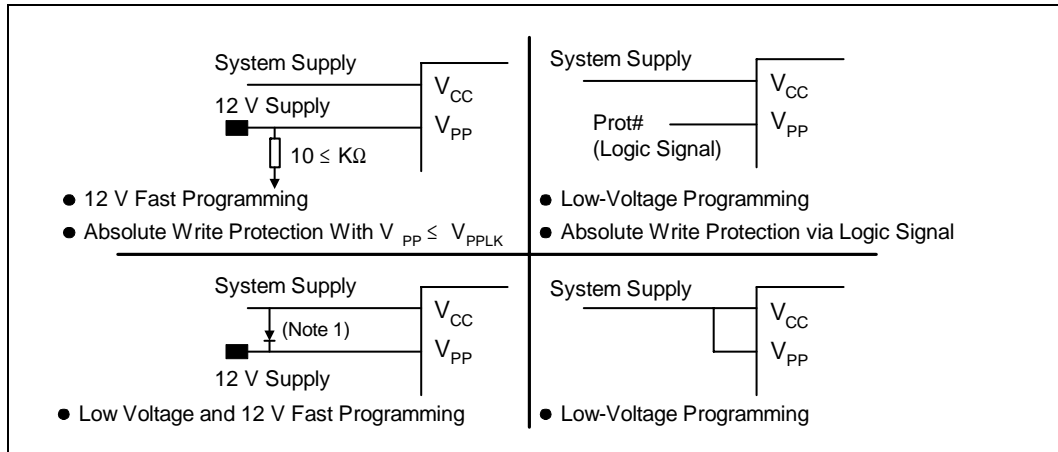
## 11.6 V<sub>PP</sub> Program and Erase Voltages

The C3 device provides in-system programming and erase in the 1.65 V–3.6 V range. For fast production programming, 12 V programming can be used. See Figure 16, “Example Power Supply Configurations” on page 53.

### 11.6.1 Program Protection

In addition to the flexible block locking, the  $V_{PP}$  programming voltage can be held low for absolute hardware write protection of all blocks in the flash device. When  $V_{PP}$  is below or equal to  $V_{PPLK}$ , any Program or Erase operation will result in an error, prompting the corresponding Status Register bit (SR[3]) to be set.

Figure 16. Example Power Supply Configurations



0645\_06

**Note:**

1. A resistor can be used if the  $V_{CC}$  supply can sink adequate current based on resistor value. See AP-657 *Designing with the Advanced+ Boot Block Flash Memory Architecture* for details.

## Appendix A Write State Machine States

Table 25 and Table 26 show the Write State Machine command state transitions based on incoming commands.

**Table 25. Write State Machine States**

| Current State           | SR.7 | Data When Read | Command Input (and Next State)         |                            |                     |                     |                        |                      |                     |                      |  |
|-------------------------|------|----------------|--|----------------------------|---------------------|---------------------|------------------------|----------------------|---------------------|----------------------|--|
|                         |      |                | Read Array (FFH)                       | Program Setup (10/40H)     | Erase Setup (20H)   | Erase Confirm (D0H) | Prog/Ers Suspend (B0H) | Prog/Ers Resume (D0) | Read Status (70H)   | Clear Status (50H)   |  |
| Read Array              | "1"  | Array          | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Read Status             | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Read Config.            | "1"  | Config         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Read Query              | "1"  | CFI            | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Lock Setup              | "1"  | Status         | Lock Command Error                     |                            |                     | Lock (Done)         | Lock Cmd. Error        | Lock (Done)          | Lock Cmd. Error     |                      |  |
| Lock Cmd. Error         | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Lock Oper. (Done)       | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Prot. Prog. Setup       | "1"  | Status         | Protection Register Program            |                            |                     |                     |                        |                      |                     |                      |  |
| Prot. Prog. (Not Done)  | "0"  | Status         | Protection Register Program (Not Done) |                            |                     |                     |                        |                      |                     |                      |  |
| Prot. Prog. (Done)      | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |
| Prog. Setup             | "1"  | Status         | Program                                |                            |                     |                     |                        |                      |                     |                      |  |
| Program (Not Done)      | "0"  | Status         | Program (Not Done)                     |                            |                     |                     | Prog. Sus. Status      | Program (Not Done)   |                     |                      |  |
| Prog. Susp. Status      | "1"  | Status         | Prog. Sus. Read Array                  | Program Suspend Read Array |                     | Prog. (Not Done)    | Prog. Sus. Rd. Array   | Program (Not Done)   | Prog. Sus. Status   | Prog. Sus. Rd. Array |  |
| Prog. Susp. Read Array  | "1"  | Array          | Prog. Sus. Read Array                  | Program Suspend Read Array |                     | Prog. (Not Done)    | Prog. Sus. Rd. Array   | Program (Not Done)   | Prog. Sus. Status   | Prog. Sus. Rd. Array |  |
| Prog. Susp. Read Config | "1"  | Config         | Prog. Sus. Read Array                  | Program Suspend Read Array |                     | Prog. (Not Done)    | Prog. Sus. Rd. Array   | Program (Not Done)   | Prog. Sus. Status   | Prog. Sus. Rd. Array |  |
| Prog. Susp. Read Query  | "1"  | CFI            | Prog. Sus. Read Array                  | Program Suspend Read Array |                     | Prog. (Not Done)    | Prog. Sus. Rd. Array   | Program (Not Done)   | Prog. Sus. Status   | Prog. Sus. Rd. Array |  |
| Program (Done)          | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Status         | Read Array           |  |
| Erase Setup             | "1"  | Status         | Erase Command Error                    |                            |                     | Erase (Not Done)    | Erase Cmd. Error       | Erase (Not Done)     | Erase Command Error |                      |  |
| Erase Cmd. Error        | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Status         | Read Array           |  |
| Erase (Not Done)        | "0"  | Status         | Erase (Not Done)                       |                            |                     |                     | Erase Sus. Status      | Erase (Not Done)     |                     |                      |  |
| Ers. Susp. Status       | "1"  | Status         | Erase Sus. Read Array                  | Prog. Setup                | Ers. Sus. Rd. Array | Erase               | Ers. Sus. Rd. Array    | Erase                | Erase Sus. Status   | Ers. Sus. Rd. Array  |  |
| Erase Susp. Array       | "1"  | Array          | Erase Sus. Read Array                  | Prog. Setup                | Ers. Sus. Rd. Array | Erase               | Ers. Sus. Rd. Array    | Erase                | Erase Sus. Status   | Ers. Sus. Rd. Array  |  |
| Ers. Susp. Read Config  | "1"  | Config         | Erase Sus. Read Array                  | Prog. Setup                | Ers. Sus. Rd. Array | Erase               | Ers. Sus. Rd. Array    | Erase                | Erase Sus. Status   | Ers. Sus. Rd. Array  |  |
| Ers. Susp. Read Query   | "1"  | CFI            | Erase Sus. Read Array                  | Prog. Setup                | Ers. Sus. Rd. Array | Erase               | Ers. Sus. Rd. Array    | Erase                | Erase Sus. Status   | Ers. Sus. Rd. Array  |  |
| Erase (Done)            | "1"  | Status         | Read Array                             | Prog. Setup                | Ers. Setup          | Read Array          |                        |                      | Read Sts.           | Read Array           |  |

