



3 VOLT ADVANCED+ BOOT BLOCK

8-, 16-, 32-MBIT

FLASH MEMORY FAMILY

28F008C3, 28F016C3, 28F032C3

28F800C3, 28F160C3, 28F320C3

- **Flexible SmartVoltage Technology**
 - 2.7 V–3.6 V Read/Program/Erase
 - 2.7 V or 1.65 V I/O Option Reduces Overall System Power
 - 12 V for Fast Production Programming
- **High Performance**
 - 2.7 V–3.6 V: 90 ns Max Access Time
 - 3.0 V–3.6 V: 80 ns Max Access Time
- **Optimized Architecture for Code Plus Data Storage**
 - Eight 8-Kbyte Blocks, Top or Bottom Locations
 - Up to Sixty-Three 64-KB Blocks
 - Fast Program Suspend Capability
 - Fast Erase Suspend Capability
- **Flexible Block Locking**
 - Lock/Unlock Any Block
 - Full Protection on Power-Up
 - WP# Pin for Hardware Block Protection
 - V_{PP} = GND Option
 - V_{CC} Lockout Voltage
- **Low Power Consumption**
 - 9 mA Typical Read Power
 - 10 μ A Typical Standby Power with Automatic Power Savings Feature
- **Extended Temperature Operation**
 - –40 °C to +85 °C
- **Easy-12 V**
 - Faster Production Programming
 - No Additional System Logic
- **128-bit Protection Register**
 - 64-bit Unique Device Identifier
 - 64-bit User Programmable OTP Cells
- **Extended Cycling Capability**
 - Minimum 100,000 Block Erase Cycles
- **Flash Data Integrator Software**
 - Flash Memory Manager
 - System Interrupt Manager
 - Supports Parameter Storage, Streaming Data (e.g., voice)
- **Automated Word/Byte Program and Block Erase**
 - Command User Interface
 - Status Registers
- **SRAM-Compatible Write Interface**
- **Cross-Compatible Command Support**
 - Intel Basic Command Set
 - Common Flash Interface
- **x 16 for High Performance**
 - 48-Ball μ BGA* Package
 - 48-Lead TSOP Package
- **x 8 I/O for Space Savings**
 - 48-Ball μ BGA* Package
 - 40-Lead TSOP Package
- **0.25 μ ETOX™ VI Flash Technology**

The 0.25 μ m 3 Volt Advanced+ Boot Block, manufactured on Intel's latest 0.25 μ technology, represents a feature-rich solution at overall lower system cost. Smart 3 flash memory devices incorporate low voltage capability (2.7 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-developed Flash Data Integrator (FDI) software and you have a cost-effective, flexible, monolithic code plus data storage solution on the market today. 3 Volt Advanced+ Boot Block products will be available in 48-lead TSOP, 40-lead TSOP, and 48-ball μ BGA* packages. Additional information on this product family can be obtained by accessing Intel's WWW page: <http://www.intel.com/design/ficomp>.

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, or life sustaining applications.

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

The 28F008C3, 28F016C3, 28F032C3, 28F800C3, 28F160C3, 28F320C3 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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 ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 5937
Denver, CO 80217-9808
or call 1-800-548-4725
or visit Intel's website at <http://www.intel.com>

CONTENTS

| | PAGE | | PAGE |
|---|-----------|---|-----------|
| 1.0 INTRODUCTION | 5 | 3.4 128-Bit Protection Register..... | 21 |
| 1.1 3 Volt Advanced+ Boot Block Flash Memory Enhancements | 5 | 3.4.1 Reading the Protection Register | 21 |
| 1.2 Product Overview | 6 | 3.4.2 Programming the Protection Register .. | 21 |
| 2.0 PRODUCT DESCRIPTION..... | 6 | 3.4.3 Locking the Protection Register | 22 |
| 2.1 Package Pinouts | 6 | 3.5 V _{PP} Program and Erase Voltages..... | 22 |
| 2.2 Block Organization | 10 | 3.5.1 Easy-12 V Operation for Fast Manufacturing Programming..... | 22 |
| 2.2.1 Parameter Blocks | 10 | 3.5.2 V _{PP} ≤ V _{PPLK} for Complete Protection ... | 22 |
| 2.2.2 Main Blocks | 10 | 3.5.3 V _{PP} Usage | 22 |
| 3.0 PRINCIPLES OF OPERATION | 11 | 3.6 Power Consumption | 23 |
| 3.1 Bus Operation | 11 | 3.6.1 Active Power (Program/Erase/Read) ... | 23 |
| 3.1.1 Read..... | 11 | 3.6.2 Automatic Power Savings (APS) | 23 |
| 3.1.2 Output Disable..... | 11 | 3.6.3 Standby Power | 23 |
| 3.1.3 Standby | 11 | 3.6.4 Deep Power-Down Mode..... | 24 |
| 3.1.4 Reset..... | 12 | 3.7 Power-Up/Down Operation..... | 24 |
| 3.1.5 Write..... | 12 | 3.7.1 RP# Connected to System Reset | 24 |
| 3.2 Modes of Operation..... | 12 | 3.7.2 V _{CC} , V _{PP} and RP# Transitions | 24 |
| 3.2.1 Read Array | 12 | 3.8 Power Supply Decoupling | 24 |
| 3.2.2 Read Configuration..... | 13 | 4.0 ABSOLUTE MAXIMUM RATINGS | 25 |
| 3.2.3 Read Status Register | 13 | 4.2 Operating Conditions..... | 25 |
| 3.2.3.1 Clearing the Status Register | 13 | 4.3 Capacitance | 26 |
| 3.2.4 Read Query..... | 13 | 4.4 DC Characteristics | 26 |
| 3.2.5 Program Mode..... | 14 | 4.5 AC Characteristics—Read Operations—Extended Temperature..... | 30 |
| 3.2.5.1 Suspending and Resuming Program | 14 | 4.6 AC Characteristics—Write Operations—Extended Temperature..... | 32 |
| 3.2.6 Erase Mode..... | 14 | 4.7 Erase and Program Timings | 33 |
| 3.2.6.1 Suspending and Resuming Erase..... | 15 | 4.8 Reset Operations | 35 |
| 3.3 Flexible Block Locking..... | 19 | 5.0 ORDERING INFORMATION..... | 36 |
| 3.3.1 Locking Operation | 19 | 6.0 ADDITIONAL INFORMATION | 37 |
| 3.3.2 Locked State | 19 | APPENDIX A: WSM Current/Next States | 38 |
| 3.3.3 Unlocked State | 19 | APPENDIX B: Program/Erase Flowcharts | 40 |
| 3.3.4 Lock-Down State | 19 | APPENDIX C: Common Flash Interface Query Structure | 46 |
| 3.3.5 Reading a Block's Lock Status | 20 | | |
| 3.3.6 Locking Operations during Erase Suspend..... | 20 | | |
| 3.3.7 Status Register Error Checking | 20 | | |

PRODUCT PREVIEW



| | | | |
|--|----|---|----|
| APPENDIX D: Architecture Block Diagram | 52 | APPENDIX G: Device ID Table | 57 |
| APPENDIX E: Word-Wide Memory Map Diagrams | 53 | APPENDIX H: Protection Register Addressing | 58 |
| APPENDIX F: Byte-Wide Memory Map Diagrams | 55 | | |

REVISION HISTORY

| Date of Revision | Version | Description |
|------------------|---------|------------------|
| 05/12/98 | -001 | Original version |



1.0 INTRODUCTION

This document contains the specifications for the 3 Volt Advanced+ Boot Block flash memory family. These flash memories add features which can be used to enhance the security of systems: instant block locking and a protection register.

Throughout this document, the term “2.7 V” refers to the full voltage range 2.7 V–3.6 V (except where noted otherwise) and “V_{PP} = 12 V” refers to 12 V ±5%. Sections 1 and 2 provide an overview of the flash memory family including applications, pinouts, pin descriptions and memory organization. Section 3 describes the operation of these products. Finally, Section 4 contains the operating specifications.

1.1 3 Volt Advanced+ Boot Block Flash Memory Enhancements

The 3 Volt Advanced+ Boot Block flash memory features:

- Zero-latency, flexible block locking
- 128-bit Protection Register
- Simple system implementation for 12 V production programming with 2.7 V in-field programming
- Ultra-low power operation at 2.7 V
- Minimum 100,000 block erase cycles
- Common Flash Interface for software query of device specs and features

Table 1. 3 Volt Advanced+ Boot Block Feature Summary

| Feature | 8 M ⁽²⁾ 16 M 32 M ⁽¹⁾ | 8 M ⁽²⁾ 16 M 32 M | Reference |
|------------------------------|--|--|---------------------------------|
| | V _{CC} Operating Voltage | 2.7 V – 3.6 V | |
| V _{PP} Voltage | Provides complete write protection with optional 12V Fast Programming | | Table 8 |
| V _{CCQ} I/O Voltage | 2.7 V– 3.6 V | | Note 3 |
| Bus Width | 8-bit | 16-bit | Table 2 |
| Speed (ns) | 90, 110 @ 2.7 V and 80, 100 @ 3.0 V | | Table 11 |
| Blocking (top or bottom) | 8 x 8-Kbyte parameter 4-Mb: 7 x 64-Kbyte main 8-Mb: 15 x 64-Kbyte main 16-Mb: 31 x 64-Kbyte main 32-Mb: 63 x 64-Kbyte main | 8 x 4-Kword parameter 4-Mb: 7 x 32-Kword main 8-Mb: 15 x 32-Kword main 16-Mb: 31 x 32-Kword main 32-Mb: 63 x 32-Kword main | Section 2.2 Appendix E and F |
| Operating Temperature | Extended: –40 °C to +85 °C | | Table 8 |
| Program/Erase Cycling | 100,000 cycles | | Table 8 |
| Packages | 40-Lead TSOP ⁽¹⁾ 48-Ball μBGA* CSP ⁽²⁾ | 48-Lead TSOP 48-Ball μBGA* CSP ⁽²⁾ | Figures 1, 2, 3, and 4 |
| Block Locking | Flexible locking of any block with zero latency | | Section 3.3 |
| Protection Register | 64-bit unique device number, 64-bit user programmable | | Section 3.4 |

NOTES:

1. 32-Mbit density not available in 40-lead TSOP.
2. 8-Mbit density not available in μBGA* CSP.
3. V_{CCQ} operation at 1.65 V — 2.5 V available upon request.

1.2 Product Overview

Intel provides secure low voltage memory solutions with the Advanced Boot Block family of products. A new block locking feature allows instant locking/unlocking of any block with zero-latency. A 128-bit protection register allows unique flash device identification.

Discrete supply pins provide single voltage read, program, and erase capability at 2.7 V while also allowing 12 V V_{PP} for faster production programming. Easy-12 V, a new feature designed to reduce external logic, simplifies board designs when combining 12 V production programming with 2.7 V in-field programming.

The 3 Volt Advanced+ Boot Block flash memory products are available in either x8 or x16 packages in the following densities: (see Section 6, *Ordering Information*)

- 8-Mbit (8,388,608 bit) flash memories organized as either 512 Kwords of 16 bits each or 1024 Kbytes or 8 bits each.
- 16-Mbit (16,777,216 bit) flash memories organized as either 1024 Kwords of 16 bits each or 2048 Kbytes of 8 bits each.
- 32-Mbit (33,554,432 bit) flash memories organized as either 2048 Kwords of 16 bits each or 4096 Kbytes of 8 bits each.

Eight 8-KB parameter blocks are located at either the top (denoted by -T suffix) or the bottom (-B suffix) of the address map in order to accommodate different microprocessor protocols for kernel code location. The remaining memory is grouped into 64-Kbyte main blocks.

All blocks can be locked or unlocked instantly to provide complete protection for code or data. (see Section 3.3 for details).

The Command User Interface (CUI) serves as the interface between the microprocessor or microcontroller and the internal operation of the flash memory. The internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for program and erase operations, including verification, thereby unburdening the microprocessor or microcontroller.

The status register indicates the status of the WSM by signifying block erase or word program completion and status.

Program and erase automation allows program and erase operations to be executed using an industry-standard two-write command sequence to the CUI. Program operations are performed in word or byte increments. Erase operations erase all locations within a block simultaneously. Both program and erase operations can be suspended by the system software in order to read from any other block. In addition, data can be programmed to another block during an erase suspend.

The 3 Volt Advanced+ Boot Block flash memories offer two low power savings features: Automatic Power Savings (APS) and standby mode. The device automatically enters APS mode following the completion of a read cycle. Standby mode is initiated when the system deselects the device by driving CE# inactive. Combined, these two power savings features significantly reduce power consumption.

The device can be reset by lowering RP# to GND. This provides CPU-memory reset synchronization and additional protection against bus noise that may occur during system reset and power-up/down sequences (see Section 3.5 and 3.6).

Refer to the *DC Characteristics* Section 4.4 for complete current and voltage specifications. Refer to the *AC Characteristics* Sections 4.5 and 4.6, for read and write performance specifications. Program and erase times and shown in Section 4.7.

2.0 PRODUCT DESCRIPTION

This section provides device pin descriptions and package pinouts for the 3 Volt Advanced+ Boot Block flash memory family, which is available in 40-Lead TSOP (x8, Figure 1), 48-lead TSOP (x16, Figure 2) and 48-ball μ BGA packages (Figures 3 and 4).

2.1 Package Pinouts

In each diagram, upgrade pins from one density to the next are circled.

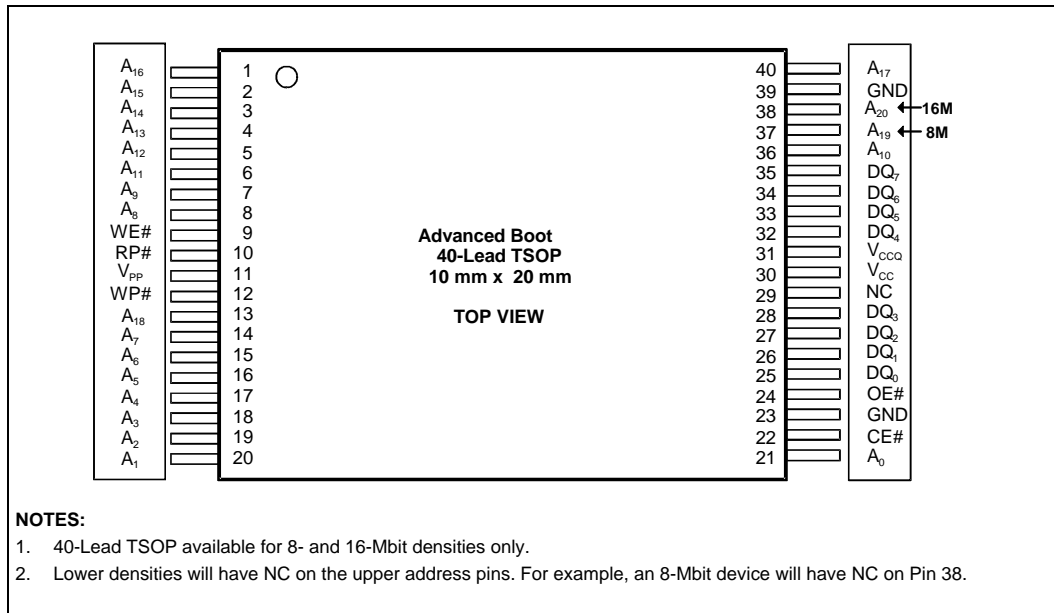


Figure 1. 40-Lead TSOP Package for x8 Configurations

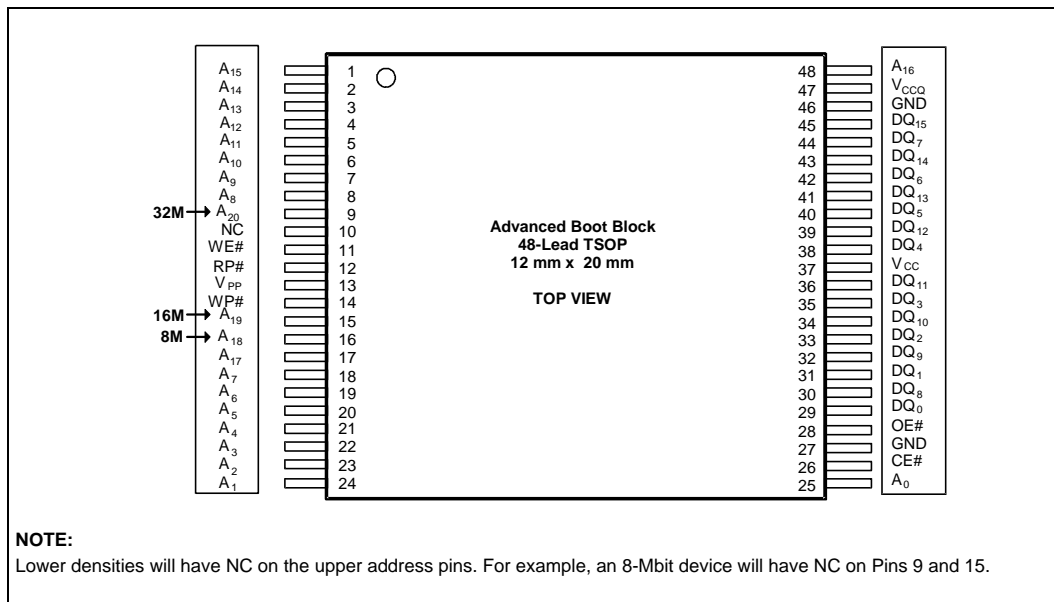


Figure 2. 48-Lead TSOP Package for x16 Configurations

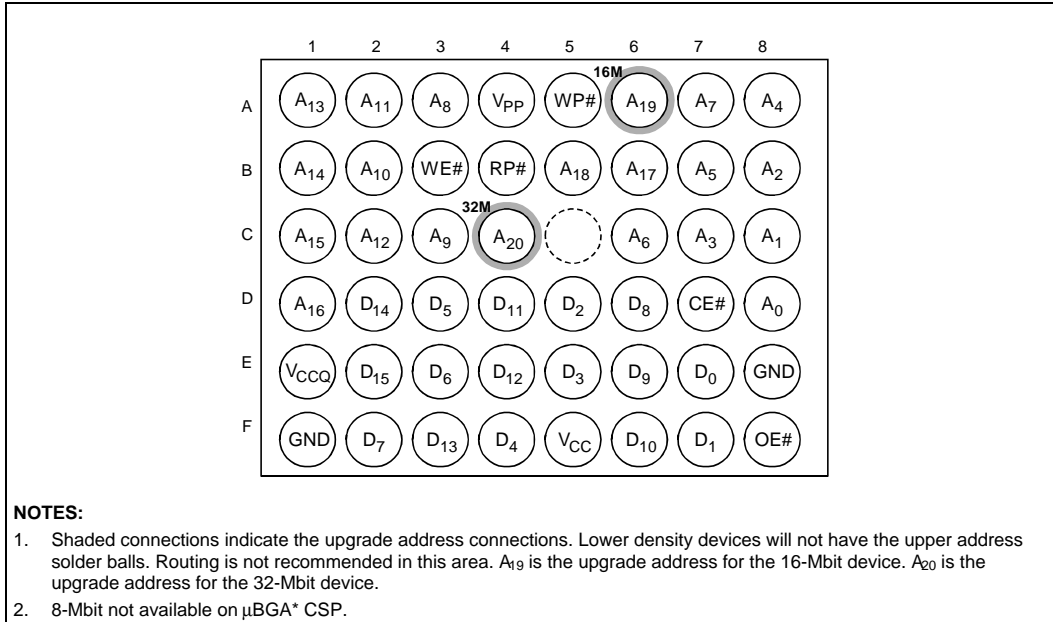


Figure 3. x16 48-Ball μBGA* Chip Size Package (Top View, Ball Down)

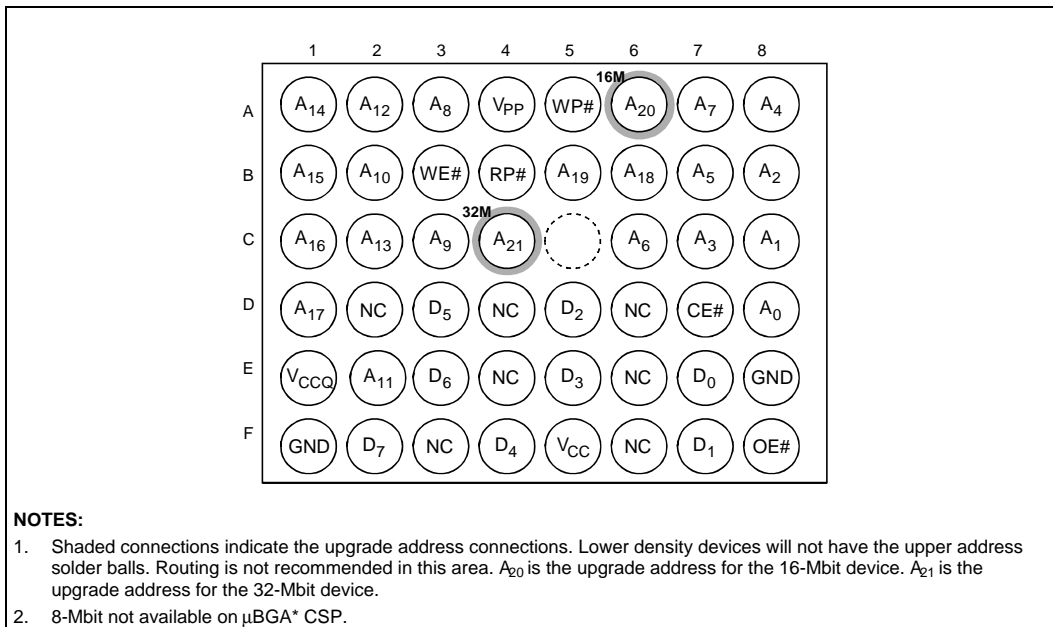


Figure 4. x8 48-Ball μBGA* Chip Size Package (Top View, Ball Down)

Table 2. 3 Volt Advanced+ Boot Block Pin Descriptions

| Symbol | Type | Name and Function |
|-----------------------------------|--------------|---|
| A ₀ –A ₂₁ | INPUT | ADDRESS INPUTS for memory addresses. Addresses are internally latched during a program or erase cycle. 8-Mbit x 8 A[0-19], 16-Mbit x 8 A[0-20], 32-Mbit x 8 A[0-21] 8-Mbit x 16 A[0-18], 16-Mbit x 16 A[0-19], 32-Mbit x 16 A[0-20] |
| DQ ₀ –DQ ₇ | INPUT/OUTPUT | DATA INPUTS/OUTPUTS: Inputs array data on the second CE# and WE# cycle during a Program command. Inputs commands to the Command User Interface when CE# and WE# are active. Data is internally latched. Outputs array, configuration and status register data. The data pins float to tri-state when the chip is de-selected or the outputs are disabled. |
| DQ ₈ –DQ ₁₅ | INPUT/OUTPUT | DATA INPUTS/OUTPUTS: Inputs array data on the second CE# and WE# cycle during a Program command. Data is internally latched. Outputs array and configuration data. The data pins float to tri-state when the chip is de-selected. Not included on x8 products. |
| CE# | INPUT | CHIP ENABLE: 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 | OUTPUT ENABLE: Enables the device's outputs through the data buffers during a read operation. OE# is active low. |
| WE# | INPUT | WRITE ENABLE: Controls writes to the Command Register and memory array. WE# is active low. Addresses and data are latched on the rising edge of the second WE# pulse. |
| RP# | INPUT | RESET/DEEP POWER-DOWN: Uses two voltage levels (V _{IL} , V _{IH}) to control reset/deep power-down mode. 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}). 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. |
| WP# | INPUT | WRITE PROTECT: Controls the lock-down function of the flexible Locking feature When WP# is a logic low, the lock-down mechanism is enabled and blocks marked lock-down cannot be unlocked through software. 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 that state. See Section 3.3 for details on block locking. |
| V _{CC} | SUPPLY | DEVICE POWER SUPPLY: [2.7 V–3.6 V] Supplies power for device operations. |

Table 2. 3 Volt Advanced+ Boot Block Pin Descriptions (Continued)

| Symbol | Type | Name and Function |
|------------------|------------------|--|
| V _{CCQ} | INPUT | I/O POWER SUPPLY: Supplies power for input/output buffers. [2.7 V–3.6 V] This input should be tied directly to V _{CC} . [1.65 V– 2.5 V] Lower I/O power supply voltage available upon request. Contact your Intel representative for more information. |
| V _{PP} | INPUT/ SUPPLY | PROGRAM/ERASE POWER SUPPLY: [1.65 V–3.6 V or 11.4 V–12.6 V] Operates as an input at logic levels to control complete device protection. Supplies power for accelerated program and erase operations in 12 V ± 5% range. This pin cannot be left floating. Lower V_{PP} ≤ V_{PPLK}, to protect all contents against Program and Erase commands. Set V_{PP} = V_{CC} for in-system read, program and erase operations. In this configuration, V _{PP} can drop as low as 1.65 V to allow for resistor or diode drop from the system supply. Note that if V _{PP} is driven by a logic signal, V _{IH} = 1.65. That is, V _{PP} must remain above 1.65V to perform in-system flash modifications. Raise V_{PP} to 12 V ± 5% for faster program and erase in a production environment. Applying 12 V ± 5% to V _{PP} can only be done for a maximum of 1000 cycles on the main blocks and 2500 cycles on the parameter blocks. V _{PP} may be connected to 12 V for a total of 80 hours maximum. See Section 3.4 for details on V _{PP} voltage configurations. |
| GND | SUPPLY | GROUND: For all internal circuitry. All ground inputs must be connected. |
| NC | | NO CONNECT: Pin may be driven or left floating. |

2.2 Block Organization

The 3 Volt Advanced+ Boot Block is an asymmetrically-blocked architecture that enables system integration of code and data within a single flash device. Each block can be erased independently of the others up to 100,000 times. For the address locations of each block, see the memory maps in Appendix E and F.

2.2.1 PARAMETER BLOCKS

The 3 Volt Advanced+ Boot Block flash memory architecture includes parameter blocks to facilitate storage of frequently updated small parameters (i.e., data that would normally be stored in an EEPROM). Each device contains eight parameter blocks of 8-Kbytes/4-Kwords (8,192 bytes/4,096 words).

2.2.2 MAIN BLOCKS

After the parameter blocks, the remainder of the array is divided into equal size (64-Kword/32-Kword; 65,536 bytes/32,768 words) main blocks for data or code storage. Each 8-Mbit, 16-Mbit, or 32-Mbit device contains 15, 31, or 63 main blocks, respectively.

3.0 PRINCIPLES OF OPERATION

The 3 Volt Advanced+ Boot Block flash memory family utilizes 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 status. The CUI handles the WE# interface to the data and address latches, as well as system status requests during WSM operation.

3.1 Bus Operation

The 3 Volt Advanced+ Boot Block flash memory devices read, program and erase in-system via the local CPU or microcontroller. All bus cycles to or from the flash memory conform to standard microcontroller bus cycles. Four control pins dictate the data flow in and out of the flash component: CE#, OE#, WE# and RP#. These bus operations are summarized in Table 3.

3.1.1 READ

The flash memory has four read modes available: read array, read configuration, read status and read query. These modes are accessible independent of

the V_{PP} voltage. The appropriate read mode command must be issued to the CUI to enter the corresponding mode. Upon initial device power-up or after exit from reset, the device automatically defaults to read array mode.

CE# and OE# must be driven active to obtain data at the outputs. CE# is the device selection control; when active it enables the flash memory device. OE# is the data output control and it drives the selected memory data onto the I/O bus. For all read modes, WE# and RP# must be at V_{IH} . Figure 9 illustrates a read cycle.

3.1.2 OUTPUT DISABLE

With OE# at a logic-high level (V_{IH}), the device outputs are disabled. Output pins are placed in a high-impedance state.

3.1.3 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 program or erase operation, the device continues to consume active power until the program or erase operation is complete.

Table 3. Bus Operations⁽¹⁾

| Mode | Note | RP# | CE# | OE# | WE# | DQ ₀₋₇ | DQ ₈₋₁₅ |
|---|-------|----------|----------|----------|----------|-------------------|--------------------|
| Read (Array, Status, Configuration, or Query) | 2-4 | V_{IH} | V_{IL} | V_{IL} | V_{IH} | D_{OUT} | D_{OUT} |
| Output Disable | 2 | V_{IH} | V_{IL} | V_{IH} | V_{IH} | High Z | High Z |
| Standby | 2 | V_{IH} | V_{IH} | X | X | High Z | High Z |
| Reset | 2,7 | V_{IL} | X | X | X | High Z | High Z |
| Write | 2,5-7 | V_{IH} | V_{IL} | V_{IH} | V_{IL} | D_{IN} | D_{IN} |

NOTES:

- 8-bit devices use only DQ[0:7], 16-bit devices use DQ[0:15]
- X must be V_{IL} , V_{IH} for control pins and addresses.
- See *DC Characteristics* for V_{PPLK} , V_{PP1} , V_{PP2} , V_{PP3} , voltages.
- Manufacturer and device codes may also be accessed in read configuration mode ($A_1-A_{20} = 0$). See Table 4.
- Refer to Table 5 for valid D_{IN} during a write operation.
- To program or erase the lockable blocks, hold WP# at V_{IH} .
- RP# must be at $GND \pm 0.2 V$ to meet the maximum deep power-down current specified.

3.1.4 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 can be initiated. After this wake-up interval, normal operation is restored. The CUI resets to read array mode, and the status register is set to 80H. This case is shown in Figure 11A.

If RP# is taken low for time t_{PLPH} during a program or erase operation, the operation will be aborted and 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: When RP# goes low, the device shuts down the operation in progress, a process which takes time t_{PLRH} to complete. After this time t_{PLRH} , the part will either reset to read array mode (if RP# has gone high during t_{PLRH} , Figure 11B) or enter reset mode (if RP# is still logic low after t_{PLRH} , Figure 11C). In both cases, after returning from an aborted operation, the relevant time t_{PHQV} or t_{PHWL}/t_{PHEL} must be waited 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 system reset. When the system comes out of reset, processor expects to read 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's 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.

3.1.5 WRITE

A write takes place when both CE# and WE# are low and OE# is high. Commands are written to the Command User Interface (CUI) using standard microprocessor write timings to control flash operations. The CUI does not occupy an

addressable memory location. The address and data buses are latched on the rising edge of the second WE# or CE# pulse, whichever occurs first. Figure 10 illustrates a program and erase operation. The available commands are shown in Table 6, and Appendix A provides detailed information on moving between the different modes of operation using CUI commands.

There are two commands that modify array data: Program (40H) and Erase (20H). Writing either of these commands to the internal Command User Interface (CUI) initiates a sequence of internally-timed functions that culminate in the completion of the requested task (unless that operation is aborted by either RP# being driven to V_{IL} for t_{PLRH} or an appropriate suspend command).

3.2 Modes of Operation

The flash memory has four read modes and two write modes. The read modes are read array, read configuration, read status, and read query. The write modes are program and block erase. Three additional modes (erase suspend to program, erase suspend to read and program suspend to read) are available only during suspended operations. These modes are reached using the commands summarized in Tables 5 and 6. A comprehensive chart showing the state transitions is in Appendix A.

3.2.1 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 (FFH) must be written to the CUI before array reads can take place.

3.2.2 READ CONFIGURATION

The Read Configuration mode outputs the manufacturer/device identifier. The device is switched to this mode by writing the Read Configuration command (90H). Once in this mode, read cycles from addresses shown in Table 4 retrieve the specified information. To return to read array mode, write the Read Array command (FFH).

The Read Configuration 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 writing the Read Configuration command (90H). Once in this mode, read cycles from addresses shown in Table 4 retrieve the specified information. To return to read array mode, write the Read Array command (FFH).

Table 4. Read Configuration Table

| Item | Address | Data |
|---------------------------------------|----------------------|---------------------|
| Manufacturer Code (x16) | 00000 | 0089 |
| Manufacturer Code (x8) | 00000 | 89 |
| Device ID (See Appendix G) | 00001 | ID |
| Block Lock Configuration ² | XX002 ⁽¹⁾ | LOCK |
| • Block Is Unlocked | | DQ ₀ = 0 |
| • Block Is Locked | | DQ ₀ = 1 |
| • Block Is Locked-Down | | DQ ₁ = 1 |
| Protection Register Lock ³ | 80 | PR-LK |
| Protection Register (x16) | 81-88 | PR |
| Protection Register (x8) | (App. H) | PR |

NOTES:

1. "XX" specifies the block address of lock configuration being read.
2. See Section 3.3.4 for valid lock status outputs.
3. See Section 3.4 for protection register information.
4. Other locations within the configuration address space are reserved by Intel for future use.

3.2.3 READ STATUS REGISTER

The status register indicates the status of device operations, and the success/failure of that operation. The Read Status Register (70H)

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 (FFH) command.

The status register bits are output on DQ₀-DQ₇. The upper byte, DQ₈-DQ₁₅, outputs 00H during a Read Status Register command.

The contents of the status register are latched on the falling edge of OE# or CE#, whichever occurs last. This prevents possible bus errors which 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 desired operation (see Table 7).

3.2.3.1 Clearing the Status Register

The WSM sets status bits 1 through 7 to "1," and clears bits 2, 6 and 7 to "0," but cannot clear status bits 1 or 3 through 5 to "0." Because bits 1, 3, 4 and 5 indicate various error conditions, these bits can only be cleared through the use of the Clear Status Register (50H) 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. Note that the Read Array command must be issued before data can be read from the memory array. Resetting the device also clears the status register.

3.2.4 READ QUERY

The Read Query mode outputs Common Flash Interface (CFI) data when the device is read. This can be accessed by writing the Read Query Command (98H). 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 retrieve the specified information. To return to read array mode, write the Read Array command (FFH).

3.2.5 PROGRAM MODE

Programming is executed using a two-write sequence. The Program Setup command (40H) is written to the CUI followed by a second write which specifies the address and data to be programmed. The WSM will execute a sequence of internally timed events to program desired 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 the user attempts 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 should be checked. If the programming operation was unsuccessful, bit SR.4 of the status register 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.

3.2.5.1 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, writing 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 written. Polling status register bits SR.7 and SR.2 will determine when the program operation has been suspended (both will be set to "1"). t_{WHRH1}/t_{EHRH1} specify the program suspend latency.

A Read Array command can now be written 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 Configuration, Read Query, and Program Resume. After the Program Resume command is written to the flash memory, the WSM will continue with the programming process and status register bits SR.2 and SR.7 will automatically be cleared. The device automatically outputs status register data when read (see Figure 13 in Appendix B, *Program Suspend/Resume Flowchart*) after the Program Resume command is written. 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} .

3.2.6 ERASE MODE

To erase a block, write 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} was 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 was not within acceptable limits.

After an erase operation, clear the status register (50H) before attempting the next operation. Any CUI instruction can follow after erasure is completed; however, to prevent inadvertent status register reads, it is advisable to place the flash in read array mode after the erase is complete.

3.2.6.1 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 in order to read data from or program data to another block in memory. Once the erase sequence is started, writing the Erase Suspend command to the CUI suspends the erase sequence at a predetermined point in the erase algorithm. The status register will indicate if/when the erase operation has been suspended. Erase suspend latency is specified by t_{WHRH2}/t_{EHRH2} .

A Read Array/Program command can now be written 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 Configuration, Read Query, Program Setup, Program Resume, Erase Resume, Lock Block, Unlock Block and Lock-Down Block. During erase suspend mode, the chip can be placed in a pseudo-standby mode by taking CE# to V_{IH} . This reduces active current consumption.