**Table 26. Write State Machine States, Continued**

| Current State            | Command Input (and Next State)         |                          |                            |                          |                       |                         |                      |
|--------------------------|--|--------------------------|----------------------------|--------------------------|-----------------------|-------------------------|----------------------|
|                          | Read Config (90H)                      | Read Query (98H)         | Lock Setup (60H)           | Prot. Prog. Setup (C0H)  | Lock Confirm (01H)    | Lock Down Confirm (2FH) | Unlock Confirm (D0H) |
| Read Array               | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Read Status              | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Read Config.             | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Read Query               | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Lock Setup               | Locking Command Error                  |                          |                            |                          | Lock Operation (Done) |                         |                      |
| Lock Cmd. Error          | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Lock Oper. (Done)        | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Prot. Prog. Setup        | Protection Register Program            |                          |                            |                          |                       |                         |                      |
| Prot. Prog. (Not Done)   | Protection Register Program (Not Done) |                          |                            |                          |                       |                         |                      |
| Prot. Prog. (Done)       | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Prog. Setup              | Program                                |                          |                            |                          |                       |                         |                      |
| Program (Not Done)       | Program (Not Done)                     |                          |                            |                          |                       |                         |                      |
| Prog. Susp. Status       | Prog. Susp. Read Config.               | Prog. Susp. Read Query   | Program Suspend Read Array |                          |                       |                         | Program (Not Done)   |
| Prog. Susp. Read Array   | Prog. Susp. Read Config.               | Prog. Susp. Read Query   | Program Suspend Read Array |                          |                       |                         | Program (Not Done)   |
| Prog. Susp. Read Config. | Prog. Susp. Read Config.               | Prog. Susp. Read Query   | Program Suspend Read Array |                          |                       |                         | Program (Not Done)   |
| Prog. Susp. Read Query.  | Prog. Susp. Read Config.               | Prog. Susp. Read Query   | Program Suspend Read Array |                          |                       |                         | Program (Not Done)   |
| Program (Done)           | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Erase Setup              | Erase Command Error                    |                          |                            |                          |                       |                         | Erase (Not Done)     |
| Erase Cmd. Error         | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |
| Erase (Not Done)         | Erase (Not Done)                       |                          |                            |                          |                       |                         |                      |
| Erase Susp. Status       | Ers. Susp. Read Config.                | Erase Suspend Read Query | Lock Setup                 | Erase Suspend Read Array |                       |                         | Erase (Not Done)     |
| Erase Suspend Array      | Ers. Susp. Read Config.                | Erase Suspend Read Query | Lock Setup                 | Erase Suspend Read Array |                       |                         | Erase (Not Done)     |
| Eras Sus. Read Config    | Erase Suspend Read Config.             | Erase Suspend Read Query | Lock Setup                 | Erase Suspend Read Array |                       |                         | Erase (Not Done)     |
| Eras Sus. Read Query     | Erase Suspend Read Config.             | Erase Suspend Read Query | Lock Setup                 | Erase Suspend Read Array |                       |                         | Erase (Not Done)     |
| Ers.(Done)               | Read Config.                           | Read Query               | Lock Setup                 | Prot. Prog. Setup        | Read Array            |                         |                      |