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

Table 5. Command Bus Definitions

| Command | Notes | First Bus Cycle | | | Second Bus Cycle | | |
|-----------------------|-------|-----------------|------|---------|------------------|------|------|
| | | Oper | Addr | Data | Oper | Addr | Data |
| Read Array | 4 | Write | X | FFH | | | |
| Read Configuration | 2, 4 | Write | X | 90H | Read | IA | ID |
| Read Query | 2, 4 | Write | X | 98H | Read | QA | QD |
| Read Status Register | 4 | Write | X | 70H | Read | X | SRD |
| Clear Status Register | 4 | Write | X | 50H | | | |
| Program | 3,4 | Write | X | 40H/10H | Write | PA | PD |
| Block Erase/Confirm | 4 | Write | X | 20H | Write | BA | D0H |
| Program/Erase Suspend | 4 | Write | X | B0H | | | |
| Program/Erase Resume | 4 | Write | X | D0H | | | |
| Lock Block | 4 | Write | X | 60H | Write | BA | 01H |
| Unlock Block | 4 | Write | X | 60H | Write | BA | D0H |
| Lock-Down Block | 4 | Write | X | 60H | Write | BA | 2FH |
| Protection Program | 4 | Write | X | C0H | 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:

1. Bus operations are defined in Table 3.
2. Following the Read Configuration or Read Query commands, read operations output device configuration or CFI query information, respectively. See Section 3.2.2 and 3.2.4.
3. Either 40H or 10H command is valid, but the Intel standard is 40H.
4. When writing commands, the upper data bus [DQ₈-DQ₁₅] should be either V_{IL} or V_{IH} , to minimize current draw.

Table 6. Command Codes and Descriptions

| Code | Device Mode | Description |
|------|-----------------------|--|
| FF | Read Array | Places device in read array mode, such that array data will be output 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 Section 3.2.5. |
| 20 | Erase Set-Up | 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 of the status register to a "1," (b) place the device into the read status register mode, and (c) wait for another command. See Section 3.2.6. |
| D0 | Erase Confirm | 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. |
| | Program/Erase Resume | If a program or erase operation was previously suspended, this command will resume that operation. |
| | Unlock Block | If the previous command was Configuration 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. (Sect. 3.3) |
| B0 | Program 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 _{IL} . See Sections 3.2.5.1 and 3.2.6.1. |
| | Erase Suspend | |
| 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 Section 3.2.3. |
| 50 | Clear Status Register | The WSM can set the Block Lock Status (SR.1), V _{PP} 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 Configuration | Puts the device into the Read Configuration mode, so that reading the device will output the manufacturer/device codes or block lock status. Section 3.2.2. |
| 60 | Configuration Set-Up | Prepares the CUI for changes to the device configuration, such as 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 Section 3.3. |
| 01 | Lock-Block | If the previous command was Configuration Set-Up, the CUI will latch the address and lock the block indicated on the address pins. (Section 3.3) |

Table 6. Command Codes and Descriptions (Continued)

| Code | Device Mode | Description |
|------|--------------------------|--|
| 2F | Lock-Down | If the previous command was a Configuration Set-Up command, the CUI will latch the address and lock-down the block indicated on the address pins. (Section 3.3) |
| 98 | Read Query | Puts the device into the Read Query mode, so that reading the device will output Common Flash Interface information. See Section 3.2.4 and Appendix C. |
| C0 | Protection Program Setup | 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 Section 3.4. |
| 10 | Alt. Prog Set-Up | Operates the same as Program Set-up command. (See 40H/Program Set-Up) |
| 00 | Invalid/Reserved | Unassigned commands that should not be used. Intel reserves the right to redefine these codes for future functions. |

NOTE:

See Appendix A for mode transition information.

Table 7. Status Register Bit Definition

| WSMS | ESS | ES | PS | VPPS | PSS | BLS | R |
|---|-----|----|----|---|-----|-----|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NOTES: | | | | | | | |
| SR.7 WRITE STATE MACHINE STATUS 1 = Ready (WSMS) 0 = Busy | | | | Check Write State Machine bit first to determine Word Program or Block Erase completion, before checking Program or Erase Status bits. | | | |
| SR.6 = ERASE-SUSPEND STATUS (ESS) 1 = Erase Suspended 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) 1 = Error In Block Erase 0 = Successful Block Erase | | | | When this bit is set to "1," WSM has applied the max. number of erase pulses to the block and is still unable to verify successful block erasure. | | | |
| SR.4 = PROGRAM STATUS (PS) 1 = Error in Programming 0 = Successful Programming | | | | When this bit is set to "1," WSM has attempted but failed to program a word/byte. | | | |
| SR.3 = V _{PP} STATUS (VPPS) 1 = V _{PP} Low Detect, Operation Abort 0 = V _{PP} OK | | | | The V _{PP} status bit does not provide continuous indication of V _{PP} level. The WSM interrogates V _{PP} level only after the Program or Erase command sequences have been entered, and informs the system if V _{PP} has not been switched on. The V _{PP} is also checked before the operation is verified by the WSM. The V _{PP} status bit is not guaranteed to report accurate feedback between V _{PP} and V _{PP1Min} . | | | |
| SR.2 = PROGRAM SUSPEND STATUS (PSS) 1 = Program Suspended 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 1 = Prog/Erase attempted on a locked block; Operation aborted. 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. | | | |

3.3 Flexible Block Locking

The Intel® 3 Volt Advanced+ Boot Block products offer 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 [XYZ]” will be used to specify locking states; e.g., “state [001],” where X = value of WP#, Y = bit DQ₁ of the Block Lock status register, and Z = bit DQ₀ of the Block Lock status register. Table 9 defines all of these possible locking states.

3.3.1 LOCKING OPERATION

The following concisely summarizes the locking functionality.

- All blocks power-up locked, then can be unlocked or locked with the Unlock and Lock commands.
- The Lock-Down command locks a block and prevents it from being unlocked when WP# = 0.
 - When WP# = 1, Lock-Down is overridden and commands can unlock/lock locked-down blocks.
 - When WP# returns to 0, locked-down blocks return to Lock-Down.
 - Lock-Down is cleared only when the device is reset or powered-down.

The locking status of each block can set to Locked, Unlocked, and Lock-Down, each of which will be described in the following sections. A comprehensive state table for the locking functions is shown in Table 9, and a flowchart for locking operations is shown in Figure 16.

3.3.2 LOCKED STATE

The default status 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 of the status register. The status 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, 60H followed by 01H.

3.3.3 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, 60H followed by D0H.

3.3.4 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, 60H followed by 2FH. Locked-Down blocks revert to the Locked state when the device is reset or powered down.

The Lock-Down function is dependent 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 desired 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.

3.3.5 READING A BLOCK'S LOCK STATUS

The lock status of every block can be read in the Configuration Read mode of the device. To enter this mode, write 90H to the device. Subsequent reads at Block Address + 00002 will output the lock status of that block. The lock status is represented by the lowest two output pins, 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.

Table 8. Block Lock Status

| Item | Address | Data |
|--------------------------|---------|---------------------|
| Block Lock Configuration | XX002 | LOCK |
| • Block Is Unlocked | | DQ ₀ = 0 |
| • Block Is Locked | | DQ ₀ = 1 |
| • Block Is Locked-Down | | DQ ₁ = 1 |

3.3.6 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 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 write the erase suspend command (B0H), then check the status register until it indicates that the erase operation has been suspended. Next write the desired lock command sequence to a block and

the lock status will be changed. After completing any desired lock, read, or program operations, resume the erase operation with the Erase Resume command (D0H).

If a block is locked or locked-down during a suspended erase of the same block, the locking status bits will 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 for detailed information on which commands are valid during erase suspend.

3.3.7 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, e.g., 60H followed by 01H to lock a block, following the Configuration Setup command (60H) 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 via the status register because of the previous locking command error.

A similar situation happens if an error occurs during a program operation error nested within an erase suspend.

Table 9. Block Locking State Transitions

| Current State | | | | Erase/Prog Allowed? | Lock Command Input Result [Next State] | | |
|---------------|-----------------|-----------------|--------------------|------------------------|--|---------------|---------------|
| WP# | DQ ₁ | DQ ₀ | Name | | Lock | Unlock | Lock-Down |
| 0 | 0 | 0 | "Unlocked" | Yes | Goes To [001] | No Change | Goes To [011] |
| 0 | 0 | 1 | "Locked" (Default) | No | No Change | Goes To [000] | Goes To [011] |
| 0 | 1 | 1 | "Locked-Down" | No | No Change | No Change | No Change |
| 1 | 0 | 0 | "Unlocked" | Yes | Goes To [101] | No Change | Goes To [111] |
| 1 | 0 | 1 | "Locked" | No | No Change | Goes To [100] | Goes To [111] |
| 1 | 1 | 0 | Lock-Down Disabled | Yes | Goes To [111] | No Change | Goes To [111] |
| 1 | 1 | 1 | Lock-Down Disabled | No | No Change | Goes To [110] | No Change |

NOTES:

1. In this table, the notation [XYZ] denotes the locking state of a block, where X = WP#, Y = DQ₁, and Z = DQ₀. The current locking state of a block is defined by the state of WP# and the two bits of the block lock status (DQ₀, DQ₁). DQ₀ indicates if a block is locked (1) or unlocked (0). DQ₁ indicates if a block has been locked-down (1) or not (0).
2. At power-up or device reset, all blocks default to Locked state [001] (if WP# = 0). Holding WP# = 0 is the recommended default.
3. The "Erase/Program Allowed?" column shows whether erase and program operations are enabled (Yes) or disabled (No) in that block's current locking state.
4. The "Lock Command Input Result [Next State]" column shows the result of writing the three locking commands (Lock, Unlock, Lock-Down) in the current locking state. For example, "Goes To [001]" would mean that writing the command to a block in the current locking state would change it to [001].

3.4 128-Bit Protection Register

The Advanced+ Boot Block 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 "mate" the flash component with other system components such as the CPU or ASIC, preventing device substitution. Additional application information can be found in Intel application note *AP-657 Designing with the Advanced+ Boot Block Flash Memory Architecture*.

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 desired. Once the customer segment is programmed, it can be locked to prevent reprogramming.

3.4.1 READING THE PROTECTION REGISTER

The protection register is read in the configuration read mode. The device is switched to this mode by writing the Read Configuration command (90H). Once in this mode, read cycles from addresses shown in Appendix H retrieve the specified information. To return to read array mode, write the Read Array command (FFH).

3.4.2 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 for word-wide parts and eight bits at a time for byte-wide parts. First write the Protection Program Setup command, C0H. The next write to the device will latch in address and data and program the specified location. The allowable addresses are shown in Appendix H. See Figure 17 for the *Protection Register Programming Flowchart*.

3 VOLT ADVANCED+ BOOT BLOCK



Any attempt to address Protection Program commands outside the defined protection register address space will result in a Status Register error (Program Error bit SR.4 will be set to 1). Attempting to program or 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).

3.4.3 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. 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 "FFFD" 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.

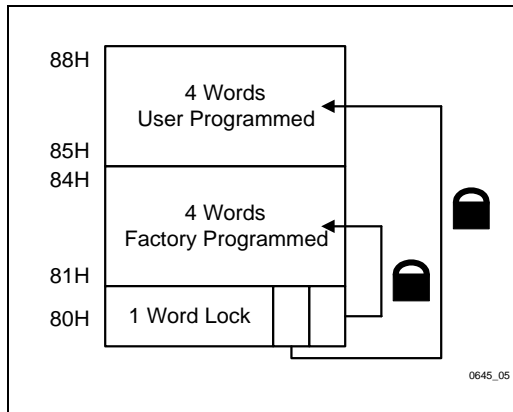


Figure 5. Protection Register Memory Map

3.5 V_{PP} Program and Erase Voltages

Intel's 3 Volt Advanced+ Boot Block products provide in-system writes plus a V_{PP} pin for 12 V production programming and complete write protection.

3.5.1 EASY-12 V OPERATION FOR FAST MANUFACTURING PROGRAMMING

Intel's 3 Volt Advanced+ Boot Block products provide in-system programming and erase in the 2.7 V–3.6 V range. For fast production programming, 3 Volt Advanced+ Boot Block includes a low-cost, backward-compatible 12 V programming feature.

When V_{PP} is between 1.65 V and 3.6 V, all program and erase current is drawn through the V_{CC} pin. Note that if V_{PP} is driven by a logic signal, V_{IH} = 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 V_{PP} pin. This eliminates the need for an external switching transistor to control the voltage V_{PP}. Figure 6 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. 12 V may be applied to V_{PP} during program and erase operations for a maximum of 1000 cycles on the main blocks and 2500 cycles on the parameter blocks. V_{PP} may be connected to 12 V for a total of 80 hours maximum. Stressing the device beyond these limits may cause permanent damage.

3.5.2 V_{PP} ≤ V_{PPLK} FOR COMPLETE 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 V_{PPLK}, any program or erase operation will result in an error, prompting the corresponding status register bit (SR.3) to be set.

3.5.3 V_{PP} USAGE

The V_{PP} pin is used for two functions: Absolute data protection and fast production programming.

When V_{PP} ≤ V_{PPLK}, then all program or erase operations to the device are inhibited, providing absolute data protection.

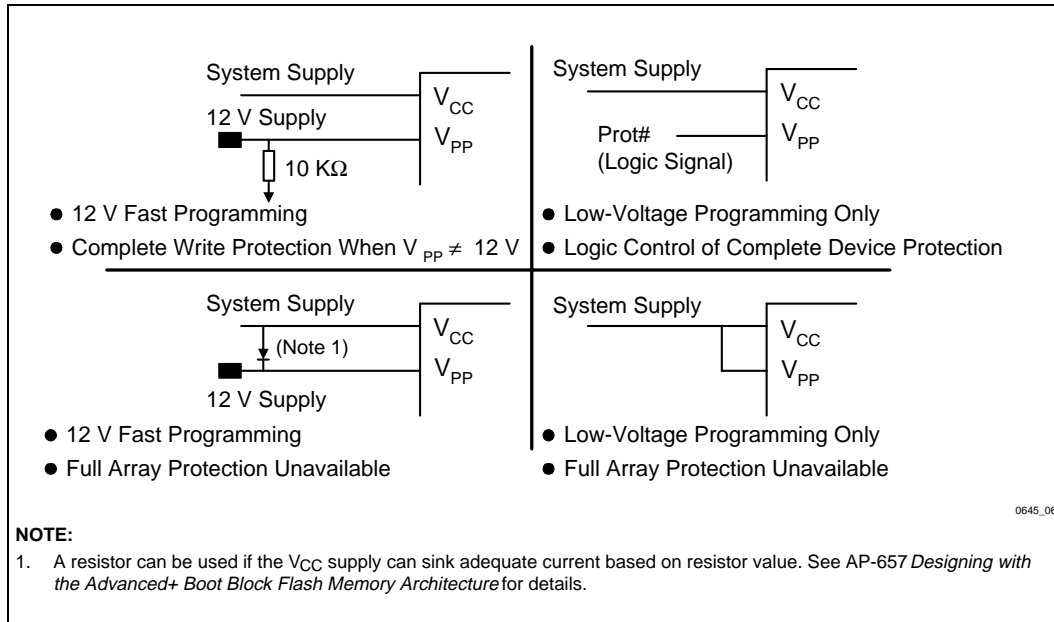


Figure 6. Example Power Supply Configurations

When V_{PP} is raised to 12 V, such as in a manufacturing situations, the device directly applies the high voltage to achieve faster program and erase.

Designing for in-system writes to the flash memory requires special consideration of power supply traces by the printed circuit board designer. Adequate power supply traces, and decoupling capacitors placed adjacent to the component, will decrease spikes and overshoots.

3.6 Power Consumption

Intel's 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 the $CE\#$ is deasserted, the flash enters its standby mode, where current consumption is even lower. The combination of these features can minimize memory power consumption, and therefore, overall system power consumption.

3.6.1 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.

3.6.2 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 quiescent, 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.

3.6.3 STANDBY POWER

With $CE\#$ at a logic-high level (V_{IH}) and device in read mode, the flash memory is in standby mode, which disables much of the device's circuitry and

3 VOLT ADVANCED+ BOOT BLOCK

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 will provide a more accurate measure of application-specific power and energy requirements.

3.6.4 DEEP POWER-DOWN MODE

The deep power-down mode is activated when $RP\# = V_{IL}$ ($GND \pm 0.2 V$). 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 will abort 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).

3.7 Power-Up/Down Operation

The device is protected against accidental block erasure or programming during power transitions. Power supply sequencing is not required, since the device is indifferent as to which power supply, V_{PP} or V_{CC} , powers-up first.

3.7.1 $RP\#$ CONNECTED TO SYSTEM RESET

The use of $RP\#$ during system reset is important with automated program/erase devices since the system expects to read 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} . Since 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 ($RP\#$ connected to system PowerGood) during power-up/down, invalid bus conditions during power-up can be masked, providing yet another level of memory protection.

3.7.2 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 via the Read Array command if access to the flash memory array is desired.

3.8 Power Supply Decoupling

Flash memory's power switching characteristics require careful device decoupling. System designers should consider three supply current issues:

1. Standby current levels (I_{CCS})
2. Read current levels (I_{CCR})
3. 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 μF ceramic capacitor connected between each V_{CC} and GND, and between its V_{PP} and GND. These high-frequency, inherently low-inductance capacitors should be placed as close as possible to the package leads.

4.0 ABSOLUTE MAXIMUM RATINGS*

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 +5.0 V¹
 V_{PP} Voltage (for Block
 Erase and Program)
 with Respect to GND -0.5 V to +13.5 V^{1,2,4}
 V_{CC} and V_{CCQ} Supply Voltage
 with Respect to GND -0.2 V to +5.0 V¹
 Output Short Circuit Current..... 100 mA³

NOTICE: This datasheet contains preliminary information on products in the design phase of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest datasheet before finalizing a design.

* **WARNING:** *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may effect device reliability.*

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. Output shorted for no more than one second. No more than one output shorted at a time.
4. V_{PP} voltage is normally 1.65 V–3.6 V. Connection to supply of 11.4 V–12.6 V can only be done for 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. See Section 3.5 for details.

4.2 Operating Conditions

Table 10. 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 | 2.7 | 3.6 | Volts |
| V_{CC2} | | 1 | 3.0 | 3.6 | |
| V_{CCQ1} | I/O Supply Voltage | 1 | 2.7 | 3.6 | Volts |
| V_{PP1} | Supply Voltage | 1 | 1.65 | 3.6 | Volts |
| V_{PP2} | | 1, 2 | 11.4 | 12.6 | Volts |
| Cycling | Block Erase Cycling | 2 | 100,000 | | Cycles |

NOTES:

1. V_{CC} and V_{CCQ} must share the same supply when they are in the V_{CC1} range.
2. Applying $V_{PP} = 11.4$ 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_{PP} may be connected to 12 V for a total of 80 hours maximum. See Section 3.5 for details.

4.3 Capacitance

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

| Sym | Parameter | Notes | Typ | Max | Units | Conditions |
|-----------|--------------------|-------|-----|-----|-------|------------------------|
| C_{IN} | Input Capacitance | 1 | 6 | 8 | pF | $V_{IN} = 0\text{ V}$ |
| C_{OUT} | Output Capacitance | 1 | 10 | 12 | pF | $V_{OUT} = 0\text{ V}$ |

NOTE:

1. Sampled, not 100% tested.

4.4 DC Characteristics

| Sym | Parameter | V_{CC} | 2.7 V–3.6 V | | Unit | Test Conditions |
|------------|----------------------------------|-----------|-------------|----------|---------------|--|
| | | V_{CCQ} | 2.7 V–3.6 V | | | |
| | | Note | Typ | Max | | |
| I_{LI} | Input Load Current | 1,7 | | ± 1 | μA | $V_{CC} = V_{CCMax}$ $V_{CCQ} = V_{CCQMax}$ $V_{IN} = V_{CCQ}$ or GND |
| I_{LO} | Output Leakage Current | 1,7 | 0.2 | ± 10 | μA | $V_{CC} = V_{CCMax}$ $V_{CCQ} = V_{CCQMax}$ $V_{IN} = V_{CCQ}$ or GND |
| I_{CCS} | V_{CC} Standby Current | 1 | 10 | 25 | μA | $V_{CC} = V_{CCMax}$ $CE\# = RP\# = V_{CC}$ |
| I_{CCD} | V_{CC} Deep Power-Down Current | 1,7 | 7 | 20 | μA | $V_{CC} = V_{CCMax}$ $V_{CCQ} = V_{CCQMax}$ $V_{IN} = V_{CCQ}$ or GND $RP\# = GND \pm 0.2\text{ V}$ |
| I_{CCR} | V_{CC} Read Current | 1,5,7 | 9 | 18 | mA | $V_{CC} = V_{CCMax}$ $V_{CCQ} = V_{CCQMax}$ $OE\# = V_{IH}$, $CE\# = V_{IL}$ $f = 5\text{ MHz}$, $I_{OUT} = 0\text{ mA}$ Inputs = V_{IL} or V_{IH} |
| I_{CCW} | V_{CC} Program Current | 1,4 | 18 | 55 | mA | $V_{PP} = V_{PP1}$ Program in Progress |
| | | | 8 | 15 | mA | $V_{PP} = V_{PP2} (12\text{ V})$ Program in Progress |
| I_{CCE} | V_{CC} Erase Current | 1,4 | 16 | 45 | mA | $V_{PP} = V_{PP1}$ Erase in Progress |
| | | | 8 | 15 | mA | $V_{PP} = V_{PP2} (12\text{ V})$ Erase in Progress |
| I_{CCES} | V_{CC} Erase Suspend Current | 1,2,4 | 10 | 25 | μA | $CE\# = V_{IH}$, Erase Suspend in Progress |
| I_{CCWS} | V_{CC} Program Suspend Current | 1,2,4 | 10 | 25 | μA | $CE\# = V_{IH}$, Program Suspend in Progress |

4.4 DC Characteristics, Continued

| Sym | Parameter | V _{CC} | 2.7 V–3.6 V | | Unit | Test Conditions |
|-------------------|---|------------------|-------------|-----|------|--|
| | | V _{CCQ} | 2.7 V–3.6 V | | | |
| | | Note | Typ | Max | | |
| I _{PPD} | V _{PP} Deep Power-Down Current | 1 | 0.2 | 5 | μA | RP# = GND ± 0.2 V |
| I _{PPS} | V _{PP} Standby Current | 1 | 0.2 | 5 | μA | V _{PP} ≤ V _{CC} |
| I _{PPR} | V _{PP} Read Current | 1 | 2 | ±15 | μA | V _{PP} ≤ V _{CC} |
| | | 1,4 | 50 | 200 | μA | V _{PP} ≥ V _{CC} |
| I _{PPW} | V _{PP} Program Current | 1,4 | 0.05 | 0.1 | mA | V _{PP} = V _{PP1} Program in Progress |
| | | | 8 | 22 | mA | V _{PP} = V _{PP2} (12 V) Program in Progress |
| I _{PPE} | V _{PP} Erase Current | 1,4 | 0.05 | 0.1 | mA | V _{PP} = V _{PP1} Program in Progress |
| | | | 8 | 22 | mA | V _{PP} = V _{PP2} (12 V) Program in Progress |
| I _{PPES} | V _{PP} Erase Suspend Current | 1,4 | 0.2 | 5 | μA | V _{PP} = V _{PP1} Erase Suspend in Progress |
| | | | 50 | 200 | μA | V _{PP} = V _{PP2} (12 V) Erase Suspend in Progress |
| I _{PPWS} | V _{PP} Program Suspend Current | 1,4 | 0.2 | 5 | μA | V _{PP} = V _{PP1} Program Suspend in Progress |
| | | | 50 | 200 | μA | V _{PP} = V _{PP2} (12 V) Program Suspend in Progress |

4.4 DC Characteristics, Continued

| Sym | Parameter | V _{CC} | 2.7 V–3.6 V | | Unit | Test Conditions |
|-------------------|---|------------------|--------------------------|------|------|---|
| | | V _{CCQ} | 2.7 V–3.6 V | | | |
| | | Note | Min | Max | | |
| V _{IL} | Input Low Voltage | | -0.4 | 0.4 | V | |
| V _{IH} | Input High Voltage | | V _{CCQ} - 0.4 V | | V | |
| V _{OL} | Output Low Voltage | 7 | -0.10 | 0.10 | V | V _{CC} = V _{CCMin} V _{CCQ} = V _{CCQMin} I _{OL} = 100 μA |
| V _{OH} | Output High Voltage | 7 | V _{CCQ} - 0.1 V | | V | V _{CC} = V _{CCMin} V _{CCQ} = V _{CCQMin} I _{OH} = -100 μA |
| V _{PPLK} | V _{PP} Lock-Out Voltage | 3 | | 1.0 | V | Complete Write Protection |
| V _{PP1} | V _{PP} during Program / Erase Operations | 3 | 1.65 | 3.6 | V | |
| V _{PP2} | | 3,6 | 11.4 | 12.6 | | |
| V _{LKO} | V _{CC} Prog/Erase Lock Voltage | | 1.5 | | V | |
| V _{LKO2} | V _{CCQ} Prog/Erase Lock Voltage | | 1.2 | | V | |

NOTES:

- All currents are in RMS unless otherwise noted. Typical values at nominal V_{CC}, T_A = +25 °C.
- I_{CCES} and I_{CCWS} are specified with device de-selected. If device is read while in erase suspend, current draw is sum of I_{CCES} and I_{CCR}. If the device is read while in program suspend, current draw is the sum of I_{CCWS} and I_{CCR}.
- Erase and Program are inhibited when V_{PP} < V_{PPLK} and not guaranteed outside the valid V_{PP} ranges of V_{PP1} and V_{PP2}.
- Sampled, not 100% tested.
- Automatic Power Savings (APS) reduces I_{CCR} to approximately standby levels in static operation (CMOS inputs).
- Applying V_{PP} = 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_{PP} may be connected to 12 V for a total of 80 hours maximum. See Section 3.4 for details.
- The test conditions V_{CCMax}, V_{CCQMax}, V_{CCMin}, and V_{CCQMin} refer to the maximum or minimum V_{CC} or V_{CCQ} voltage listed at the top of each column.

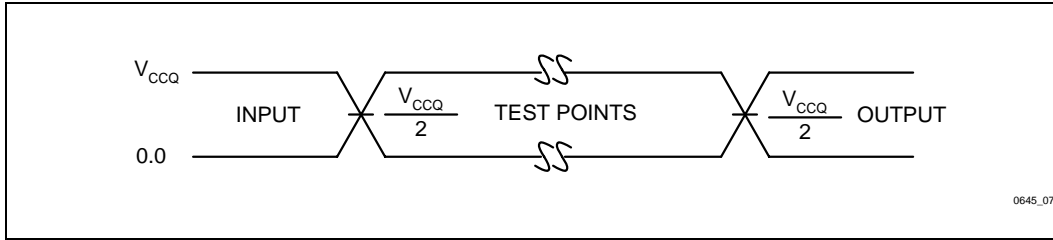


Figure 7. Input Range and Measurement Points

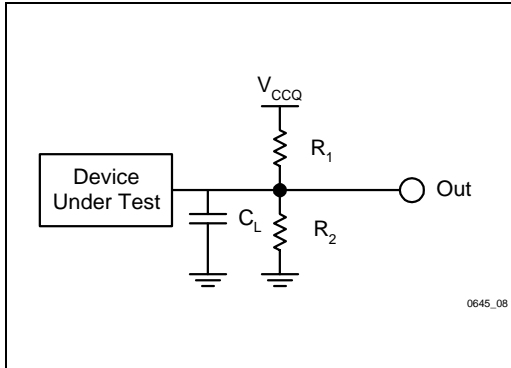


Figure 8. Test Configuration

Test Configuration Component Values Table

| Test Configuration | C _L (pF) | R ₁ (Ω) | R ₂ (Ω) |
|---------------------------|---------------------|--------------------|--------------------|
| 2.7 V–3.6 V Standard Test | 50 | 25K | 25K |

NOTE:
C_L includes jig capacitance.

4.5 AC Characteristics—Read Operations⁽¹⁾—Extended Temperature

| # | Sym | Parameter | Note | Product | | -90 | | -110 | | Unit | | | | |
|-----|-------------------|--|------|-----------------|-----|-------------|-----|-------------|-----|------|-------------|-----|-------------|--|
| | | | | V _{CC} | | 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 | Max | Min | Max | Min | Max | | Min | Max | | |
| R1 | t _{AVAV} | Read Cycle Time | | 80 | | 90 | | 100 | | 110 | ns | | | |
| R2 | t _{AVQV} | Address to Output Delay | | | 80 | | 90 | | 100 | | 110 | ns | | |
| R3 | t _{ELQV} | CE# to Output Delay | 2 | | 80 | | 90 | | 100 | | 110 | ns | | |
| R4 | t _{GLQV} | OE# to Output Delay | 2 | | 30 | | 30 | | 30 | | 30 | ns | | |
| R5 | t _{PHQV} | RP# to Output Delay | | | 150 | | 150 | | 150 | | 150 | ns | | |
| R6 | t _{ELQX} | CE# to Output in Low Z | 3 | 0 | | 0 | | 0 | | 0 | | ns | | |
| R7 | t _{GLQX} | OE# to Output in Low Z | 3 | 0 | | 0 | | 0 | | 0 | | ns | | |
| R8 | t _{EHQZ} | CE# to Output in High Z | 3 | | 20 | | 20 | | 20 | | 20 | ns | | |
| R9 | t _{GHQZ} | OE# to Output in High Z | 3 | | 20 | | 20 | | 20 | | 20 | ns | | |
| R10 | t _{OH} | Output Hold from Address, CE#, or OE# Change, Whichever Occurs First | 3 | 0 | | 0 | | 0 | | 0 | | ns | | |

NOTES:

1. See *AC Waveform: Read Operations*.
2. OE# may be delayed up to t_{ELQV}–t_{GLQV} after the falling edge of CE# without impact on t_{ELQV}.
3. Sampled, but not 100% tested.
4. See Test Configuration (Figure 8).

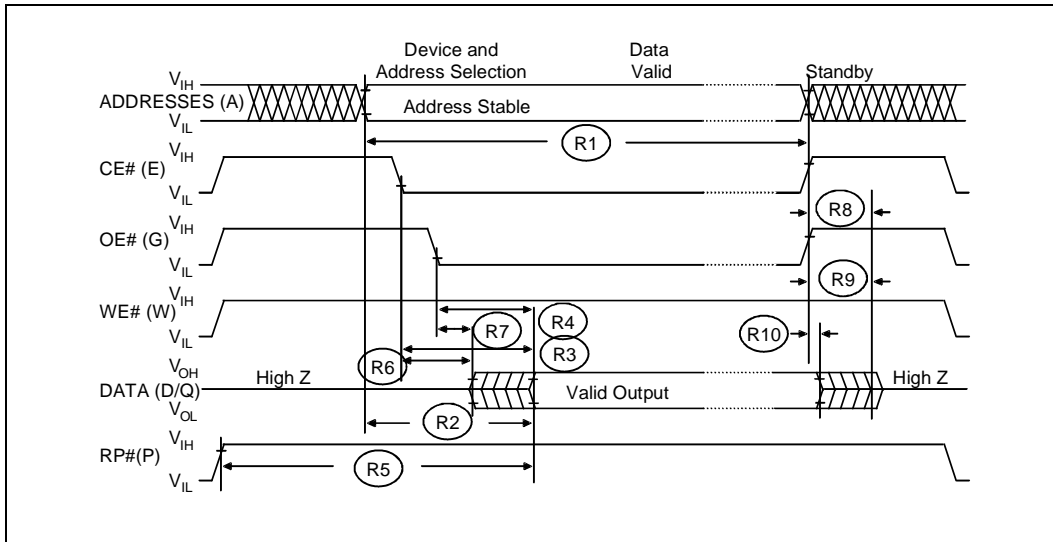


Figure 9. AC Waveform: Read Operations

4.6 AC Characteristics—Write Operations⁽¹⁾—Extended Temperature

| # | Symbol | Parameter | Note | Product | | -90 | | -110 | | Unit |
|-----|----------------------------|--|------|---------------|-----|-----|-----|------|-----|------|
| | | | | 3.0 V – 3.6 V | | 80 | | 100 | | |
| | | | | 2.7 V – 3.6 V | | | 90 | | 110 | |
| Min | Min | Min | Min | Min | Min | Min | Min | | | |
| W1 | t_{PHWL} / t_{PHEL} | RP# High Recovery to WE# (CE#) Going Low | | 150 | 150 | 150 | 150 | ns | | |
| W2 | t_{ELWL} / t_{WLEL} | CE# (WE#) Setup to WE# (CE#) Going Low | | 0 | 0 | 0 | 0 | ns | | |
| W3 | t_{ELEH} / t_{WLWH} | WE# (CE#) Pulse Width | 4 | 50 | 60 | 70 | 70 | ns | | |
| W4 | t_{DVWH} / t_{DVEH} | Data Setup to WE# (CE#) Going High | 2 | 50 | 50 | 60 | 60 | ns | | |
| W5 | t_{AVWH} / t_{AVEH} | Address Setup to WE# (CE#) Going High | 2 | 50 | 60 | 70 | 70 | ns | | |
| W6 | t_{WHEH} / t_{EHWH} | CE# (WE#) Hold Time from WE# (CE#) High | | 0 | 0 | 0 | 0 | ns | | |
| W7 | t_{WHDX} / t_{EHDX} | Data Hold Time from WE# (CE#) High | 2 | 0 | 0 | 0 | 0 | ns | | |
| W8 | t_{WHAX} / t_{EHAX} | Address Hold Time from WE# (CE#) High | 2 | 0 | 0 | 0 | 0 | ns | | |
| W9 | t_{WHWL} / t_{EHEL} | WE# (CE#) Pulse Width High | 4 | 30 | 30 | 30 | 30 | ns | | |
| W10 | t_{VPWH} / t_{VPEH} | V_{PP} Setup to WE# (CE#) Going High | 3 | 200 | 200 | 200 | 200 | ns | | |
| W11 | t_{QVVL} | V_{PP} Hold from Valid SRD | 3 | 0 | 0 | 0 | 0 | ns | | |

NOTES:

- Write timing characteristics during erase suspend are the same as during write-only operations.
- Refer to Table 5 for valid A_{IN} or D_{IN} .
- Sampled, but not 100% tested.
- Write pulse width (t_{WP}) is defined from CE# or WE# going low (whichever goes low last) to CE# or WE# going high (whichever goes high first). Hence, $t_{WP} = t_{WLWH} = t_{ELEH} = t_{WLEH} = t_{ELWH}$. Similarly, Write pulse width high (t_{WPH}) is defined from CE# or WE# going high (whichever goes high first) to CE# or WE# going low (whichever goes low first). Hence, $t_{WPH} = t_{WHWL} = t_{EHEL} = t_{WHEL} = t_{EHWL}$.
- See Test Configuration (Figure 8).