## Appendix B Flow Charts

Figure 17. Word Program Flowchart

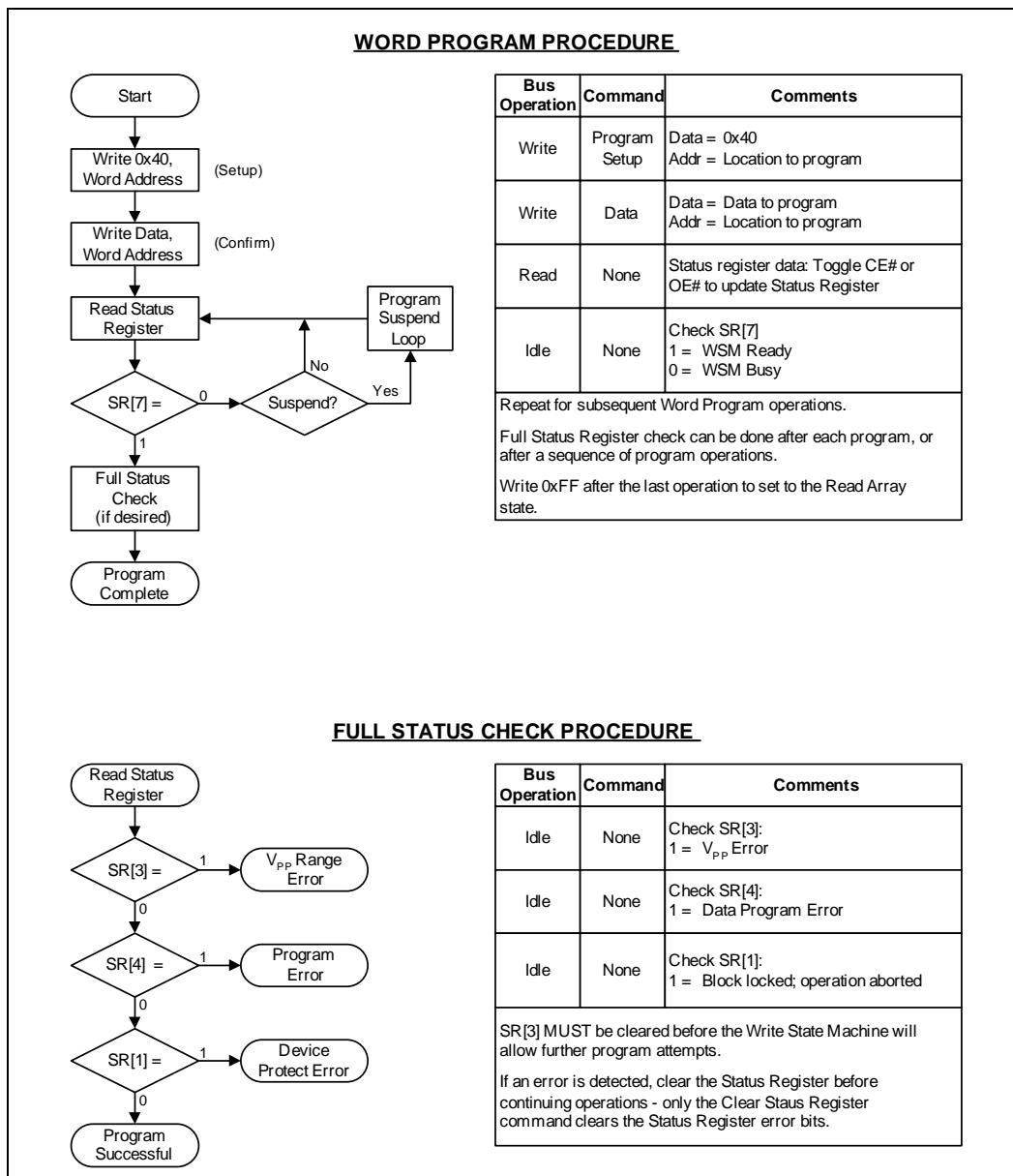




Figure 18. Program Suspend / Resume Flowchart

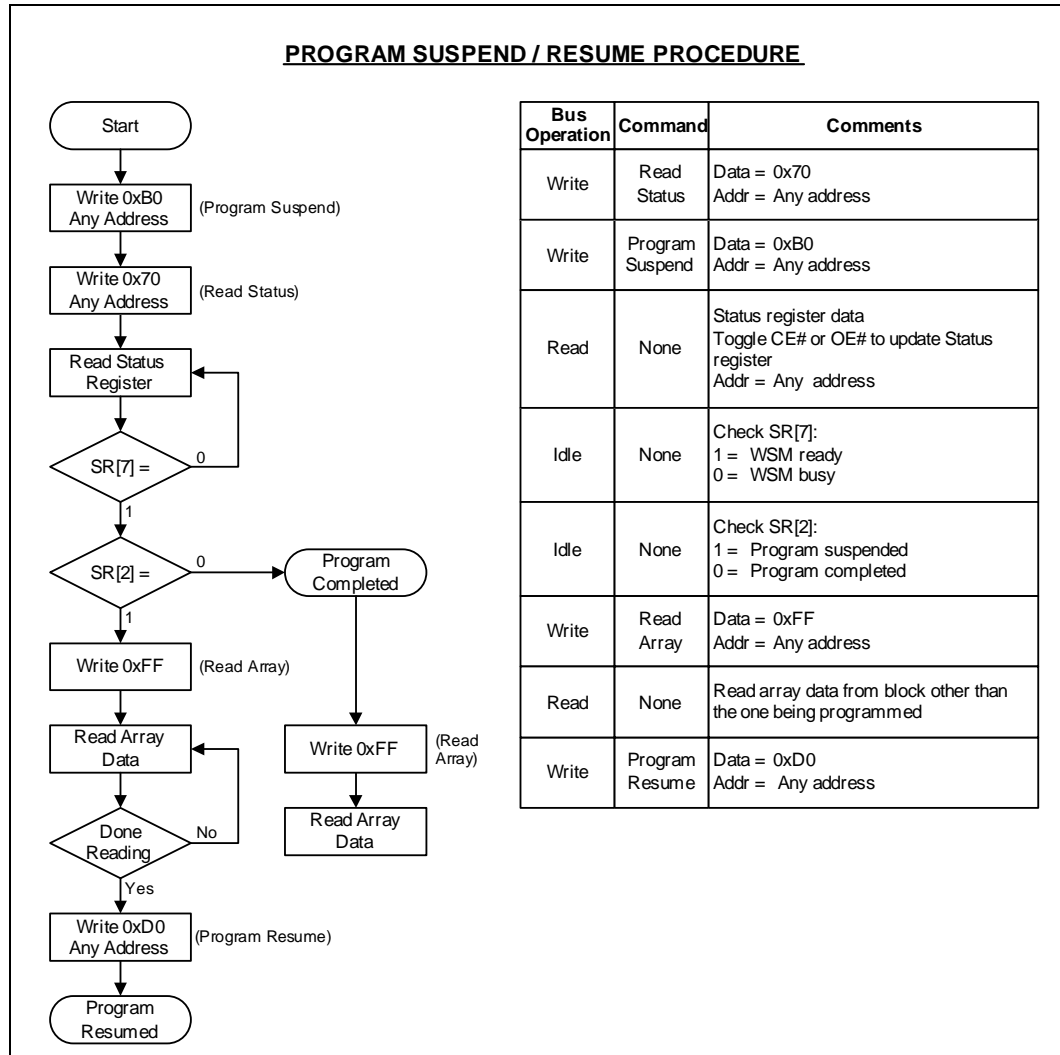


Figure 19. Erase Suspend / Resume Flowchart

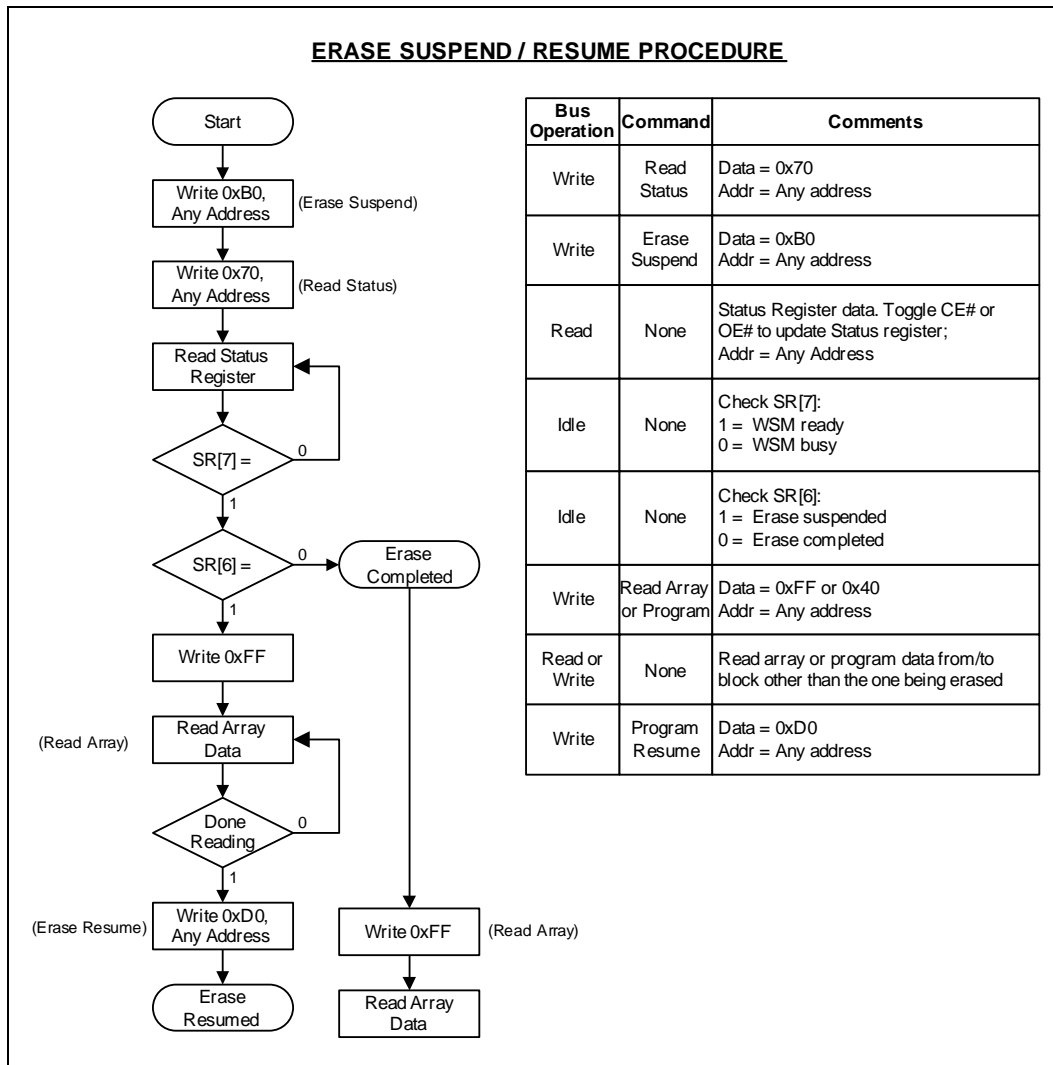


Figure 20. Block Erase Flowchart

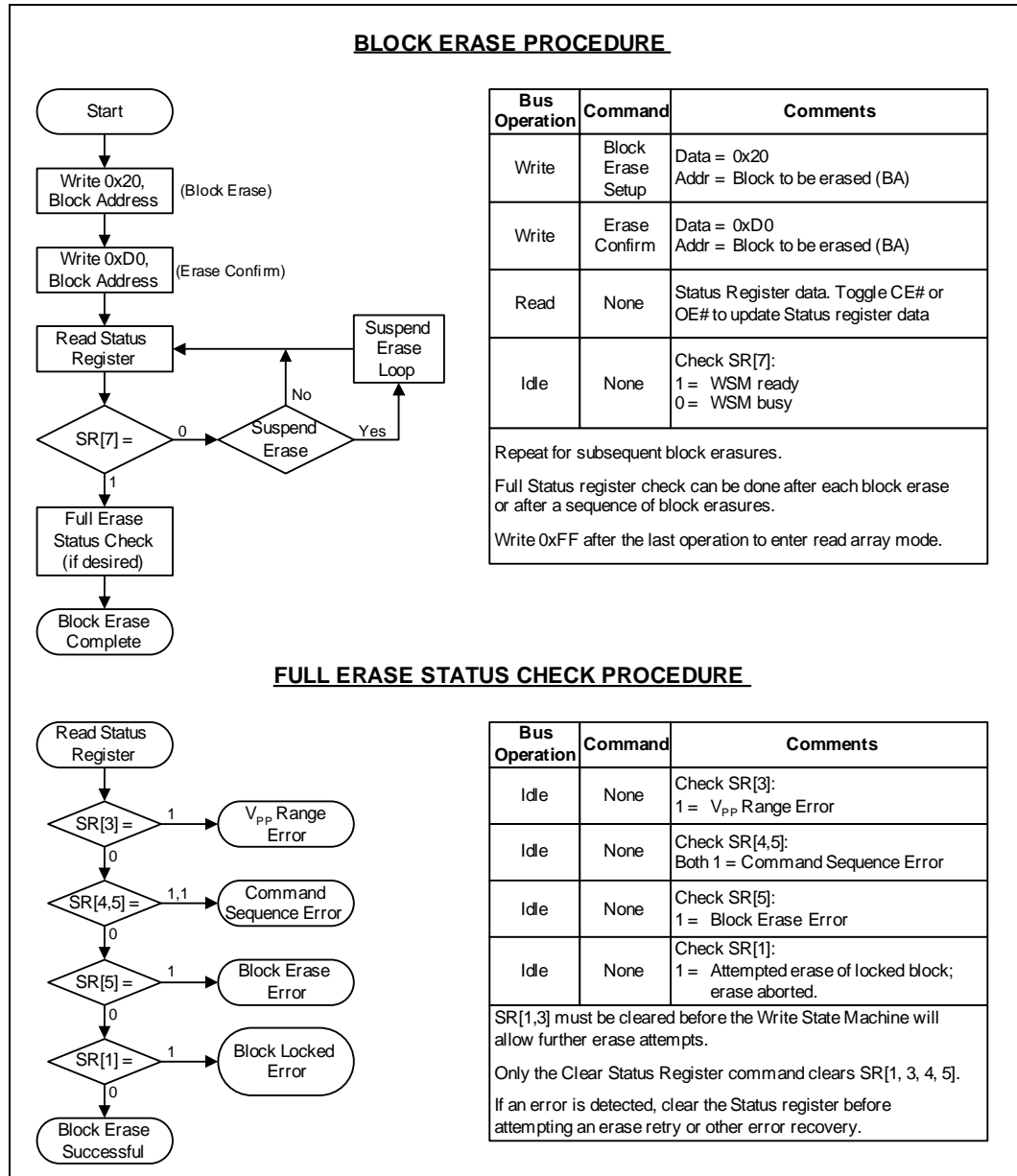


Figure 21. Locking Operations Flowchart

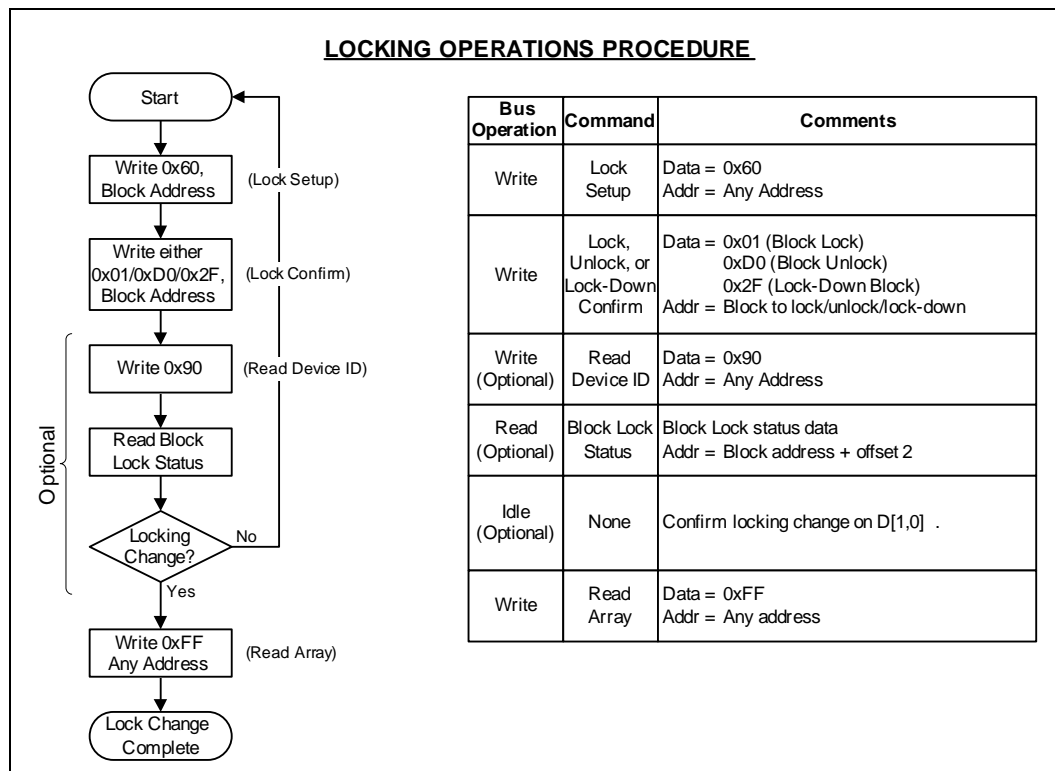
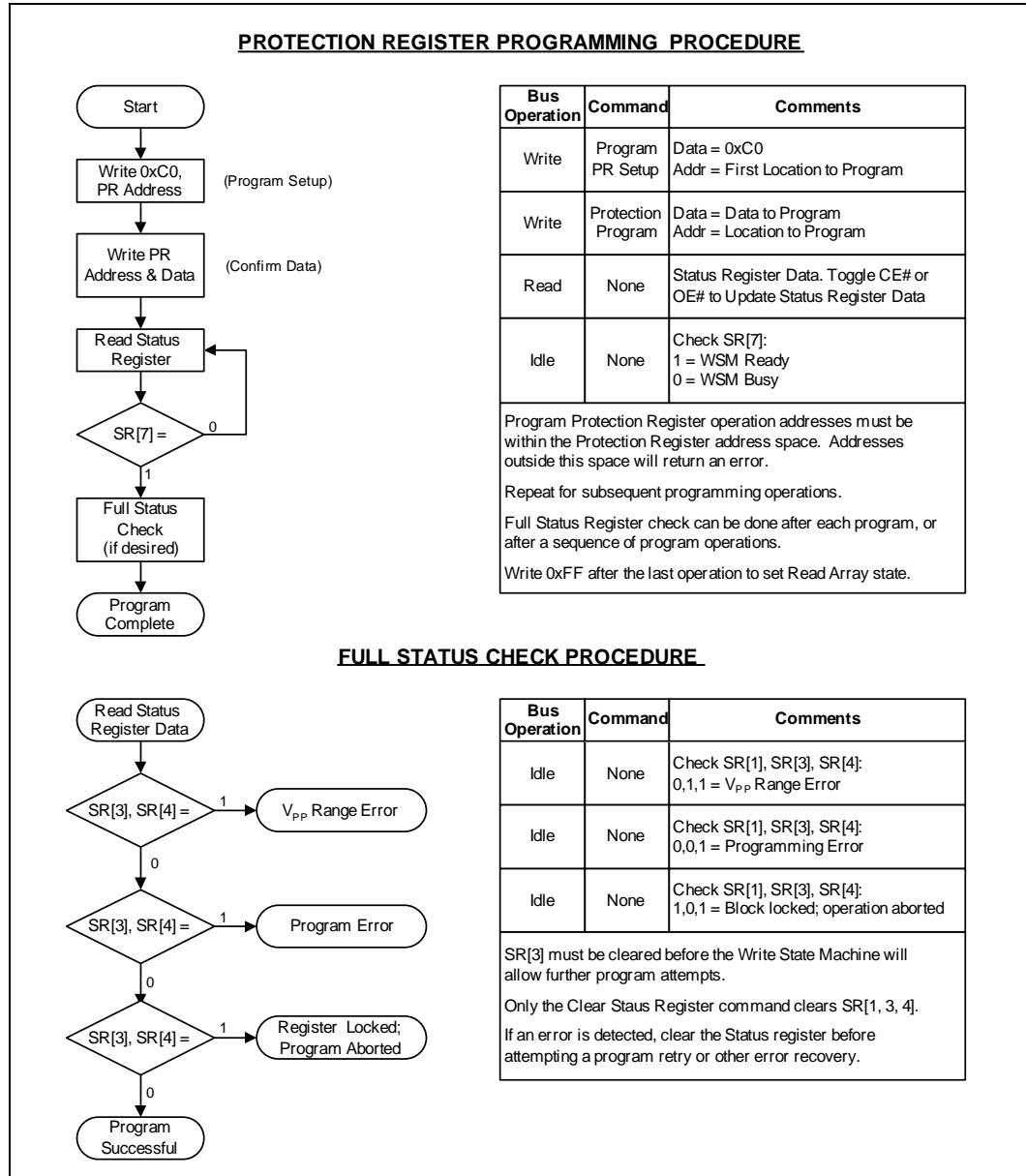


Figure 22. Protection Register Programming Flowchart



## Appendix C Common Flash Interface

This appendix defines the data structure or “database” returned by the Common Flash Interface (CFI) Query command. System software should parse this structure to gain critical information such as block size, density, x8/x16, and electrical specifications. Once this information has been obtained, the software detects which command sets to use to enable flash writes, block erases, and otherwise control the flash component. The Query is part of an overall specification for multiple command set and control interface descriptions called Common Flash Interface, or CFI.

### C.1 Query Structure Output

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

Query data are presented on the lowest-order data outputs (DQ0-DQ7) 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 0x10, 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 0x10 and 0x11. This CFI-compliant device outputs 0x00 data on upper bytes. The device outputs ASCII “Q” in the low byte (DQ0-DQ7) and 0x00 in the high byte (DQ8-DQ15).

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.

For tables in this appendix, 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 “0x00,” the leading “00” has been dropped from the table notation and only the lower byte value is shown. Any x16 device outputs can be assumed to have 0x00 on the upper byte in this mode.

**Table 27. 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 28. Example of Query Structure Output of x16 Devices**

| Word Addressing: |          |           |
|------------------|----------|-----------|
| Offset           | Hex Code | Value     |
| A[X-0]           | DQ[16:0] |           |
| 0x00010          | 0051     | "Q"       |
| 0x00011          | 0052     | "R"       |
| 0x00012          | 0059     | "Y"       |
| 0x00013          | P_IDLO   | PrVendor  |
| 0x00014          | P_IDHI   | ID #      |
| 0x00015          | PLO      | PrVendor  |
| 0x00016          | PHI      | TblAdr    |
| 0x00017          | A_IDLO   | AltVendor |
| 0x00018          | A_IDHI   | ID #      |
| ...              | ...      | ...       |

## C.2 Query Structure Overview

The Query command causes the flash component to display the Common Flash Interface (CFI) Query structure or “database.” Table 29 summarizes the structure sub-sections and address locations.