4.7 Erase and Program Timings⁽¹⁾

| Symbol | Parameter | V _{PP} | 1.65 V–3.6 V | | 11.4 V–12.6 V | | Unit |
|---|--|-----------------|--------------------|------|--------------------|------|------|
| | | Note | Typ ⁽¹⁾ | Max | Typ ⁽¹⁾ | Max | |
| t _{BWPB} | 8-KB Parameter Block Program Time (Byte) | 2, 3 | 0.16 | 0.48 | 0.08 | 0.24 | s |
| | 4-KW Parameter Block Program Time (Word) | 2, 3 | 0.10 | 0.30 | 0.03 | 0.12 | s |
| t _{BWMB} | 64-KB Main Block Program Time (Byte) | 2, 3 | 1.2 | 3.7 | 0.6 | 1.7 | s |
| | 32-KW Main Block Program Time(Word) | 2, 3 | 0.8 | 2.4 | 0.24 | 1 | s |
| t _{WHQV1} / t _{EHQV1} | Byte Program Time | 2, 3 | 17 | 165 | 8 | 185 | μs |
| | Word Program Time | 2, 3 | 22 | 200 | 8 | 185 | μs |
| t _{WHQV2} / t _{EHQV2} | 8-KB Parameter Block Erase Time (Byte) | 2, 3 | 1 | 5 | 0.8 | 4.8 | s |
| | 4-KW Parameter Block Erase Time (Word) | 2, 3 | 0.5 | 5 | 0.4 | 4.8 | s |
| t _{WHQV3} / t _{EHQV3} | 64-KB Main Block Erase Time (Byte) | 2, 3 | 1 | 8 | 1 | 7 | s |
| | 32-KW Main Block Erase Time (Word) | 2, 3 | 1 | 8 | 0.6 | 7 | s |
| t _{WHRH1} / t _{EHRH1} | Program Suspend Latency | 3 | 5 | 10 | 5 | 10 | μs |
| t _{WHRH2} / t _{EHRH2} | Erase Suspend Latency | 3 | 5 | 20 | 5 | 20 | μs |

NOTES:

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

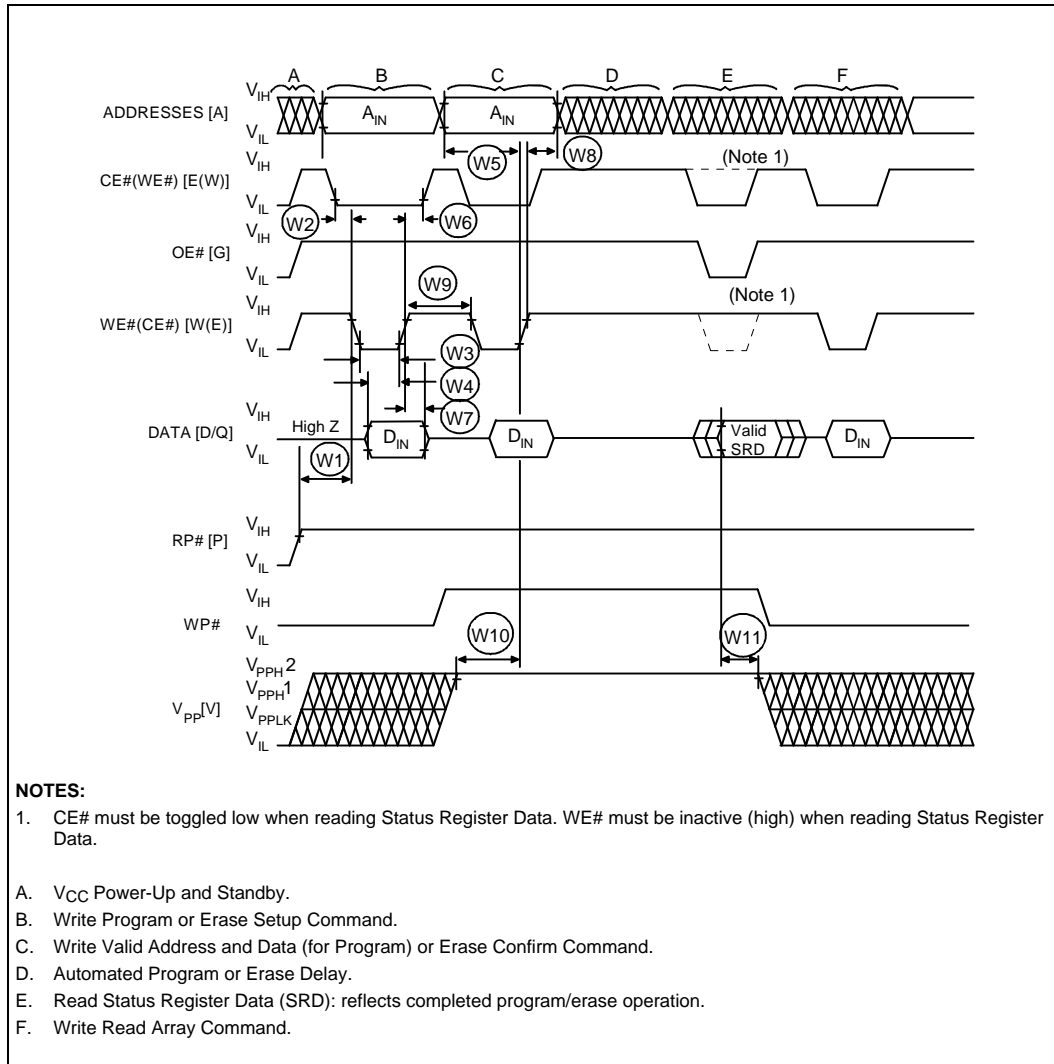


Figure 10. AC Waveform: Program and Erase Operations

4.8 Reset Operations

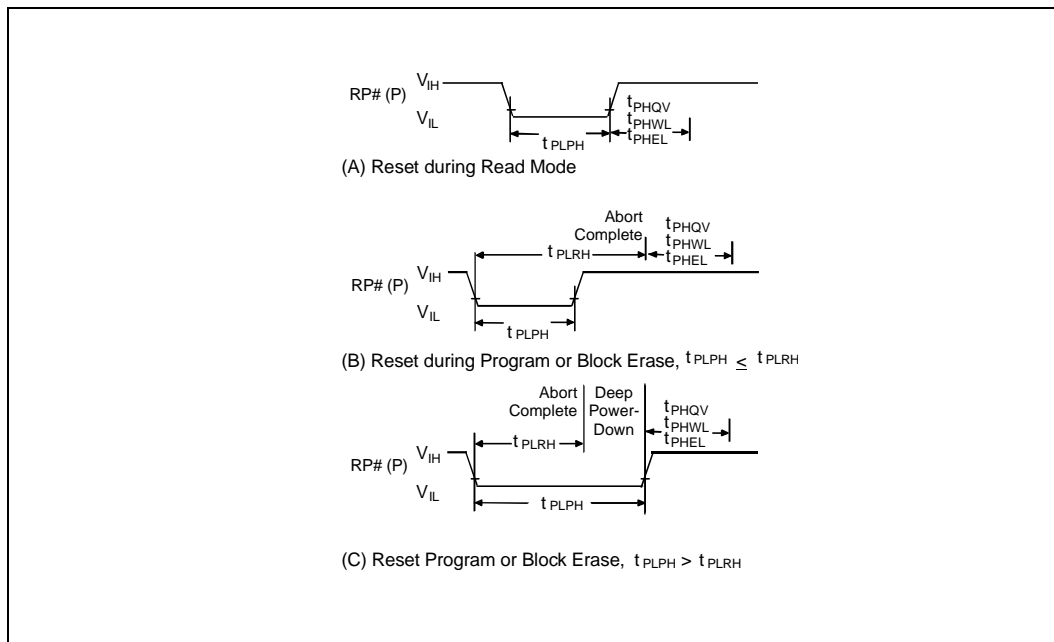


Figure 11. AC Waveform: Reset Operation

Table 11. Reset Specifications⁽¹⁾

| Symbol | Parameter | Notes | V _{CC} 2.7V–3.6V | | Unit |
|--------------------|--|-------|---------------------------|-----|------|
| | | | Min | Max | |
| t _{PLPH} | RP# Low to Reset during Read (If RP# is tied to V _{CC} , this specification is not applicable) | 2,4 | 100 | | ns |
| t _{PLRH1} | RP# Low to Reset during Block Erase | 3,4 | | 22 | μs |
| t _{PLRH2} | RP# Low to Reset during Program | 3,4 | | 12 | μs |

NOTES:

1. See Section 3.1.4 for a full description of these conditions.
2. If t_{PLPH} is < 100 ns the device may still reset but this is not guaranteed.
3. If RP# is asserted while a block erase or word program operation is not executing, the reset will complete within 100 ns.
4. Sampled, but not 100% tested.

5.0 ORDERING INFORMATION

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | E | 2 | 8 | F | 3 | 2 | 0 | C | 3 | T | 9 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Package
TE = 48-Lead TSOP
GT = 48-Ball μ BGA* CSP

Product line designator
for all Intel® Flash products

Device Density
320 = x16 (32 Mbit)
032 = x8 (32 Mbit)
160 = x16 (16 Mbit)
800 = x16 (8 Mbit)
016 = x8 (16 Mbit)
008 = x8 (8 Mbit)

Access Speed (ns)
(90, 110)

T = Top Blocking
B = Bottom Blocking

Product Family
C3 = Advanced+ Boot Block
 $V_{CC} = 2.7\text{ V} - 3.6\text{ V}$
 $V_{PP} = 2.7\text{ V} - 3.6\text{ V}$ or $11.4\text{ V} - 12.6\text{ V}$

VALID COMBINATIONS (All Extended Temperature)

| | 40-Lead TSOP | 48-Ball μ BGA* CSP ⁽¹⁾ | 48-Lead TSOP | 48-Ball μ BGA CSP ⁽¹⁾ |
|---------------------|----------------|---------------------------------------|----------------|--------------------------------------|
| Extended 32M | | GT28F032C3T90 | TE28F320C3T90 | GT28F320C3T90 |
| | | GT28F032C3B90 | TE28F320C3B90 | GT28F320C3B90 |
| Extended 16M | | GT28F032C3T110 | TE28F320C3T110 | GT28F320C3T110 |
| | | GT28F032C3B110 | TE28F320C3B110 | GT28F320C3B110 |
| | TE28F016C3T90 | GT28F016C3T90 | TE28F160C3T90 | GT28F160C3T90 |
| | TE28F016C3B90 | GT28F016C3B90 | TE28F160C3B90 | GT28F160C3B90 |
| Extended 8M | TE28F016C3T110 | GT28F016C3T110 | TE28F160C3T110 | GT28F160C3T110 |
| | TE28F016C3B110 | GT28F016C3B110 | TE28F160C3B110 | GT28F160C3B110 |
| | TE28F008C3T90 | | TE28F800C3T90 | |
| | TE28F008C3B90 | | TE28F800C3B90 | |
| Extended 8M | TE28F008C3T110 | | TE28F800C3T110 | |
| | TE28F008C3B110 | | TE28F800C3B110 | |

NOTE:

1. The 48-Ball μ BGA package top side mark reads FXX0C3 where XX is the device density. This mark is identical for both x8 and x16 products. All product shipping boxes or trays provide the correct information regarding bus architecture, however once the devices are removed from the shipping media, it may be difficult to differentiate based on the top side mark. The device identifier (accessible through the Device ID command: see Section 3.2.2 for further details) enables x8 and x16 μ BGA package product differentiation.

6.0 ADDITIONAL INFORMATION(1,2)

| Order Number | Document/Tool |
|-----------------------------------|---|
| 210830 | <i>1998 Flash Memory Databook</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> |
| | <i>3 Volt Advanced+ Boot Block Algorithms ('C' and assembly)</i> http://developer.intel.com/design/flcomp |
| Contact your Intel Representative | <i>Flash Data Integrator (FDI) Software Developer's Kit</i> |
| 297874 | <i>FDI Interactive: Play with Intel's Flash Data Integrator on Your PC</i> |

NOTES:

1. Please 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. Visit Intel's World Wide Web home page at <http://www.intel.com> or <http://developer.intel.com> for technical documentation and tools.

APPENDIX A WSM CURRENT/NEXT 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 | Program Setup | Erase Setup | Read Array | | | Read Status | Read Array | |
| Read Status | "1" | Status | Read Array | Program Setup | Erase Setup | Read Array | | | Read Status | Read Array | |
| Read Config. | "1" | Config | Read Array | Program Setup | Erase Setup | Read Array | | | Read Status | Read Array | |
| Read Query | "1" | CFI | Read Array | Program Setup | Erase Setup | Read Array | | | Read Status | 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 | Program Setup | Erase Setup | Read Array | | | Read Status | Read Array | |
| Lock Oper. (Done) | "1" | Status | Read Array | Program Setup | Erase Setup | Read Array | | | Read Status | 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 | Program Setup | Erase Setup | Read Array | | | Read Status | 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 | | Program (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 | | Program (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 | | Program (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 | | Program (Not Done) | Prog. Sus. Rd. Array | Program (Not Done) | Prog. Sus. Status | Prog. Sus. Rd. Array | |
| Program (Done) | "1" | Status | Read Array | Program Setup | Erase 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 | Program Setup | Erase 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 | Program 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 | Program 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 | Program 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 | Program Setup | Ers. Sus. Rd. Array | Erase | Ers. Sus. Rd. Array | Erase | Erase Sus. Status | Ers. Sus. Rd. Array | |
| Erase (Done) | "1" | Status | Read Array | Program Setup | Erase Setup | Read Array | | | Read Status | Read Array | |

APPENDIX A WSM CURRENT/NEXT 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 Operation (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 Suspend Status | Erase Suspend Read Config. | Erase Suspend Read Query | Lock Setup | Erase Suspend Read Array | | | Erase (Not Done) |
| Erase Suspend Array | Erase Suspend 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 PROGRAM/ERASE FLOWCHARTS