**Table 29. Query Structure**

| Offset                | Sub-Section Name                            | Description <sup>1</sup>   |
|-----------------------|---|--|
| 0x00000               |   | Manufacturer Code  |
| 0x00001               |   | Device Code  |
| 0x(BA+2) <sup>2</sup> | Block Status register                       | Block-specific information   |
| 0x00004-0xF           | Reserved                                    | Reserved for vendor-specific information                                       |
| 0x00010               | CFI query identification string             | Command set ID and vendor data offset  |
| 0x0001B               | System interface information                | Device timing & voltage information  |
| 0x00027               | Device geometry definition                  | Flash device layout  |
| P <sup>3</sup>        | Primary Intel-specific Extended Query Table | Vendor-defined additional information specific to the Primary Vendor Algorithm |

**Notes:**

1. Refer to the Query Structure Output section and offset 0x28 for the detailed definition of offset address as a function of device bus width and mode.
2. BA = Block Address beginning location (i.e., 0x08000 is block 1's beginning location when the block size is 32K-word).
3. Offset 15 defines “P” which points to the Primary Intel-specific Extended Query Table.

## C.3 Block Status Register

The Block Status Register indicates whether an erase operation completed successfully or whether a given block is locked or can be accessed for flash program/erase operations. See [Table 30](#).

Block Erase Status (BSR[1]) allows system software to determine the success of the last block erase operation. BSR[1] can be used just after power-up to verify that the VCC supply was not accidentally removed during an erase operation.

**Table 30. Block Status Register**

| Offset                | Length | Description   | Add. | Value           |
|-----------------------|--------|---|------|-----------------|
| 0x(BA+2) <sup>1</sup> | 1      | Block Lock Status Register  | BA+2 | --00 or --01    |
|                       |        | BSR[0] Block lock status<br>0 = Unlocked<br>1 = Locked                  | BA+2 | (bit 0): 0 or 1 |
|                       |        | BSR[1] Block lock-down status<br>0 = Not locked down<br>1 = Locked down | BA+2 | (bit 1): 0 or 1 |
|                       |        | BSR[7:2]: <i>Reserved for future use</i>                                | BA+2 | (bit 2-7): 0    |

**Notes:**

1. BA = Block Address beginning location (i.e., 0x08000 is block 1's beginning location when the block size is 32K-word).



## C.4 CFI Query Identification String

The Identification String provides verification that the component supports the Common Flash Interface specification. It also indicates the specification version and supported vendor-specified command set(s). See [Table 31](#).

**Table 31. CFI Identification**

| Offset | Length | Description  | Add.              | Hex Code             | Value             |
|--------|--------|--|-------------------|----------------------|-------------------|
| 0x10   | 3      | Query-unique ASCII string "QRY"  | 10:<br>11:<br>12: | --51<br>--52<br>--59 | "Q"<br>"R"<br>"Y" |
| 0x13   | 2      | Primary vendor command set and control interface ID code<br>16-bit ID code for vendor-specified algorithms             | 13:<br>14:        | --03<br>--00         |                   |
| 0x15   | 2      | Extended Query Table primary algorithm address   | 15:<br>16:        | --35<br>--00         |                   |
| 0x17   | 2      | Alternate vendor command set and control interface ID code<br>0x0000 means no second vendor-specified algorithm exists | 17:<br>18:        | --00<br>--00         |                   |
| 0x19   | 2      | Secondary algorithm Extended Query Table address<br>0x0000 means none exists   | 19:<br>1A:        | --00<br>--00         |                   |

Table 32. System Interface Information

| Offset | Length | Description   | Add. | Hex Code | Value  |
|--------|--------|---|------|----------|--------|
| 0x1B   | 1      | V <sub>CC</sub> logic supply minimum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 BCD volts         | 1B:  | --27     | 2.7 V  |
| 0x1C   | 1      | V <sub>CC</sub> logic supply maximum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 BCD volts         | 1C:  | --36     | 3.6 V  |
| 0x1D   | 1      | V <sub>PP</sub> [programming] supply minimum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 HEX volts | 1D:  | --B4     | 11.4 V |
| 0x1E   | 1      | V <sub>PP</sub> [programming] supply maximum program/erase voltage<br>bits 0–3 BCD 100 mV<br>bits 4–7 HEX volts | 1E:  | --C6     | 12.6 V |
| 0x1F   | 1      | "n" such that typical single word program time-out = 2 <sup>n</sup> μs  | 1F:  | --05     | 32 μs  |
| 0x20   | 1      | "n" such that typical max. buffer write time-out = 2 <sup>n</sup> μs  | 20:  | --00     | NA     |
| 0x21   | 1      | "n" such that typical block erase time-out = 2 <sup>n</sup> ms  | 21:  | --0A     | 1 s    |
| 0x22   | 1      | "n" such that typical full chip erase time-out = 2 <sup>n</sup> ms  | 22:  | --00     | NA     |
| 0x23   | 1      | "n" such that maximum word program time-out = 2 <sup>n</sup> times typical                                      | 23:  | --04     | 512 μs |
| 0x24   | 1      | "n" such that maximum buffer write time-out = 2 <sup>n</sup> times typical                                      | 24:  | --00     | NA     |
| 0x25   | 1      | "n" such that maximum block erase time-out = 2 <sup>n</sup> times typical                                       | 25:  | --03     | 8s     |
| 0x26   | 1      | "n" such that maximum chip erase time-out = 2 <sup>n</sup> times typical  | 26:  | --00     | NA     |

## C.5 Device Geometry Definition

Table 33. Device Geometry Definition

| Offset | Length | Description   | Add.                     | Hex Code   | Value |
|--------|--------|---|--------------------------|--|-------|
| 0x27   | 1      | "n" such that device size = 2 <sup>n</sup> in number of bytes   | 27                       | See Table 34, "Device Geometry Details" on page 67 |       |
| 0x28   | 2      | Flash device interface: <u>x8_async</u> <u>x16_async</u> <u>x8/x16_async</u><br>28:00,29:00    28:01,29:00    28:02,29:00   | 28:<br>29:               | --01<br>--00                                       | x16   |
| 0x2A   | 2      | "n" such that maximum number of bytes in write buffer = 2 <sup>n</sup>  | 2A:<br>2B:               | --00<br>--00                                       | 0     |
| 0x2C   | 1      | Number of erase block regions within device:<br>1. x = 0 means no erase blocking; the device erases in "bulk"<br>2. x specifies the number of device or partition regions with one or more contiguous same-size erase blocks.<br>3. Symmetrically blocked partitions have one blocking region<br>4. Partition size = (total blocks) x (individual block size) | 2C:                      | --02   | 2     |
| 0x2D   | 4      | Erase Block Region 1 Information<br>bits 0–15 = y, y+1 = number of identical-size erase blocks<br>bits 16–31 = z, region erase block(s) size are z x 256 bytes  | 2D:<br>2E:<br>2F:<br>30: | See Table 34, "Device Geometry Details" on page 67 |       |
| 0x2D   | 14     | Erase Block Region 2 Information<br>bits 0–15 = y, y+1 = number of identical-size erase blocks<br>bits 16–31 = z, region erase block(s) size are z x 256 bytes  | 31:<br>32:<br>33:<br>34: | See Table 34, "Device Geometry Details" on page 67 |       |