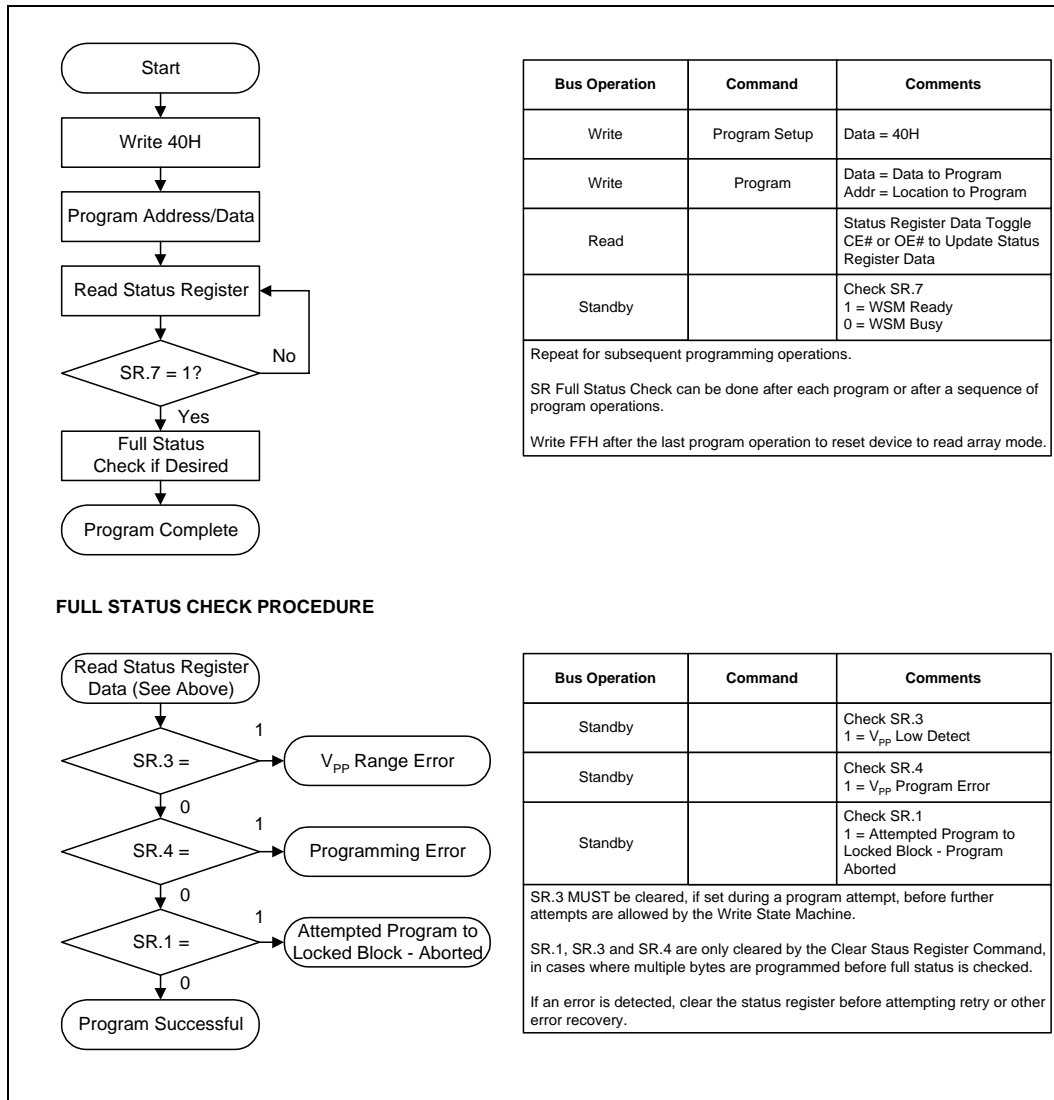


Figure 12. Automated Word Programming Flowchart

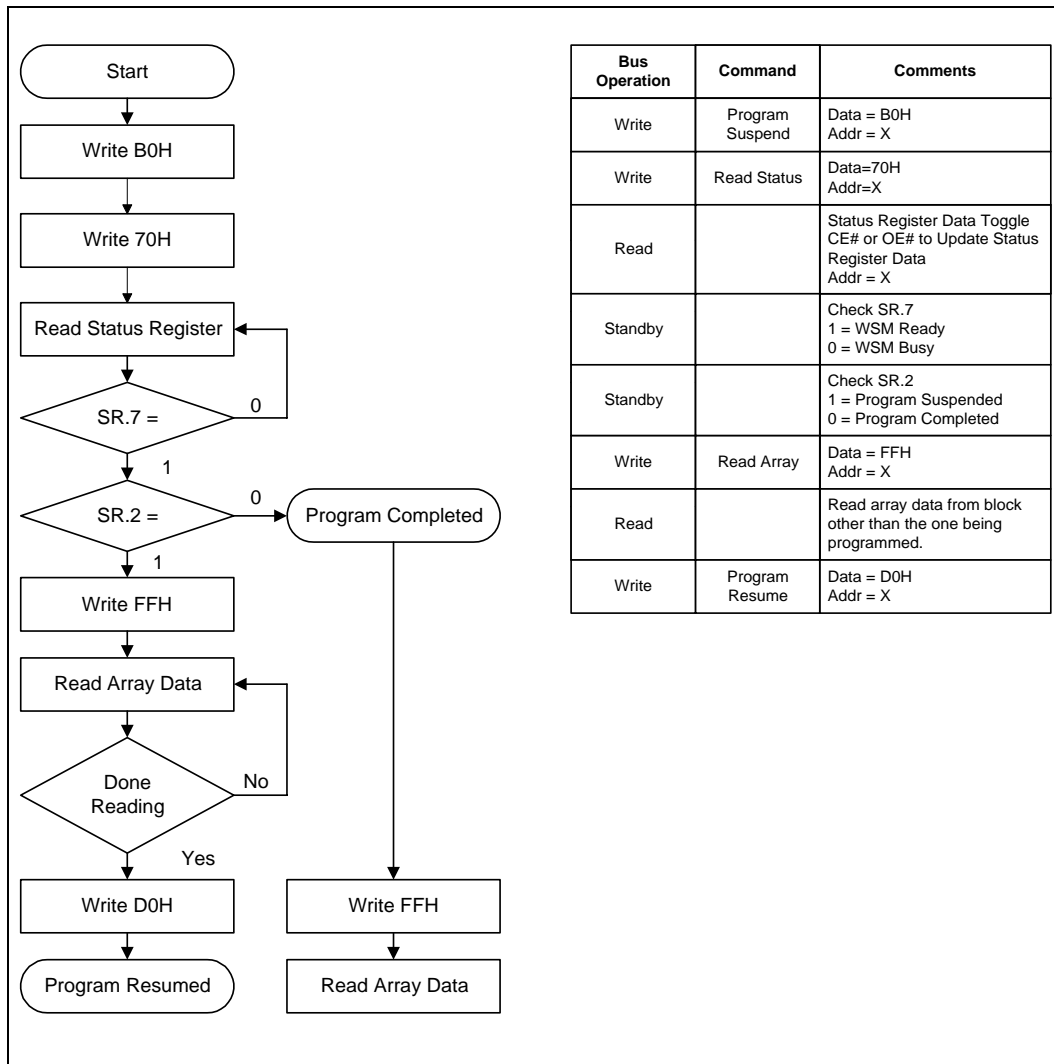


Figure 13. Program Suspend/Resume Flowchart

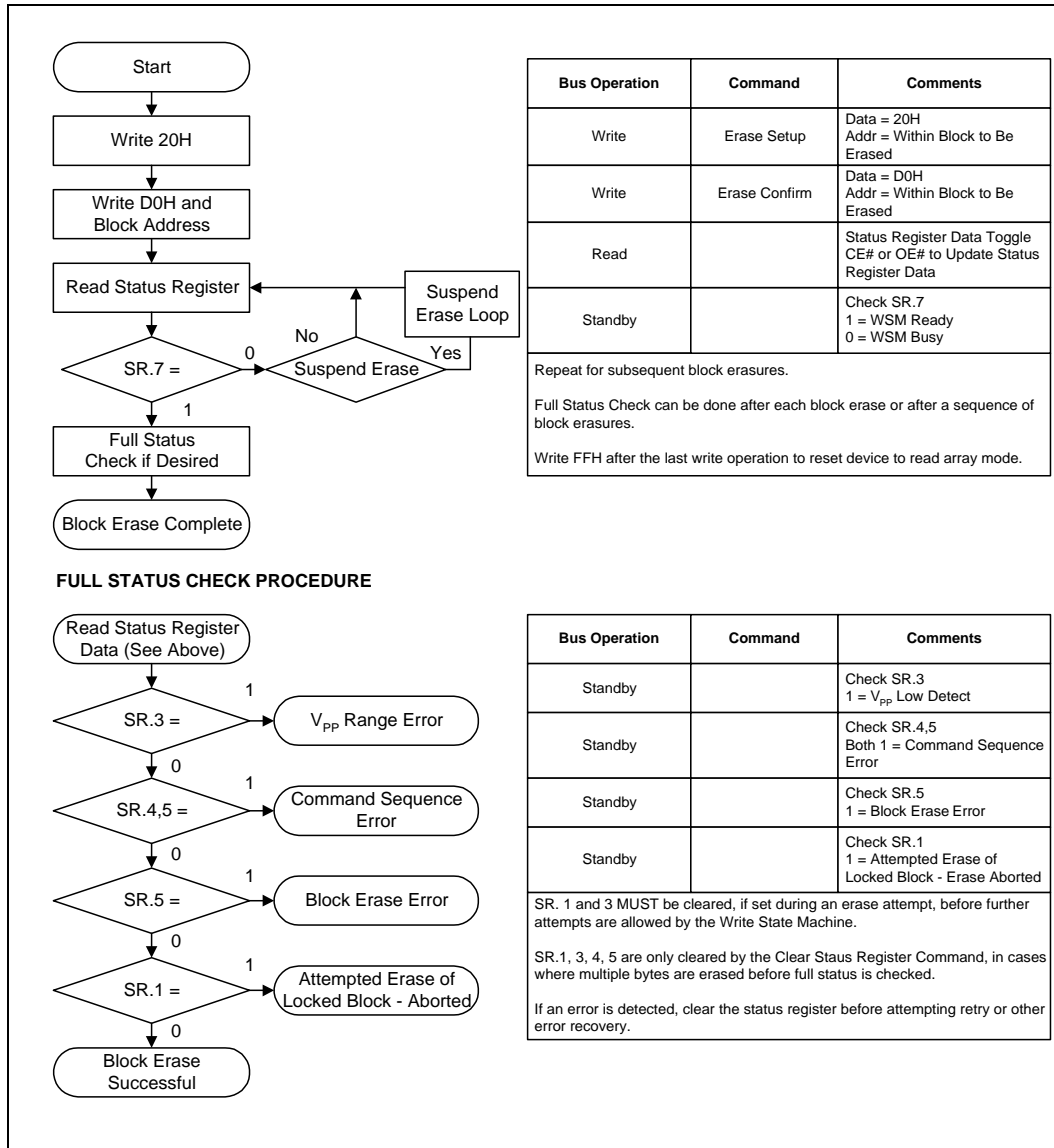


Figure 14. Automated Block Erase Flowchart

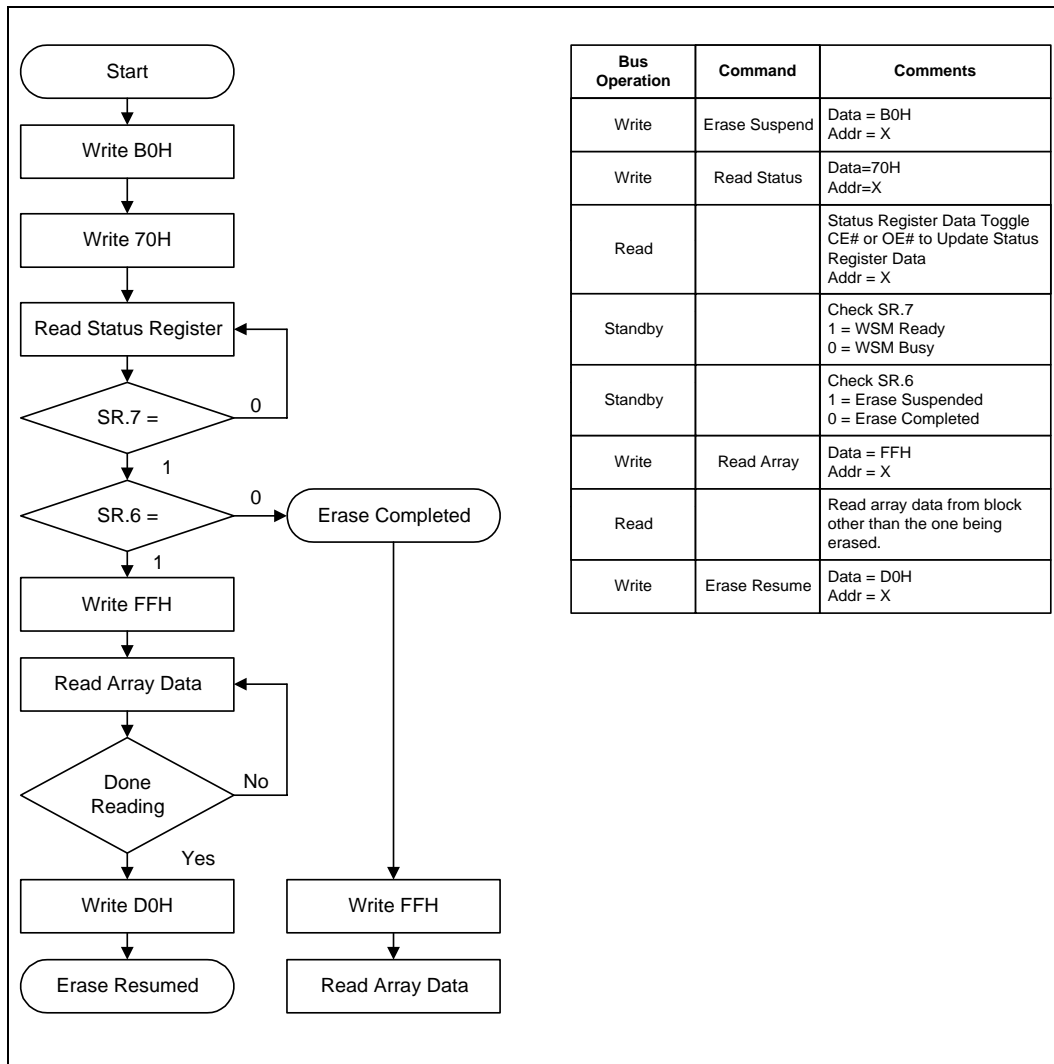


Figure 15. Erase Suspend/Resume Flowchart

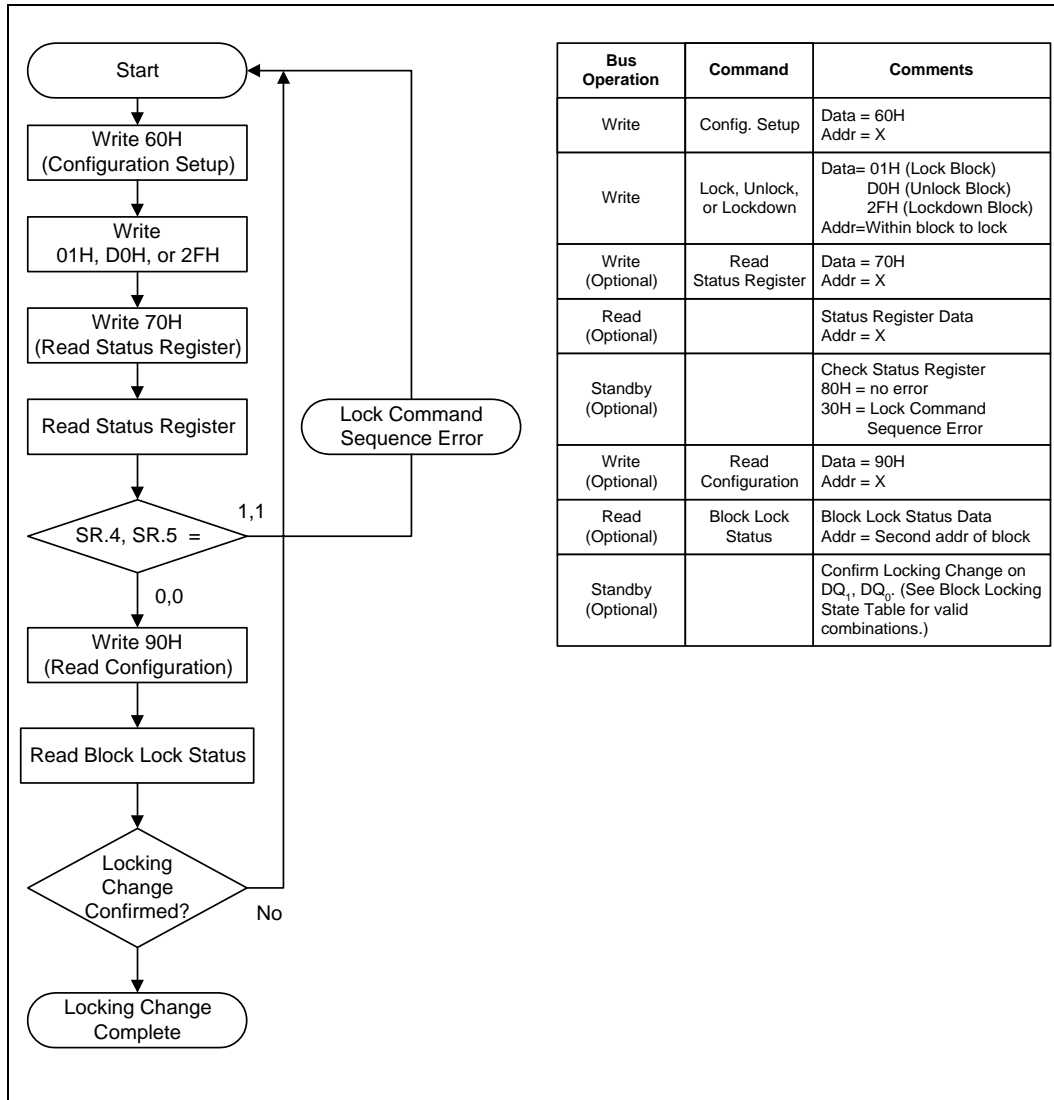


Figure 16. Locking Operations Flowchart

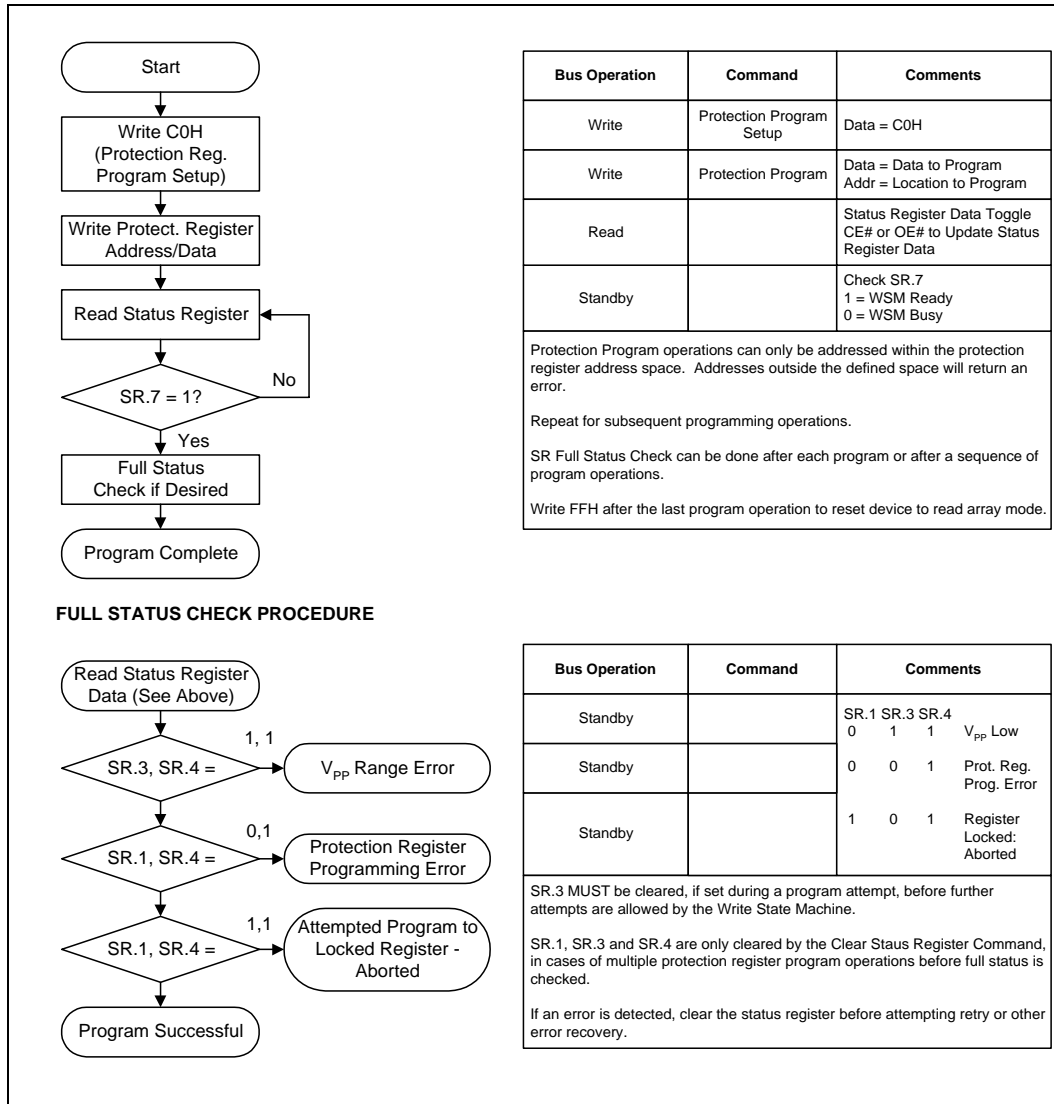


Figure 17. Protection Register Programming Flowchart

APPENDIX C

COMMON FLASH INTERFACE QUERY STRUCTURE

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 will know 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 gain critical information for controlling the flash component. This section describes the device’s CFI-compliant interface that allows the host system to access Query data.

Query data are always presented on the lowest-order data outputs (DQ₀₋₇) only. The numerical offset value is the address relative to the maximum bus width supported by the device. On this family of devices, the Query table device starting address is a 10h, which is a word address for x16 devices or a byte address for x8 devices.

For a word-wide (x16) device, the first two bytes of the Query structure, “Q”, “R”, and “Y” in ASCII, appear on the low byte at word addresses 10h, 11h, and 12h. This CFI-compliant device outputs 00H data on upper bytes. Thus, the device outputs ASCII “Q” in the low byte (DQ₀₋₇) and 00h in the high byte (DQ₈₋₁₅).

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

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

Table C1. Summary of Query Structure Output As a Function of Device and Mode

| Device | Location | Query Data (Hex, ASCII) |
|---|----------|-------------------------|
| 8-Mbit x8/8-Mbit x 16, 16-Mbit x 8/16-Mbit x 16 (Word or Byte Addresses) | 10 | 51 “Q” |
| | 11 | 52 “R” |
| | 12 | 59 “Y” |

Table C2. Example of Query Structure Output of x16 and x8 Devices

| Device Address | Word Addressing: Query Data | Byte Address | Byte Addressing: Query Data |
|---------------------------------|---|--------------------------------|--------------------------------------|
| A ₁₆ -A ₁ | D ₁₅ -D ₀ | A ₇ -A ₀ | D ₇ -D ₀ |
| 0010h | 0051h "Q" | 10h | 51h "Q" |
| 0011h | 0052h "R" | 11h | 52h "R" |
| 0012h | 0059h "Y" | 12h | 59h "Y" |
| 0013h | P_ID _{LO} PrVendor ID# (Lo byte) | 13h | P_ID _{LO} PrVendor ID# (Lo) |
| 0014h | P_ID _{HI} PrVendor ID# (HI byte) | 14h | P_ID _{HI} PrVendor ID# (Hi) |
| 0015h | P _{LO} PrVendor TblAddr (Lo) | 15h | P _{LO} PrVndr TblAdr (Lo) |
| 0016h | P _{HI} PrVendor TblAddr (Hi) | 16h | P _{HI} PrVndr TblAdr (Hi) |
| 0017h | A_ID _{LO} AltVendor ID# (Lo) | 17h | A_ID _{LO} AltVndr ID# (Lo) |
| 0018h | A_ID _{HI} AltVendor ID# (Hi) | 18h | A_ID _{HI} AltVndr ID# (Hi) |
| ... | ... | ... | ... |

C.2 QUERY STRUCTURE OVERVIEW

The Query command causes the flash component to display the Common Flash Interface (CFI) Query structure or "database." The structure sub-sections and address locations are summarized in Table D3.

The following sections describe the Query structure sub-sections in detail.

Table C3. Query Structure⁽¹⁾

| Offset | Sub-Section Name | Description |
|------------------|---|--|
| 00h | | Manufacturer Code |
| 01h | | Device Code |
| 02-0Fh | <i>Reserved</i> | <i>Reserved for vendor-specific information</i> |
| 10h | CFI Query Identification String | Command set ID and vendor data offset |
| 1Bh | System Interface Information | Device timing & voltage information |
| 27h | Device Geometry Definition | Flash device layout |
| P ⁽³⁾ | Primary Intel-specific Extended Query table | Vendor-defined additional information specific to the Primary Vendor Algorithm |

NOTES:

1. Refer to Section D.1 and Table D1 for the detailed definition of offset address as a function of device bus width and mode.
2. BA = The beginning location of a Block Address (e.g., 08000h is the beginning location of block 1 when the block size is 32 Kword).
3. Offset 15 defines "P" which points to the Primary Intel-specific Extended Query Table.

C.3 BLOCK LOCK STATUS

The Block Lock Status indicates the locking settings of a block.

Table C4. Block Lock Status Register

| Offset | Length (bytes) | Description | C3 x16 Device/Mode |
|------------------------|----------------|-------------------|----------------------------|
| (BA+2)h ⁽¹⁾ | 01h | Block Lock Status | BA+2: (see Section 3.3) |

NOTE:

1. BA = The beginning location of a Block Address (i.e., 008000h is the beginning location of block 1 in word mode.)

C.4 CFI QUERY IDENTIFICATION STRING

The Identification String provides verification that the component supports the Common Flash Interface specification. Additionally, it indicates which version of the spec and which vendor-specified command set(s) is (are) supported.

Table C5. CFI Identification

| Offset | Length (Bytes) | Description | 8-Mbit, 16-Mbit, 32-Mbit |
|--------|----------------|---|----------------------------|
| 10h | 03h | Query-Unique ASCII string "QRY" | 10: 51 11: 52 12: 59 |
| 13h | 02h | Primary Vendor Command Set and Control Interface ID Code 16-bit ID Code for Vendor-Specified Algorithms | 13: 03 14: 00 |
| 15h | 02h | Address for Primary Algorithm Extended Query Table Offset value = $P = 35h$ | 15: 35 16: 00 |
| 17h | 02h | Alternate Vendor Command Set and Control Interface ID Code Second Vendor-Specified Algorithm Supported Note: 0000h means none exists | 17: 00 18: 00 |
| 19h | 02h | Address for Secondary Algorithm Extended Query Table Note: 0000h means none exists | 19: 00 1A: 00 |

C.5 SYSTEM INTERFACE INFORMATION

The following device information can be useful in optimizing system interface software

Table C6. System Interface Information

| Offset | Length (bytes) | Description | 8-Mbit, 16-Mbit, 32-Mbit |
|--------|----------------|---|--------------------------|
| 1Bh | 01h | V _{CC} Logic Supply Minimum Program/Erase Voltage bits 7–4 BCD volts bits 3–0 BCD 100 mv | 1B:27 |
| 1Ch | 01h | V _{CC} Logic Supply Maximum Program/Erase Voltage bits 7–4 BCD volts bits 3–0 BCD 100 mv | 1C:36 |
| 1Dh | 01h | V _{PP} [Programming] Supply Minimum Program/Erase Voltage bits 7–4 HEX volts bits 3–0 BCD 100 mv | 1D:B4 |
| 1Eh | 01h | V _{PP} [Programming] Supply Maximum Program/Erase Voltage bits 7–4 HEX volts bits 3–0 BCD 100 mv | 1E:C6 |
| 1Fh | 01h | Typical Time-Out per Single Byte/Word Program, 2 ^N μ-sec | 1F:05 |
| 20h | 01h | Typical Time-Out for Max. Buffer Write, 2 ^N μ-sec | 20:00 |
| 21h | 01h | Typical Time-Out per Individual Block Erase, 2 ^N m-sec | 21:0A |
| 22h | 01h | Typical Time-Out for Full Chip Erase, 2 ^N m-sec | 22:00 |
| 23h | 01h | Maximum Time-Out for Byte/Word Program, 2 ^N Times Typical | 23:04 |
| 24h | 01h | Maximum Time-Out for Buffer Write, 2 ^N Times Typical | 24:00 |
| 25h | 01h | Maximum Time-Out per Individual Block Erase, 2 ^N Times Typical | 25:03 |
| 26h | 01h | Maximum Time-Out for Chip Erase, 2 ^N Times Typical | 26:00 |

C.6 DEVICE GEOMETRY DEFINITION

This field provides critical details of the flash device geometry.

Table C7. Device Geometry Definition

| Offset | Length (bytes) | Description | | | | | | |
|--------------|----------------|---|--------------|----------------|--------------|-----------|--------------|------------|
| 27h | 01h | Device Size = 2 ^N in Number of Bytes | | | | | | |
| 28h | 02h | Flash Device Interface Description <table border="0"> <tr> <td><u>value</u></td> <td><u>meaning</u></td> </tr> <tr> <td>28:00, 29:00</td> <td>x8 asynch</td> </tr> <tr> <td>28:01, 29:00</td> <td>x16 asynch</td> </tr> </table> | <u>value</u> | <u>meaning</u> | 28:00, 29:00 | x8 asynch | 28:01, 29:00 | x16 asynch |
| <u>value</u> | <u>meaning</u> | | | | | | | |
| 28:00, 29:00 | x8 asynch | | | | | | | |
| 28:01, 29:00 | x16 asynch | | | | | | | |
| 2Ah | 02h | Maximum Number of Bytes in Write Buffer = 2 ^N | | | | | | |
| 2Ch | 01h | Number of Erase Block Regions within Device: bits 7-0 = x = # of Erase Block Regions | | | | | | |
| 2Dh | 04h | Erase Block Region Information bits 15-0 = y , Where y+1 = Number of Erase Blocks of Identical Size within Region bits 31-16 = z , Where the Erase Block(s) within This Region are (z) × 256 Bytes | | | | | | |

Device Geometry Definition

| Offset | 8 Mbit | | 16 Mbit | | 32 Mbit | |
|--------|--|--|--|--|--|--|
| | -T | -B | -T | -B | -T | -B |
| 27h | 27:14 | 27:14 | 27:15 | 27:15 | 27:16 | 27:16 |
| 28h | 28:00 (008) 29:00 (008) 28:01 (800) 29:00 (800) | 28:00 (008) 29:00 (008) 28:01 (800) 29:00 (800) | 28:00 (016) 29:00 (016) 28:01 (160) 29:00 (160) | 28:00 (016) 29:00 (016) 28:01 (160) 29:00 (160) | 28:00 (032) 29:00 (032) 28:01 (320) 29:00 (320) | 28:00 (032) 29:00 (032) 28:01 (320) 29:00 (320) |
| 2Ah | 2A:00 2B:00 | 2A:00 2B:00 | 2A:00 2B:00 | 2A:00 2B:00 | 2A:00 2B:00 | 2A:00 2B:00 |
| 2Ch | 2C:02 | 2C:02 | 2C:02 | 2C:02 | 2C:02 | 2C:02 |
| 2Dh | 2D:0E 2E:00 2F:00 30:01 31:07 32:00 33:20 34:00 | 2D:07 2E:00 2F:20 30:00 31:0E 32:00 33:00 34:01 | 2D:1E 2E:00 2F:00 30:01 31:07 32:00 33:20 34:00 | 2D:07 2E:00 2F:20 30:00 31:1E 32:00 33:00 34:01 | 2D:3E 2E:00 2F:00 30:01 31:07 32:00 33:20 34:00 | 2D:07 2E:00 2F:20 30:00 31:3E 32:00 33:00 34:01 |

C.7 INTEL-SPECIFIC EXTENDED QUERY TABLE

Certain flash features and commands are optional. The Intel-Specific Extended Query table specifies this and other similar types of information.

Table C8. Primary-Vendor Specific Extended Query

| Offset ⁽¹⁾ | Length (bytes) | Description | 8-Mbit, 16-Mbit, 32-Mbit |
|-----------------------|----------------|---|--------------------------------------|
| (P)h | 03h | Primary Extended Query Table Unique ASCII String "PRI" | 35: 50 36: 52 37: 49 |
| (P+3)h | 01h | Major Version Number, ASCII | 38: 31 |
| (P+4)h | 01h | Minor Version Number, ASCII | 39: 30 |
| (P+5)h | 04h | Optional Feature & Command Support bit 0 Chip Erase Supported (1=yes, 0=no) bit 1 Suspend Erase Supported (1=yes, 0=no) bit 2 Suspend Program Supported (1=yes, 0=no) bit 3 Lock/Unlock Supported (1=yes, 0=no) bit 4 Queued Erase Supported (1=yes, 0=no) <i>bits 5–31 reserved for future use; undefined bits are "0"</i> | 3A: 06 3B: 00 3C: 00 3D: 00 |
| (P+9)h | 01h | Supported Functions after Suspend Read Array, Status, and Query are always supported during suspended Erase or Program operation. This field defines other operations supported. bit 0 Program Supported after Erase Suspend (1=yes, 0=no) <i>bits 1-7 reserved for future use; undefined bits are "0"</i> | 3E: 01 |
| (P+A)h | 02h | Block Lock Status Defines which bits in the Block Status Register section of the Query are implemented. bit 0 Block Lock Status Register Lock/Unlock bit (bit 0) active (1=yes, 0=no) bit 1 Block Lock Status Register Lock-Down bit (bit 1) active (1=yes, 0=no) <i>Bits 2—15 reserved for future use. Undefined bits are 0.</i> | 3F: 03 40: 00 |

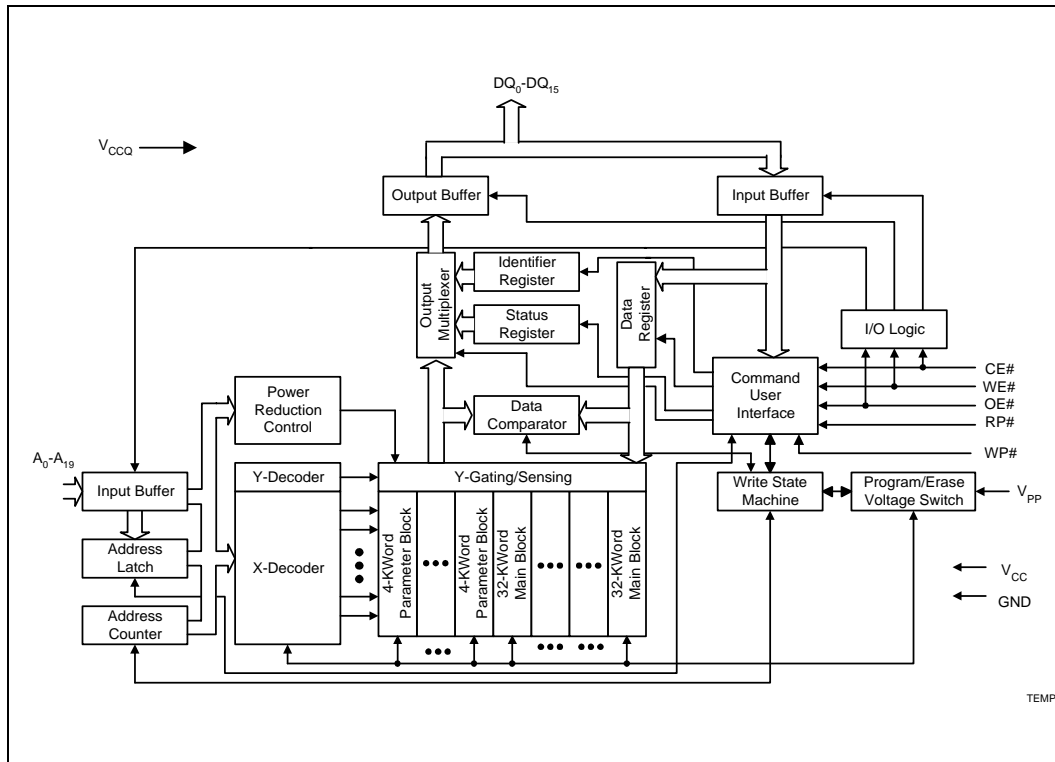
Table C8. Primary-Vendor Specific Extended Query (Continued)

| Offset ⁽¹⁾ | Length (bytes) | Description | 8-Mbit, 16-Mbit, 32-Mbit |
|-----------------------|-----------------|---|--------------------------|
| (P+C)h | 01h | V _{CC} Logic Supply Optimum Program/Erase voltage (highest performance) bits 7–4 BCD value in volts bits 3–0 BCD value in 100 mv | 41: 27 |
| (P+D)h | 01h | V _{PP} [Programming] Supply Optimum Program/Erase voltage bits 7–4 HEX value in volts bits 3–0 BCD value in 100 mv | 42: C0 |
| (P+E)h | <i>Reserved</i> | <i>Reserved for future use</i> | |

NOTE:

1. The variable P is a pointer which is defined at offset 15h in Table D5.

APPENDIX D ARCHITECTURE BLOCK DIAGRAM





APPENDIX E WORD-WIDE MEMORY MAP DIAGRAMS

8-Mbit, 16-Mbit, and 32-Mbit Word-Wide Memory Addressing

| Top Boot | | | | Bottom Boot | | | |
|-----------|-------------|-------------|---------------|-------------|----|-------------|---------------|
| Size (KW) | 8M | 16M | 32M | Size (KW) | 8M | 16M | 32M |
| 4 | 7F000-7FFFF | FF000-FFFFF | 1FF000-1FFFFF | 32 | | | 1F8000-1FFFFF |
| 4 | 7E000-7EFFF | FE000-FEFFF | 1FE000-1FEFFF | 32 | | | 1F0000-1F7FFF |
| 4 | 7D000-7DFFF | FD000-FDFFF | 1FD000-1FDFFF | 32 | | | 1E8000-1EFFFF |
| 4 | 7C000-7CFFF | FC000-FCFFF | 1FC000-1FCFFF | 32 | | | 1E0000-1E7FFF |
| 4 | 7B000-7BFFF | FB000-FBFFF | 1FB000-1FBFFF | 32 | | | 1D8000-1DFFFF |
| 4 | 7A000-7AFFF | FA000-FAFFF | 1FA000-1FAFFF | 32 | | | 1D0000-1D7FFF |
| 4 | 79000-79FFF | F9000-F9FFF | 1F9000-1F9FFF | 32 | | | 1C8000-1CFFFF |
| 4 | 78000-78FFF | F8000-F8FFF | 1F8000-1F8FFF | 32 | | | 1C0000-1C7FFF |
| 32 | 70000-77FFF | F0000-F7FFF | 1F0000-1F7FFF | 32 | | | 1B8000-1BFFFF |
| 32 | 68000-6FFFF | E8000-EFFFF | 1E8000-1EFFFF | 32 | | | 1B0000-1B7FFF |
| 32 | 60000-67FFF | E0000-E7FFF | 1E0000-1E7FFF | 32 | | | 1A8000-1AFFFF |
| 32 | 58000-5FFFF | D8000-DFFFF | 1D8000-1DFFFF | 32 | | | 1A0000-1A7FFF |
| 32 | 50000-57FFF | D0000-D7FFF | 1D0000-1D7FFF | 32 | | | 198000-19FFFF |
| 32 | 48000-4FFFF | C8000-CFFFF | 1C8000-1CFFFF | 32 | | | 190000-197FFF |
| 32 | 40000-47FFF | C0000-C7FFF | 1C0000-1C7FFF | 32 | | | 188000-18FFFF |
| 32 | 38000-37FFF | B8000-BFFFF | 1B8000-1BFFFF | 32 | | | 180000-187FFF |
| 32 | 30000-37FFF | B0000-B7FFF | 1B0000-1B7FFF | 32 | | | 178000-17FFFF |
| 32 | 28000-2FFFF | A8000-AFFFF | 1A8000-1AFFFF | 32 | | | 170000-177FFF |
| 32 | 20000-27FFF | A0000-A7FFF | 1A0000-1A7FFF | 32 | | | 168000-16FFFF |
| 32 | 18000-1FFFF | 98000-9FFFF | 198000-19FFFF | 32 | | | 160000-167FFF |
| 32 | 10000-17FFF | 90000-97FFF | 190000-197FFF | 32 | | | 158000-15FFFF |
| 32 | 08000-0FFFF | 88000-8FFFF | 188000-18FFFF | 32 | | | 150000-157FFF |
| 32 | 00000-07FFF | 80000-87FFF | 180000-187FFF | 32 | | | 148000-14FFFF |
| 32 | | 78000-7FFFF | 178000-17FFFF | 32 | | | 140000-147FFF |
| 32 | | 70000-77FFF | 170000-177FFF | 32 | | | 138000-13FFFF |
| 32 | | 68000-6FFFF | 168000-16FFFF | 32 | | | 130000-137FFF |
| 32 | | 60000-67FFF | 160000-167FFF | 32 | | | 128000-12FFFF |
| 32 | | 58000-5FFFF | 158000-15FFFF | 32 | | | 120000-127FFF |
| 32 | | 50000-57FFF | 150000-157FFF | 32 | | | 118000-11FFFF |
| 32 | | 48000-4FFFF | 148000-14FFFF | 32 | | | 110000-117FFF |
| 32 | | 40000-47FFF | 140000-147FFF | 32 | | | 108000-10FFFF |
| 32 | | 38000-3FFFF | 138000-13FFFF | 32 | | | 100000-107FFF |
| 32 | | 30000-37FFF | 130000-137FFF | 32 | | F8000-FFFFF | 0F8000-0FFFFF |
| 32 | | 28000-2FFFF | 128000-12FFFF | 32 | | F0000-F7FFF | 0F0000-0F7FFF |
| 32 | | 20000-27FFF | 120000-127FFF | 32 | | E8000-EFFFF | 0E8000-0EFFFF |
| 32 | | 18000-1FFFF | 118000-11FFFF | 32 | | E0000-E7FFF | 0E0000-0E7FFF |
| 32 | | 10000-17FFF | 110000-117FFF | 32 | | D8000-DFFFF | 0D8000-0DFFFF |
| 32 | | 08000-0FFFF | 108000-10FFFF | 32 | | D0000-D7FFF | 0D0000-0D7FFF |
| 32 | | 00000-07FFF | 100000-107FFF | 32 | | C8000-CFFFF | 0C8000-0CFFFF |