**Table 34. Device Geometry Details**

| Address | 16 Mbit |      | 32 Mbit |      | 64 Mbit |      |
|---------|---------|------|---------|------|---------|------|
|         | -B      | -T   | -B      | -T   | -B      | -T   |
| 0x27    | --15    | -15  | --16    | -16  | --17    | --17 |
| 0x28    | --01    | --01 | --01    | --01 | --01    | --01 |
| 0x29    | --00    | --00 | --00    | -00  | -00     | -00  |
| 0x2A    | --00    | --00 | --00    | -00  | -00     | -00  |
| 0x2B    | --00    | --00 | --00    | -00  | -00     | -00  |
| 0x2C    | --02    | --02 | --02    | --02 | --02    | --02 |
| 0x2D    | --07    | --1E | --07    | --3E | --07    | --7E |
| 0x2E    | --00    | --00 | --00    | -00  | -00     | -00  |
| 0x2F    | --20    | --00 | --20    | -00  | --20    | --00 |
| 0x30    | --00    | --01 | --00    | --01 | --00    | --01 |
| 0x31    | --1E    | --07 | --3E    | --07 | --7E    | --07 |
| 0x32    | --00    | --00 | --00    | -00  | -00     | -00  |
| 0x33    | --00    | --20 | --00    | --20 | --00    | --20 |
| 0x34    | --01    | --00 | --01    | --00 | --01    | --00 |

## C.6 Intel-Specific Extended Query Table

Certain flash features and commands are optional as shown in Table 35, “Primary-Vendor Specific Extended Query” on page 68. The Intel-specific Extended Query table specifies these features as well as other similar types of information.

**Table 35. Primary-Vendor Specific Extended Query**

| Offset <sup>1</sup><br>P = 0x15          | Length | Description<br>(Optional Flash Features and Commands)   | Address   | Hex Code   | Value             |
|--|--------|---|---|--|-------------------|
| 0x(P+0)<br>0x(P+1)<br>0x(P+2)            | 3      | Primary extended query table<br>Unique ASCII string “PR”  | 35:<br>36:<br>37:   | --50<br>--52<br>--49                                   | “P”<br>“R”<br>“I” |
| 0x(P+3)                                  | 1      | Major version number, ASCII   | 38:   | --31   | “1”               |
| 0x(P+4)                                  | 1      | Minor version number, ASCII   | 39:   | --30   | “0”               |
| 0x(P+5)<br>0x(P+6)<br>0x(P+7)<br>0x(P+8) | 4      | Optional feature and command support (1=yes, 0=no)<br>bits 9–31 are reserved; undefined bits are “0.” If bit 31 is “1” then another 31 bit field of optional features follows at the end of the bit-30 field.   | 3A:<br>3B:<br>3C:<br>3D:  | --66<br>--00<br>--00<br>--00                           |                   |
|  |        | bit 0 Chip erase supported<br>bit 1 Suspend erase supported<br>bit 2 Suspend program supported<br>bit 3 Legacy lock/unlock supported<br>bit 4 Queued erase supported<br>bit 5 Instant individual block locking supported<br>bit 6 Protection bits supported<br>bit 7 Page mode read supported<br>bit 8 Synchronous read supported | bit 0 = 0<br>bit 1 = 1<br>bit 2 = 1<br>bit 3 = 0<br>bit 4 = 0<br>bit 5 = 1<br>bit 6 = 1<br>bit 7 = 0<br>bit 8 = 0 | No<br>Yes<br>Yes<br>No<br>No<br>Yes<br>Yes<br>No<br>No |                   |
| 0x(P+9)                                  | 1      | Supported functions after suspend: Read Array, Status, Query<br>Other supported operations are:<br><i>bits 1–7 reserved; undefined bits are “0”</i>   | 3E:   | --01   |                   |
|  |        | bit 0 Program supported after erase suspend   | bit 0 = 1   |  | Yes               |
| 0x(P+A)<br>0x(P+B)                       | 2      | Block Status Register mask<br>bits 2–15 are Reserved; undefined bits are “0”<br>bit 0 Block Lock-Bit Status Register active<br>bit 1 Block Lock-Down Bit Status active  | 3F:<br>40:  | --03<br>--00   |                   |
|  |        |   | bit 0 = 1<br>bit 1 = 1  | Yes<br>Yes   |                   |
| 0x(P+C)                                  | 1      | V <sub>CC</sub> logic supply highest performance program/erase voltage<br>bits 0–3 BCD value in 100 mV<br>bits 4–7 BCD value in volts   | 41:   | --33   | 3.3 V             |
| 0x(P+D)                                  | 1      | V <sub>PP</sub> optimum program/erase supply voltage<br>bits 0–3 BCD value in 100 mV<br>bits 4–7 HEX value in volts   | 42:   | --C0   | 12.0 V            |

**Notes:**

- The variable P is a pointer which is defined at CFI offset 0x15.

**Table 36. Protection Register Information**

| Offset <sup>1</sup><br>P = 0x35 | Length | Description<br>(Optional Flash Features and Commands)   | Address | Hex Code | Value  |
|---------------------------------|--------|---|---------|----------|--------|
| 0x(P+E)                         | 1      | Number of Protection register fields in JEDEC ID space. "00h," indicates that 256 protection bytes are available  | 43:     | --01     | 01     |
| 0x(P+F)<br>0x(P+10)<br>(0xP+11) | 4      | Protection Field 1: Protection Description<br>This field describes user-available One Time Programmable (OTP) Protection register bytes. Some are pre-programmed with device-unique serial numbers. Others are user programmable. Bits 0–15 point to the Protection register Lock byte, the section's first byte. The following bytes are factory pre-programmed and user-programmable.<br>bits 0–7 = Lock/bytes JEDEC-plane physical low address<br>bits 8–15 = Lock/bytes JEDEC -plane physical high address<br>bits 16–23 = "n" such that 2 <sup>n</sup> = factory pre-programmed bytes<br>bits 24–31 = "n" such that 2 <sup>n</sup> = user programmable bytes | 44:     | --80     | 80h    |
| 0x(P+12)                        |        |   | 45:     | --00     | 00h    |
|                                 |        |   | 46:     | --03     | 8 byte |
|                                 |        |   | 47:     | --03     | 8 byte |
| 0x(P+13)                        |        | Reserved for future use   | 48:     |          |        |

**Notes:**

1. The variable P is a pointer which is defined at CFI offset 0x15.

## Appendix D Additional Information

| Order Number                      | Document/Tool   |
|-----------------------------------|---|
| 297938                            | <i>3 Volt Advanced+ Boot Block Flash Memory Specification Update</i>            |
| 292216                            | <i>AP-658 Designing for Upgrade to the Advanced+ Boot Block Flash Memory</i>    |
| 292215                            | <i>AP-657 Designing with the Advanced+ Boot Block Flash Memory Architecture</i> |
| Contact your Intel Representative | <i>Intel® Flash Data Integrator (Intel® FDI) Software Developer's Kit</i>       |
| 297874                            | <i>IFDI Interactive: Play with Intel® Flash Data Integrator on Your PC</i>      |

**Notes:**

1. Call the Intel Literature Center at (800) 548-4725 to request Intel documentation. International customers should contact their local Intel or distribution sales office.
2. See the Intel page at '<http://www.intel.com/design/flash>' for technical documentation and tools.

## Appendix E Ordering Information

Figure 23. Component Ordering Information

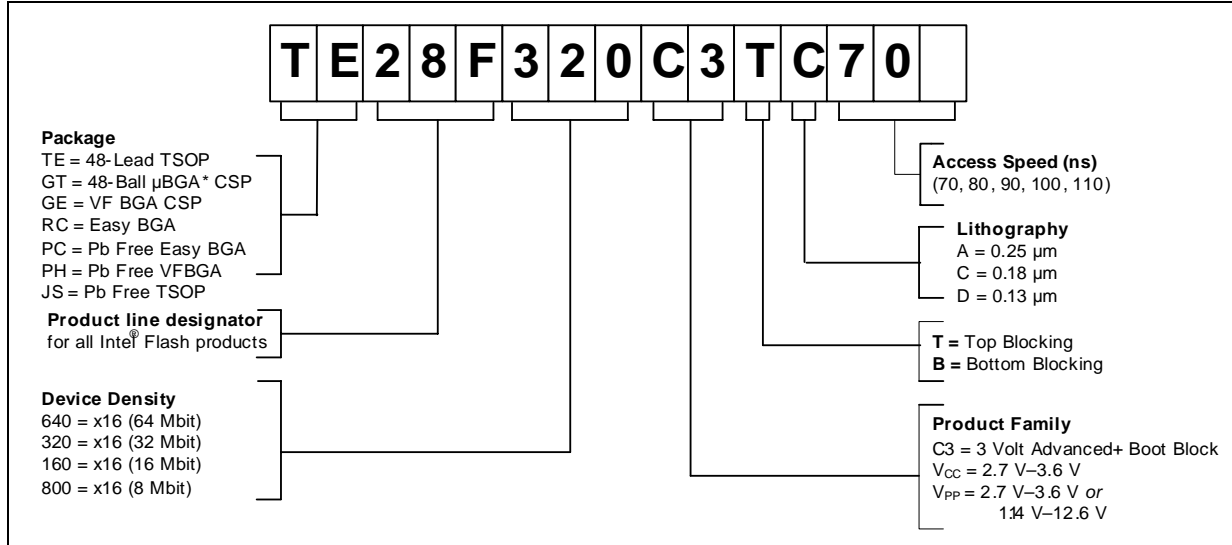


Table 37. Product Information Ordering Matrix

| VALID COMBINATIONS (All Extended Temperature) |  |  |  |  |
|---|--|--|--|--|
|   | 48-Lead TSOP   | 48-Ball $\mu$ BGA* CSP   | 48-Ball VF BGA   | Easy BGA   |
| Extended<br>64 Mbit                           |  |  |  |  |
| Extended<br>32 Mbit                           | TE28F320C3TD70<br>TE28F320C3BD70<br>TE28F320C3TC70<br>TE28F320C3BC70<br>TE28F320C3TC90<br>TE28F320C3BC90<br>TE28F320C3TA100<br>TE28F320C3BA100<br>TE28F320C3TA110<br>TE28F320C3BA110<br>JS28F320C3BD70<br>JS28F320C3TD70<br>JS28F320C3BD90<br>JS28F320C3TD90 | GT28F320C3TA100<br>GT28F320C3BA100<br>GT28F320C3TA110<br>GT28F320C3BA110 | GE28F320C3TD70<br>GE28F320C3BD70<br>GE28F320C3TC70<br>GE28F320C3BC70<br>GE28F320C3TC90<br>GE28F320C3BC90<br>PH28F320C3BD70<br>PH28F320C3TD70<br>PH28F320C3BD90<br>PH28F320C3TD90   | RC28F320C3TD70<br>RC28F320C3BD70<br>RC28F320C3TD90<br>RC28F320C3BD90<br>RC28F320C3TC90<br>RC28F320C3BC90<br>RC28F320C3TA100<br>RC28F320C3BA100<br>RC28F320C3TA110<br>RC28F320C3BA110<br>PC28F320C3BD70<br>PC28F320C3TD70<br>PC28F320C3BD90<br>PC28F320C3TD90 |
| Extended<br>16 Mbit                           | TE28F160C3TD70<br>TE28F160C3BD70<br>TE28F160C3TC70<br>TE28F160C3BC70<br>TE28F160C3TC80<br>TE28F160C3BC80<br>TE28F160C3TC90<br>TE28F160C3BC90<br>TE28F160C3TA90<br>TE28F160C3BA90<br>TE28F160C3TA110<br>TE28F160C3BA110<br>JS28F160C3BD70<br>JS28F160C3TD70   | GT28F160C3TA90<br>GT28F160C3BA90<br>GT28F160C3TA110<br>GT28F160C3BA110   | GE28F160C3TD70<br>GE28F160C3BD70<br>GE28F160C3TC70<br>GE28F160C3BC70<br>GE28F160C3TC80<br>GE28F160C3BC80<br>GE28F160C3TC90<br>GE28F160C3BC90<br>GE28F160C3TA90<br>GE28F160C3BA90<br>GE28F160C3TA110<br>GE28F160C3BA110<br>PH28F160C3BD70<br>PH28F160C3TD70 | RC28F160C3TD70<br>RC28F160C3BD70<br>RC28F160C3TC70<br>RC28F160C3BC70<br>RC28F160C3TC80<br>RC28F160C3BC80<br>RC28F160C3TC90<br>RC28F160C3BC90<br>RC28F160C3TA90<br>RC28F160C3BA90<br>RC28F160C3TA110<br>RC28F160C3BA110<br>PC28F160C3BD70<br>PC28F160C3TD70   |
| Extended<br>8 Mbit                            | TE28F800C3TD70<br>TE28F800C3BD70<br>TE28F800C3TA90<br>TE28F800C3BA90<br>TE28F800C3TA110<br>TE28F800C3BA110<br>JS28F800C3BD70<br>JS28F800C3TD70   |  |  | RC28F800C3TD70<br>RC28F800C3BD70<br>RC28F800C3TA90<br>RC28F800C3BA90<br>RC28F800C3TA110<br>RC28F800C3BA110<br>PC28F800C3BD70<br>PC28F800C3TD70   |

**Note:** The second line of the 48-ball  $\mu$ BGA package top side mark specifies assembly codes. For samples only, the first character signifies either "E" for engineering samples or "S" for silicon daisy chain samples. All other assembly codes without an "E" or "S" as the first character are production units.