8-Mbit, 16-Mbit, and 32-Mbit Word-Wide Memory Addressing (Continued)

| Top Boot | | | | Bottom Boot | | | |
|-----------|----|-----|---------------|-------------|-------------|-------------|---------------|
| Size (KW) | 8M | 16M | 32M | Size (KW) | 8M | 16M | 32M |
| 32 | | | 0F8000-0FFFFF | 32 | | C0000-C7FFF | 0C0000-0C7FFF |
| 32 | | | 0F0000-0F7FFF | 32 | | B8000-BFFFF | 0B8000-0BFFFF |
| 32 | | | 0E8000-0EFFFF | 32 | | B0000-B7FFF | 0B0000-0B7FFF |
| 32 | | | 0E0000-0E7FFF | 32 | | A8000-AFFFF | 0A8000-0AFFFF |
| 32 | | | 0D8000-0DFFFF | 32 | | A0000-A7FFF | 0A0000-0A7FFF |
| 32 | | | 0D0000-0D7FFF | 32 | | 98000-9FFFF | 098000-09FFFF |
| 32 | | | 0C8000-0CFFFF | 32 | | 90000-97FFF | 090000-097FFF |
| 32 | | | 0C0000-0C7FFF | 32 | | 88000-8FFFF | 088000-08FFFF |
| 32 | | | 0B8000-0BFFFF | 32 | | 80000-87FFF | 080000-087FFF |
| 32 | | | 0B0000-0B7FFF | 32 | 78000-7FFFF | 78000-7FFFF | 78000-7FFFF |
| 32 | | | 0A8000-0AFFFF | 32 | 70000-77FFF | 70000-77FFF | 70000-77FFF |
| 32 | | | 0A0000-0A7FFF | 32 | 68000-6FFFF | 68000-6FFFF | 68000-6FFFF |
| 32 | | | 098000-09FFFF | 32 | 60000-67FFF | 60000-67FFF | 60000-67FFF |
| 32 | | | 090000-097FFF | 32 | 58000-5FFFF | 58000-5FFFF | 58000-5FFFF |
| 32 | | | 088000-08FFFF | 32 | 50000-57FFF | 50000-57FFF | 50000-57FFF |
| 32 | | | 080000-087FFF | 32 | 48000-4FFFF | 48000-4FFFF | 48000-4FFFF |
| 32 | | | 078000-077FFF | 32 | 40000-47FFF | 40000-47FFF | 40000-47FFF |
| 32 | | | 070000-077FFF | 32 | 38000-3FFFF | 38000-3FFFF | 38000-3FFFF |
| 32 | | | 068000-06FFFF | 32 | 30000-37FFF | 30000-37FFF | 30000-37FFF |
| 32 | | | 060000-067FFF | 32 | 28000-2FFFF | 28000-2FFFF | 28000-2FFFF |
| 32 | | | 058000-05FFFF | 32 | 20000-27FFF | 20000-27FFF | 20000-27FFF |
| 32 | | | 050000-057FFF | 32 | 18000-1FFFF | 18000-1FFFF | 18000-1FFFF |
| 32 | | | 048000-04FFFF | 32 | 10000-17FFF | 10000-17FFF | 10000-17FFF |
| 32 | | | 040000-047FFF | 32 | 08000-0FFFF | 08000-0FFFF | 08000-0FFFF |
| 32 | | | 038000-03FFFF | 4 | 07000-07FFF | 07000-07FFF | 07000-07FFF |
| 32 | | | 030000-037FFF | 4 | 06000-06FFF | 06000-06FFF | 06000-06FFF |
| 32 | | | 028000-02FFFF | 4 | 05000-05FFF | 05000-05FFF | 05000-05FFF |
| 32 | | | 020000-027FFF | 4 | 04000-04FFF | 04000-04FFF | 04000-04FFF |
| 32 | | | 018000-01FFFF | 4 | 03000-03FFF | 03000-03FFF | 03000-03FFF |
| 32 | | | 010000-017FFF | 4 | 02000-02FFF | 02000-02FFF | 02000-02FFF |
| 32 | | | 008000-00FFFF | 4 | 01000-01FFF | 01000-01FFF | 01000-01FFF |
| 32 | | | 000000-007FFF | 4 | 00000-00FFF | 00000-00FFF | 00000-00FFF |



APPENDIX F BYTE-WIDE MEMORY MAP DIAGRAMS

Byte-Wide Memory Addressing

| Top Boot | | | | Bottom Boot | | | |
|-----------|-------------|---------------|----------------|-------------|----|---------------|----------------|
| Size (KB) | 8M | 16M | 32M | Size (KB) | 8M | 16M | 32M |
| 8 | FE000-FFFFF | 1FE000-1FFFFF | 3FE000-3FFFFFF | 64 | | | 3F0000-3FFFFFF |
| 8 | FC000-FDFFF | 1FC000-1FDFFF | 3FC000-3FDFFF | 64 | | | 3E0000-3FFFFFF |
| 8 | FA000-FBFFF | 1FA000-1FBFFF | 3FA000-3FBFFF | 64 | | | 3D0000-3FFFFFF |
| 8 | F8000-F9FFF | 1F8000-1F9FFF | 3F8000-3F9FFF | 64 | | | 3C0000-3FFFFFF |
| 8 | F6000-F7FFF | 1F6000-1F7FFF | 3F6000-3F7FFF | 64 | | | 3B0000-3FFFFFF |
| 8 | F4000-F5FFF | 1F4000-1F5FFF | 3F4000-3F5FFF | 64 | | | 3A0000-3FFFFFF |
| 8 | F2000-F3FFF | 1F2000-1F3FFF | 3F2000-3F3FFF | 64 | | | 390000-39FFFF |
| 8 | F0000-F1FFF | 1F0000-1F1FFF | 3F0000-3F1FFF | 64 | | | 380000-38FFFF |
| 64 | E0000-EFFFF | 1E0000-1EFFFF | 3E0000-3EFFFF | 64 | | | 370000-37FFFF |
| 64 | D0000-DFFFF | 1D0000-1DFFFF | 3D0000-3DFFFF | 64 | | | 360000-36FFFF |
| 64 | C0000-CFFFF | 1C0000-1CFFFF | 3C0000-3CFFFF | 64 | | | 350000-35FFFF |
| 64 | B0000-BFFFF | 1B0000-1BFFFF | 3B0000-3BFFFF | 64 | | | 340000-34FFFF |
| 64 | A0000-AFFFF | 1A0000-1AFFFF | 3A0000-3AFFFF | 64 | | | 330000-33FFFF |
| 64 | 90000-9FFFF | 190000-19FFFF | 390000-39FFFF | 64 | | | 320000-32FFFF |
| 64 | 80000-8FFFF | 180000-18FFFF | 380000-38FFFF | 64 | | | 310000-31FFFF |
| 64 | 70000-7FFFF | 170000-17FFFF | 370000-37FFFF | 64 | | | 300000-30FFFF |
| 64 | 60000-6FFFF | 160000-16FFFF | 360000-36FFFF | 64 | | | 2F0000-2FFFFF |
| 64 | 50000-5FFFF | 150000-15FFFF | 350000-35FFFF | 64 | | | 2E0000-2EFFFF |
| 64 | 40000-4FFFF | 140000-14FFFF | 340000-34FFFF | 64 | | | 2D0000-2DFFFF |
| 64 | 30000-3FFFF | 130000-13FFFF | 330000-33FFFF | 64 | | | 2C0000-2CFFFF |
| 64 | 20000-2FFFF | 120000-12FFFF | 320000-32FFFF | 64 | | | 2B0000-2BFFFF |
| 64 | 10000-1FFFF | 110000-11FFFF | 310000-31FFFF | 64 | | | 2A0000-2AFFFF |
| 64 | 00000-0FFFF | 100000-10FFFF | 300000-30FFFF | 64 | | | 290000-29FFFF |
| 64 | | 0F0000-0FFFFF | 2F0000-2FFFFF | 64 | | | 280000-28FFFF |
| 64 | | 0E0000-0EFFFF | 2E0000-2EFFFF | 64 | | | 270000-27FFFF |
| 64 | | 0D0000-0DFFFF | 2D0000-2DFFFF | 64 | | | 260000-26FFFF |
| 64 | | 0C0000-0CFFFF | 2C0000-2CFFFF | 64 | | | 250000-25FFFF |
| 64 | | 0B0000-0BFFFF | 2B0000-2BFFFF | 64 | | | 240000-24FFFF |
| 64 | | 0A0000-0AFFFF | 2A0000-2AFFFF | 64 | | | 230000-23FFFF |
| 64 | | 090000-09FFFF | 290000-29FFFF | 64 | | | 220000-22FFFF |
| 64 | | 080000-08FFFF | 280000-28FFFF | 64 | | | 210000-21FFFF |
| 64 | | 070000-07FFFF | 270000-27FFFF | 64 | | | 200000-20FFFF |
| 64 | | 060000-06FFFF | 260000-26FFFF | 64 | | 1F0000-1FFFFF | 1F0000-1FFFFF |
| 64 | | 050000-05FFFF | 250000-25FFFF | 64 | | 1E0000-1EFFFF | 1E0000-1EFFFF |
| 64 | | 040000-04FFFF | 240000-24FFFF | 64 | | 1D0000-1DFFFF | 1D0000-1DFFFF |
| 64 | | 030000-03FFFF | 230000-23FFFF | 64 | | 1C0000-1CFFFF | 1C0000-1CFFFF |
| 64 | | 020000-02FFFF | 220000-22FFFF | 64 | | 1B0000-1BFFFF | 1B0000-1BFFFF |
| 64 | | 010000-01FFFF | 210000-21FFFF | 64 | | 1A0000-1AFFFF | 1A0000-1AFFFF |
| 64 | | 000000-00FFFF | 200000-20FFFF | 64 | | 190000-19FFFF | 190000-19FFFF |

Byte-Wide Memory Addressing (Continued)

| Top Boot | | | | Bottom Boot | | | |
|-----------|----|-----|---------------|-------------|-------------|---------------|---------------|
| Size (KB) | 8M | 16M | 32M | Size (KB) | 8M | 16M | 32M |
| 64 | | | 1F0000-1FFFFF | 64 | | 180000-18FFFF | 180000-18FFFF |
| 64 | | | 1E0000-1EFFFF | 64 | | 170000-17FFFF | 170000-17FFFF |
| 64 | | | 1D0000-1DFFFF | 64 | | 160000-16FFFF | 160000-16FFFF |
| 64 | | | 1C0000-1CFFFF | 64 | | 150000-15FFFF | 150000-15FFFF |
| 64 | | | 1B0000-1BFFFF | 64 | | 140000-14FFFF | 140000-14FFFF |
| 64 | | | 1A0000-1AFFFF | 64 | | 130000-13FFFF | 130000-13FFFF |
| 64 | | | 190000-19FFFF | 64 | | 120000-12FFFF | 120000-12FFFF |
| 64 | | | 180000-18FFFF | 64 | | 110000-11FFFF | 110000-11FFFF |
| 64 | | | 170000-17FFFF | 64 | | 100000-10FFFF | 100000-10FFFF |
| 64 | | | 160000-16FFFF | 64 | F0000-FFFFF | 0F0000-0FFFFF | 0F0000-0FFFFF |
| 64 | | | 150000-15FFFF | 64 | E0000-EFFFF | 0E0000-0EFFFF | 0E0000-0EFFFF |
| 64 | | | 140000-14FFFF | 64 | D0000-DFFFF | 0D0000-0DFFFF | 0D0000-0DFFFF |
| 64 | | | 130000-13FFFF | 64 | C0000-CFFFF | 0C0000-0CFFFF | 0C0000-0CFFFF |
| 64 | | | 120000-12FFFF | 64 | B0000-BFFFF | 0B0000-0BFFFF | 0B0000-0BFFFF |
| 64 | | | 110000-11FFFF | 64 | A0000-AFFFF | 0A0000-0AFFFF | 0A0000-0AFFFF |
| 64 | | | 100000-10FFFF | 64 | 90000-9FFFF | 090000-09FFFF | 090000-09FFFF |
| 64 | | | 0F0000-0FFFFF | 64 | 80000-8FFFF | 080000-08FFFF | 080000-08FFFF |
| 64 | | | 0E0000-0EFFFF | 64 | 70000-7FFFF | 070000-07FFFF | 070000-07FFFF |
| 64 | | | 0D0000-0DFFFF | 64 | 60000-6FFFF | 060000-06FFFF | 060000-06FFFF |
| 64 | | | 0C0000-0CFFFF | 64 | 50000-5FFFF | 050000-05FFFF | 050000-05FFFF |
| 64 | | | 0B0000-0BFFFF | 64 | 40000-4FFFF | 040000-04FFFF | 040000-04FFFF |
| 64 | | | 0A0000-0AFFFF | 64 | 30000-3FFFF | 030000-03FFFF | 030000-03FFFF |
| 64 | | | 090000-09FFFF | 64 | 20000-2FFFF | 020000-02FFFF | 020000-02FFFF |
| 64 | | | 080000-08FFFF | 64 | 10000-1FFFF | 010000-01FFFF | 010000-01FFFF |
| 64 | | | 070000-07FFFF | 8 | 0E000-0FFFF | 00E000-00FFFF | 00E000-00FFFF |
| 64 | | | 060000-06FFFF | 8 | 0C000-0DFFF | 00C000-00DFFF | 00C000-00DFFF |
| 64 | | | 050000-05FFFF | 8 | 0A000-0BFFF | 00A000-00BFFF | 00A000-00BFFF |
| 64 | | | 040000-04FFFF | 8 | 08000-09FFF | 008000-009FFF | 008000-009FFF |
| 64 | | | 030000-03FFFF | 8 | 06000-07FFF | 006000-007FFF | 006000-007FFF |
| 64 | | | 020000-02FFFF | 8 | 04000-05FFF | 004000-005FFF | 004000-005FFF |
| 64 | | | 010000-01FFFF | 8 | 02000-03FFF | 002000-003FFF | 002000-003FFF |
| 64 | | | 000000-00FFFF | 8 | 00000-01FFF | 000000-001FFF | 000000-001FFF |

APPENDIX G DEVICE ID TABLE

Read Configuration Addresses and Data

| Item | | Address | Data |
|-------------------|-----|---------|------|
| Manufacturer Code | x16 | 00000 | 0089 |
| | x8 | 00000 | 89 |
| Device Code | | | |
| 8-Mbit x 16-T | x16 | 00001 | 88C0 |
| 8-Mbit x 16-B | x16 | 00001 | 88C1 |
| 16-Mbit x 16-T | x16 | 00001 | 88C2 |
| 16-Mbit x 16-B | x16 | 00001 | 88C3 |
| 32-Mbit x 16-T | x16 | 00001 | 88C4 |
| 32-Mbit x 16-B | x16 | 00001 | 88C5 |
| 8-Mbit x 8-T | x8 | 00001 | C0 |
| 8-Mbit x 8-B | x8 | 00001 | C1 |
| 16-Mbit x 8-T | x8 | 00001 | C2 |
| 16-Mbit x 8-B | x8 | 00001 | C3 |
| 32-Mbit x 8-T | x8 | 00001 | C4 |
| 32-Mbit x 8-B | x8 | 00001 | C5 |

NOTE: Other locations within the configuration address space are reserved by Intel for future use.

APPENDIX H PROTECTION REGISTER ADDRESSING

Word-Wide Protection Register Addressing

| Word | Use | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|------|---------|----|----|----|----|----|----|----|----|
| LOCK | Both | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | Factory | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | Factory | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | Factory | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | Factory | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 4 | User | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 5 | User | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 6 | User | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 7 | User | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Byte-Wide Protection Register Addressing

| Byte | Use | A11 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|------|---------|-----|----|----|----|----|----|----|----|----|
| LOCK | Both | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | Factory | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | Factory | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | Factory | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | Factory | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | Factory | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 5 | Factory | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 6 | Factory | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 7 | Factory | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | User | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 9 | User | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 10 | User | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 11 | User | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 12 | User | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 13 | User | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 14 | User | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 15 | User | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |