

# Micron Parallel NOR Flash Embedded Memory (P33-65nm)

**JS28F512P33BFD, JS28F512P33TFA, JS28F512P33EFA**  
**PC28F512P33BFD, PC28F512P33TFA, PC28F512P33EFA**  
**JS28F00AP33BFA, JS28F00AP33TFA, JS28F00AP33EFA**  
**PC28F00AP33BFA, PC28F00AP33TFA, PC28F00AP33EFA,**  
**PC28F00BP33EFA**

## Features

- High performance
- Easy BGA package features
  - 95ns initial access for 512Mb, 1Gb Easy BGA
  - 100ns initial access for 2Gb Easy BGA
  - 25ns 16-word asynchronous page read mode
  - 52 MHz (Easy BGA) with zero WAIT states and 17ns clock-to-data output synchronous burst read mode
  - 4-, 8-, 16-, and continuous word options for burst mode
- TSOP package features
  - 105ns initial access for 512Mb, 1Gb TSOP
- Both Easy BGA and TSOP package features
  - Buffered enhanced factory programming (BEFP) at 2 MB/s (TYP) using a 512-word buffer
  - 3.0V buffered programming at 1.46 MB/s (TYP) using a 512-word buffer
- Architecture
  - MLC: highest density at lowest cost
  - Symmetrically blocked architecture (512Mb, 1Gb, 2Gb)
  - Asymmetrically blocked architecture (512Mb, 1Gb); four 32KB parameter blocks: top or bottom configuration
  - 128KB main blocks
  - Blank check to verify an erased block
- Voltage and power
  - $V_{CC}$  (core) voltage: 2.3–3.6V
  - $V_{CCQ}$  (I/O) voltage: 2.3–3.6V
  - Standby current: 70 $\mu$ A (TYP) for 512Mb; 75 $\mu$ A (TYP) for 1Gb
  - 52 MHz continuous synchronous read current: 21mA (TYP), 24mA (MAX)
- Security
  - One-time programmable register: 64 OTP bits, programmed with unique information from Micron; 2112 OTP bits available for customer programming
  - Absolute write protection:  $V_{PP} = V_{SS}$
  - Power-transition erase/program lockout
  - Individual zero-latency block locking
  - Individual block lock-down
  - Password access
- Software
  - 25 $\mu$ s (TYP) program suspend
  - 25 $\mu$ s (TYP) erase suspend
  - Flash Data Integrator optimized
  - Basic command set and extended function Interface (EFI) command set compatible
  - Common flash interface
- Density and Packaging
  - 56-lead TSOP package (512Mb, 1Gb)
  - 64-ball Easy BGA package (512Mb, 1Gb, 2Gb)
  - 16-bit wide data bus
- Quality and reliability
  - JESD47 compliant
  - Operating temperature: –40°C to +85°C
  - Minimum 100,000 ERASE cycles per block
  - 65nm process technology

## Discrete and MCP Part Numbering Information

Devices are shipped from the factory with memory content bits erased to 1. For available options, such as packages or for further information, contact your Micron sales representative. Part numbers can be verified at [www.micron.com](http://www.micron.com). Feature and specification comparison by device type is available at [www.micron.com/products](http://www.micron.com/products). Contact the factory for devices not found.

**Note:** Not all part numbers listed here are available for ordering.

**Table 1: Discrete Part Number Information**

| Part Number Category | Category Details   |
|----------------------|--|
| Package              | JS = 56-lead TSOP, lead free   |
|                      | PC = 64-ball Easy BGA, lead-free   |
| Product Line         | 28F = Micron Flash memory  |
| Density              | 512 = 512Mb<br>00A = 1Gb<br>00B = 2Gb  |
| Product Family       | P33 ( $V_{CC} = 2.3\text{--}3.6\text{V}$ ; $V_{CCQ} = 2.3\text{--}3.6\text{V}$ ) |
| Parameter Location   | B/T = Bottom/Top parameter<br>E = Symmetrical Blocks                             |
| Lithography          | F = 65nm   |
| Features             | *  |

**Note:** 1. The last digit is assigned randomly to cover packaging media, features, or other specific configuration information. Sample part number: JS28F512P33EF\*

**Table 2: Standard Part Numbers**

| Density | Configuration | Medium      | JS             | PC             |
|---------|---------------|-------------|----------------|----------------|
| 512Mb   | Bottom boot   | Tray        | JS28F512P33BFD | PC28F512P33BFD |
|         |               | Tape & Reel | –              | –              |
|         | Top boot      | Tray        | JS28F512P33TFA | PC28F512P33TFA |
|         |               | Tape & Reel | –              | –              |
|         | Uniform       | Tray        | JS28F512P33EFA | PC28F512P33EFA |
|         |               | Tape & Reel | –              | –              |
| 1Gb     | Bottom boot   | Tray        | JS28F00AP33BFA | PC28F00AP33BFA |
|         |               | Tape & Reel | –              | –              |
|         | Top boot      | Tray        | JS28F00AP33TFA | PC28F00AP33TFA |
|         |               | Tape & Reel | –              | –              |
|         | Uniform       | Tray        | JS28F00AP33EFA | PC28F00AP33EFA |
|         |               | Tape & Reel | –              | –              |
| 2Gb     | Uniform       | Tray        | –              | PC28F00BP33EFA |
|         |               | Tape & Reel | –              | –              |



## Contents

|  |    |
|--|----|
| General Description .....                              | 7  |
| Virtual Chip Enable Description .....                  | 8  |
| Memory Map .....                                       | 9  |
| Package Dimensions .....                               | 11 |
| Pinouts and Ballouts .....                             | 13 |
| Signal Descriptions .....                              | 15 |
| Bus Operations .....                                   | 17 |
| Read .....   | 17 |
| Write .....  | 17 |
| Output Disable .....                                   | 17 |
| Standby .....  | 17 |
| Reset .....  | 18 |
| Device Command Codes .....                             | 19 |
| Device Command Bus Cycles .....                        | 22 |
| Read Operations .....                                  | 24 |
| Asynchronous Single Word Read .....                    | 24 |
| Asynchronous Page Mode Read (Easy BGA Only) .....      | 24 |
| Synchronous Burst Mode Read (Easy BGA Only) .....      | 25 |
| Read CFI .....   | 25 |
| Read Device ID .....                                   | 25 |
| Device ID Codes .....                                  | 26 |
| Program Operations .....                               | 27 |
| Word Programming (40h) .....                           | 27 |
| Buffered Programming (E8h, D0h) .....                  | 27 |
| Buffered Enhanced Factory Programming (80h, D0h) ..... | 28 |
| Program Suspend .....                                  | 30 |
| Program Resume .....                                   | 31 |
| Program Protection .....                               | 31 |
| Erase Operations .....                                 | 32 |
| BLOCK ERASE Command .....                              | 32 |
| BLANK CHECK Command .....                              | 32 |
| ERASE SUSPEND Command .....                            | 33 |
| ERASE RESUME Command .....                             | 33 |
| Erase Protection .....                                 | 33 |
| Security Operations .....                              | 34 |
| Block Locking .....                                    | 34 |
| BLOCK LOCK Command .....                               | 34 |
| BLOCK UNLOCK Command .....                             | 34 |
| BLOCK LOCK DOWN Command .....                          | 34 |
| Block Lock Status .....                                | 34 |
| Block Locking During Suspend .....                     | 35 |
| Selectable OTP Blocks .....                            | 36 |
| Password Access .....                                  | 36 |
| Status Register .....                                  | 37 |
| Read Status Register .....                             | 37 |
| Clear Status Register .....                            | 38 |
| Configuration Register .....                           | 39 |
| Read Configuration Register .....                      | 39 |
| Read Mode .....  | 39 |
| Latency Count .....                                    | 40 |



|  |    |
|--|----|
| End of Wordline Considerations .....           | 41 |
| WAIT Signal Polarity and Functionality .....   | 42 |
| WAIT Delay .....                               | 43 |
| Burst Sequence .....                           | 43 |
| Clock Edge .....                               | 44 |
| Burst Wrap .....                               | 44 |
| Burst Length .....                             | 44 |
| One-Time Programmable Registers .....          | 45 |
| Read OTP Registers .....                       | 45 |
| Program OTP Registers .....                    | 46 |
| Lock OTP Registers .....                       | 46 |
| Common Flash Interface .....                   | 48 |
| READ CFI Structure Output .....                | 48 |
| Flowcharts .....                               | 62 |
| Power and Reset Specifications .....           | 71 |
| Power Supply Decoupling .....                  | 72 |
| Maximum Ratings and Operating Conditions ..... | 73 |
| DC Electrical Specifications .....             | 74 |
| AC Test Conditions and Capacitance .....       | 76 |
| AC Read Specifications .....                   | 78 |
| AC Write Specifications .....                  | 85 |
| Program and Erase Characteristics .....        | 91 |
| Revision History .....                         | 92 |
| Rev. C – 12/13 .....                           | 92 |
| Rev. B – 11/13 .....                           | 92 |
| Rev. A – 8/13 .....                            | 92 |

## List of Figures

|  |    |
|--|----|
| Figure 1: Easy BGA Block Diagram .....   | 8  |
| Figure 2: Memory Map – 512Mb and 1Gb .....                                     | 9  |
| Figure 3: Memory Map – 2Gb .....   | 10 |
| Figure 4: 56-Pin TSOP – 14mm x 20mm .....                                      | 11 |
| Figure 5: 64-Ball Easy BGA – 8mm x 10mm x 1.2mm .....                          | 12 |
| Figure 6: 56-Lead TSOP Pinout – 512Mb and 1Gb .....                            | 13 |
| Figure 7: 64-Ball Easy BGA (Top View – Balls Down) – 512Mb, 1Gb, and 2Gb ..... | 14 |
| Figure 8: Example $V_{pp}$ Supply Connections .....                            | 31 |
| Figure 9: Block Locking State Diagram .....                                    | 35 |
| Figure 10: First Access Latency Count .....                                    | 40 |
| Figure 11: Example Latency Count Setting Using Code 3 .....                    | 41 |
| Figure 12: End of Wordline Timing Diagram .....                                | 41 |
| Figure 13: OTP Register Map .....  | 46 |
| Figure 14: Word Program Procedure .....  | 62 |
| Figure 15: Buffer Program Procedure .....                                      | 63 |
| Figure 16: Buffered Enhanced Factory Programming (BEFP) Procedure .....        | 64 |
| Figure 17: Block Erase Procedure .....   | 65 |
| Figure 18: Program Suspend/Resume Procedure .....                              | 66 |
| Figure 19: Erase Suspend/Resume Procedure .....                                | 67 |
| Figure 20: Block Lock Operations Procedure .....                               | 68 |
| Figure 21: OTP Register Programming Procedure .....                            | 69 |
| Figure 22: Status Register Procedure .....                                     | 70 |
| Figure 23: Reset Operation Waveforms .....                                     | 72 |
| Figure 24: AC Input/Output Reference Timing .....                              | 76 |
| Figure 25: Transient Equivalent Load Circuit .....                             | 76 |
| Figure 26: Clock Input AC Waveform .....                                       | 76 |
| Figure 27: Asynchronous Single-Word Read (ADV# LOW) .....                      | 80 |
| Figure 28: Asynchronous Single-Word Read (ADV# Latch) .....                    | 80 |
| Figure 29: Asynchronous Page Mode Read .....                                   | 81 |
| Figure 30: Synchronous Single-Word Array or Nonarray Read .....                | 82 |
| Figure 31: Continuous Burst Read with Output Delay .....                       | 83 |
| Figure 32: Synchronous Burst Mode 4-Word Read .....                            | 84 |
| Figure 33: Write to Write Timing .....   | 87 |
| Figure 34: Asynchronous Read to Write Timing .....                             | 87 |
| Figure 35: Write to Asynchronous Read Timing .....                             | 88 |
| Figure 36: Synchronous Read to Write Timing .....                              | 89 |
| Figure 37: Write to Synchronous Read Timing .....                              | 90 |

## List of Tables

|   |    |
|---|----|
| Table 1: Discrete Part Number Information .....                                     | 2  |
| Table 2: Standard Part Numbers .....  | 2  |
| Table 3: Virtual Chip Enable Truth Table for Easy BGA Packages .....                | 8  |
| Table 4: TSOP and Easy BGA Signal Descriptions .....                                | 15 |
| Table 5: Bus Operations .....   | 17 |
| Table 6: Command Codes and Definitions .....  | 19 |
| Table 7: Command Bus Cycles .....   | 22 |
| Table 8: Device ID Information .....  | 25 |
| Table 9: Device ID codes .....  | 26 |
| Table 10: BEFP Requirements .....   | 29 |
| Table 11: BEFP Considerations .....   | 29 |
| Table 12: Status Register Description .....   | 37 |
| Table 13: Read Configuration Register .....   | 39 |
| Table 14: End of Wordline Data and WAIT State Comparison .....                      | 42 |
| Table 15: WAIT Functionality Table .....  | 42 |
| Table 16: Burst Sequence Word Ordering .....  | 43 |
| Table 17: Example of CFI Output (x16 device) as a Function of Device and Mode ..... | 48 |
| Table 18: CFI Database: Addresses and Sections .....                                | 49 |
| Table 19: CFI ID String .....   | 49 |
| Table 20: System Interface Information .....  | 50 |
| Table 21: Device Geometry .....   | 51 |
| Table 22: Block Region Map Information .....  | 51 |
| Table 23: Primary Vendor-Specific Extended Query .....                              | 52 |
| Table 24: Optional Features Field .....   | 54 |
| Table 25: One Time Programmable (OTP) Space Information .....                       | 54 |
| Table 26: Burst Read Information .....  | 55 |
| Table 27: Partition and Block Erase Region Information .....                        | 56 |
| Table 28: Partition Region 1 Information: Top and Bottom Offset/Address .....       | 57 |
| Table 29: Partition Region 1 Information .....                                      | 57 |
| Table 30: Partition Region 1: Partition and Erase Block Map Information .....       | 60 |
| Table 31: CFI Link Information – 2Gb .....  | 61 |
| Table 32: Power and Reset .....   | 71 |
| Table 33: Maximum Ratings .....   | 73 |
| Table 34: Operating Conditions .....  | 73 |
| Table 35: DC Current Characteristics .....  | 74 |
| Table 36: DC Voltage Characteristics .....  | 75 |
| Table 37: Test Configuration: Worst-Case Speed Condition .....                      | 76 |
| Table 38: Capacitance .....   | 77 |
| Table 39: AC Read Specifications .....  | 78 |
| Table 40: AC Write Specifications .....   | 85 |
| Table 41: Program and Erase Specifications .....                                    | 91 |

## General Description

The Micron Parallel NOR Flash memory is the latest generation of Flash memory devices. Benefits include more density in less space, high-speed interface device, and support for code and data storage. Features include high-performance synchronous-burst read mode, fast asynchronous access times, low power, flexible security options, and three industry-standard package choices. The product family is manufactured using Micron 65nm process technology.

The NOR Flash device provides high performance at low voltage on a 16-bit data bus. Individually erasable memory blocks are sized for optimum code and data storage.

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

In addition to the enhanced architecture and interface, the device incorporates technology that enables fast factory PROGRAM and ERASE operations. Designed for low-voltage systems, the device supports READ operations with  $V_{CC}$  at the low voltages, and ERASE and PROGRAM operations with  $V_{PP}$  at the low voltages or  $V_{PPH}$ . Buffered enhanced factory programming (BEFP) provides the fastest Flash array programming performance with  $V_{PP}$  at  $V_{PPH}$ , which increases factory throughput. With  $V_{PP}$  at low voltages,  $V_{CC}$  and  $V_{PP}$  can be tied together for a simple, ultra low-power design. In addition to voltage flexibility, a dedicated  $V_{PP}$  connection provides complete data protection when  $V_{PP} \leq V_{PPLK}$ .

A command user interface is the interface between the system processor and all internal operations of the device. The device automatically executes the algorithms and timings necessary for block erase and program. A status register indicates ERASE or PROGRAM completion and any errors that may have occurred.

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

The protection register enables unique device identification that can be used to increase system security. The individual block lock feature provides zero-latency block locking and unlocking. The device includes enhanced protection via password access; this new feature supports write and/or read access protection of user-defined blocks. In addition, the device also provides the full-device OTP security feature.

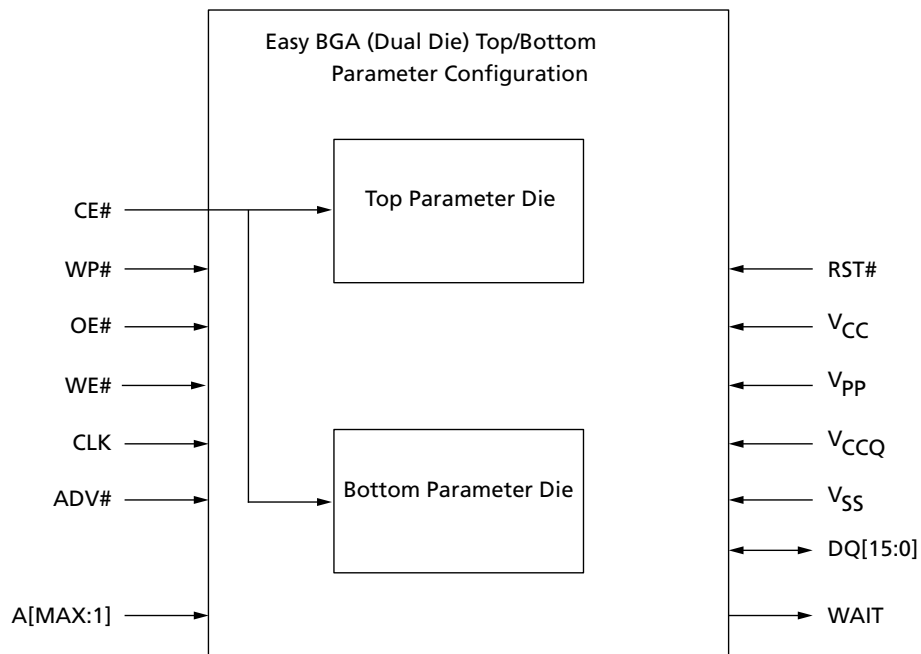
## Virtual Chip Enable Description

The 2Gb device employs a virtual chip enable feature, which combines two 1Gb die with a common chip enable, CE# for Easy BGA packages. The maximum address bit is then used to select between the die pair with CE# asserted. When CE# is asserted and the maximum address bit is LOW, the lower parameter die is selected; when CE# is asserted and the maximum address bit is HIGH, the upper parameter die is selected.

**Table 3: Virtual Chip Enable Truth Table for Easy BGA Packages**

| Die Selected        | CE# | A[MAX] |
|---------------------|-----|--------|
| Lower parameter die | L   | L      |
| Upper parameter die | L   | H      |

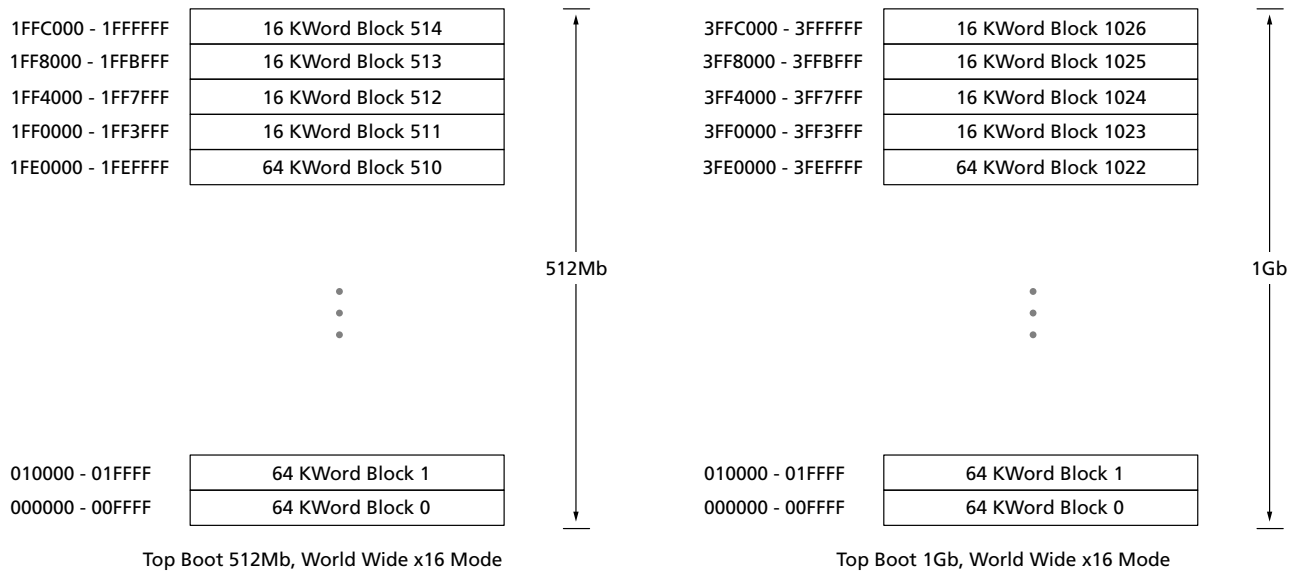
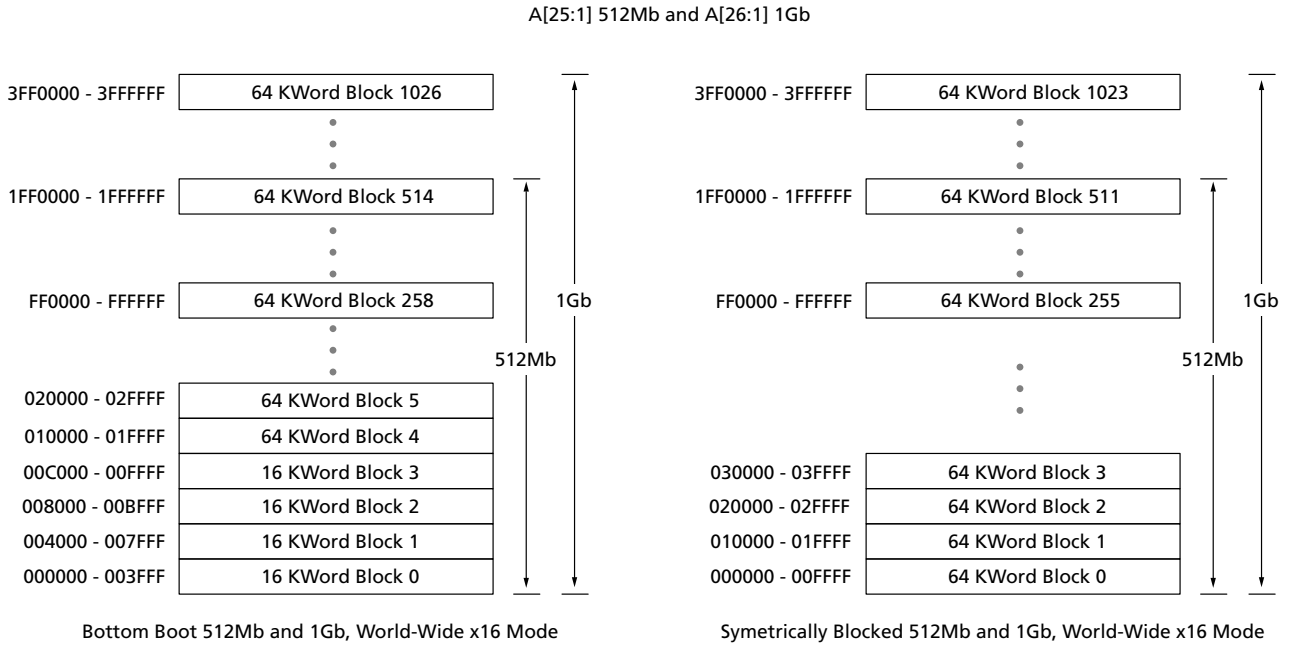
**Figure 1: Easy BGA Block Diagram**



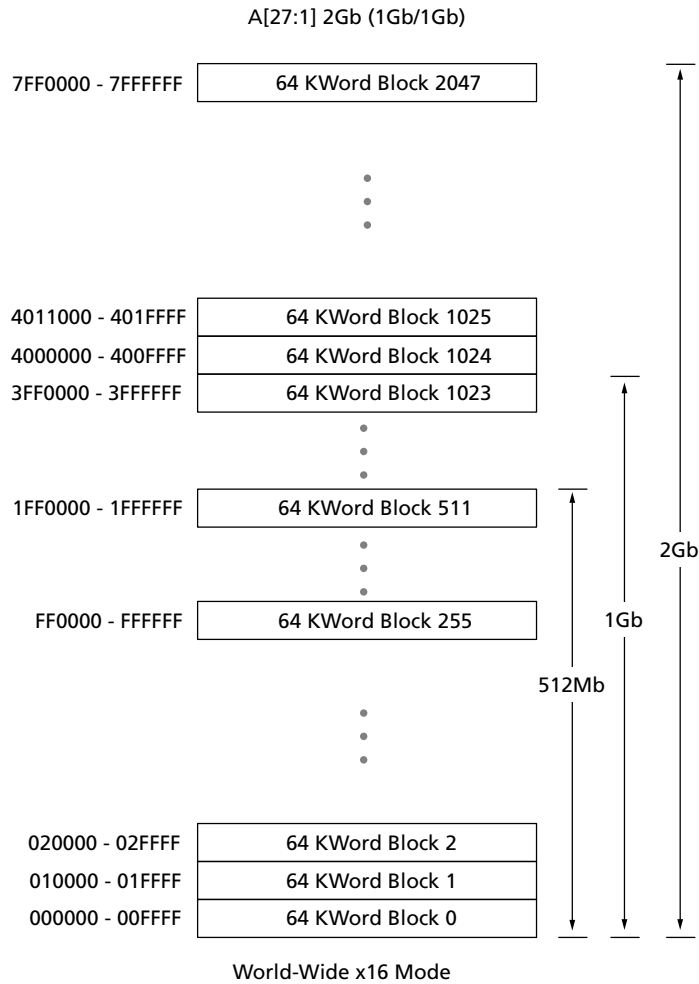


## Memory Map

**Figure 2: Memory Map – 512Mb and 1Gb**

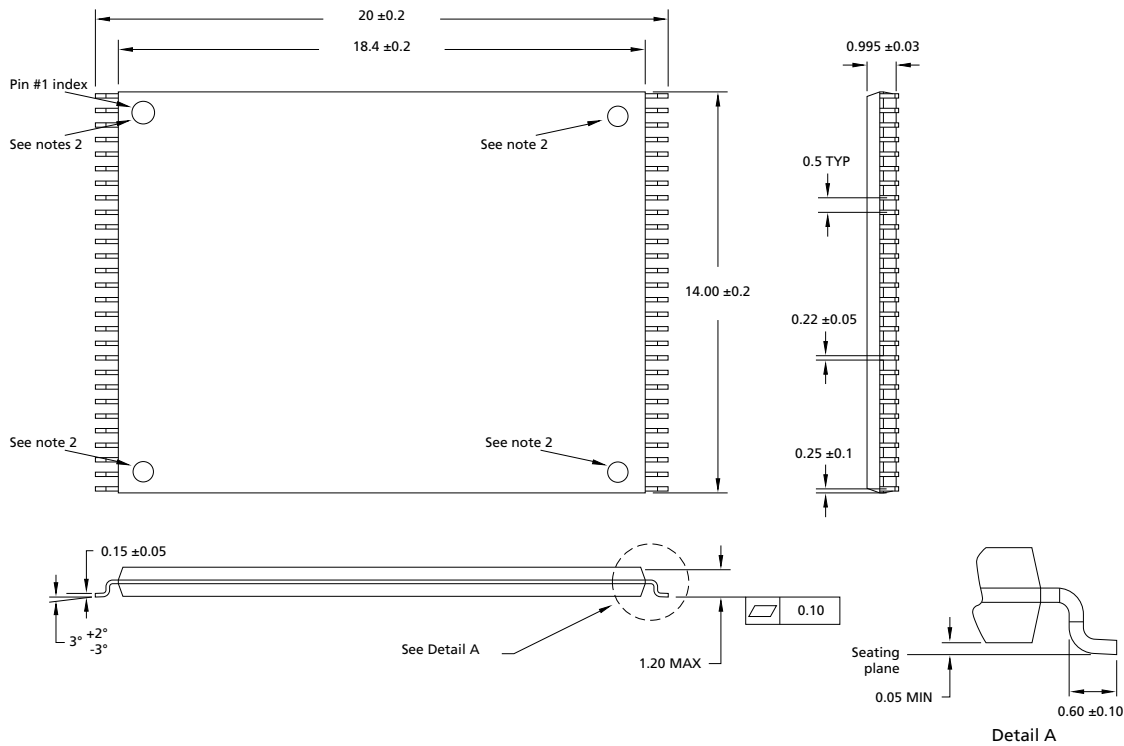


**Figure 3: Memory Map – 2Gb**



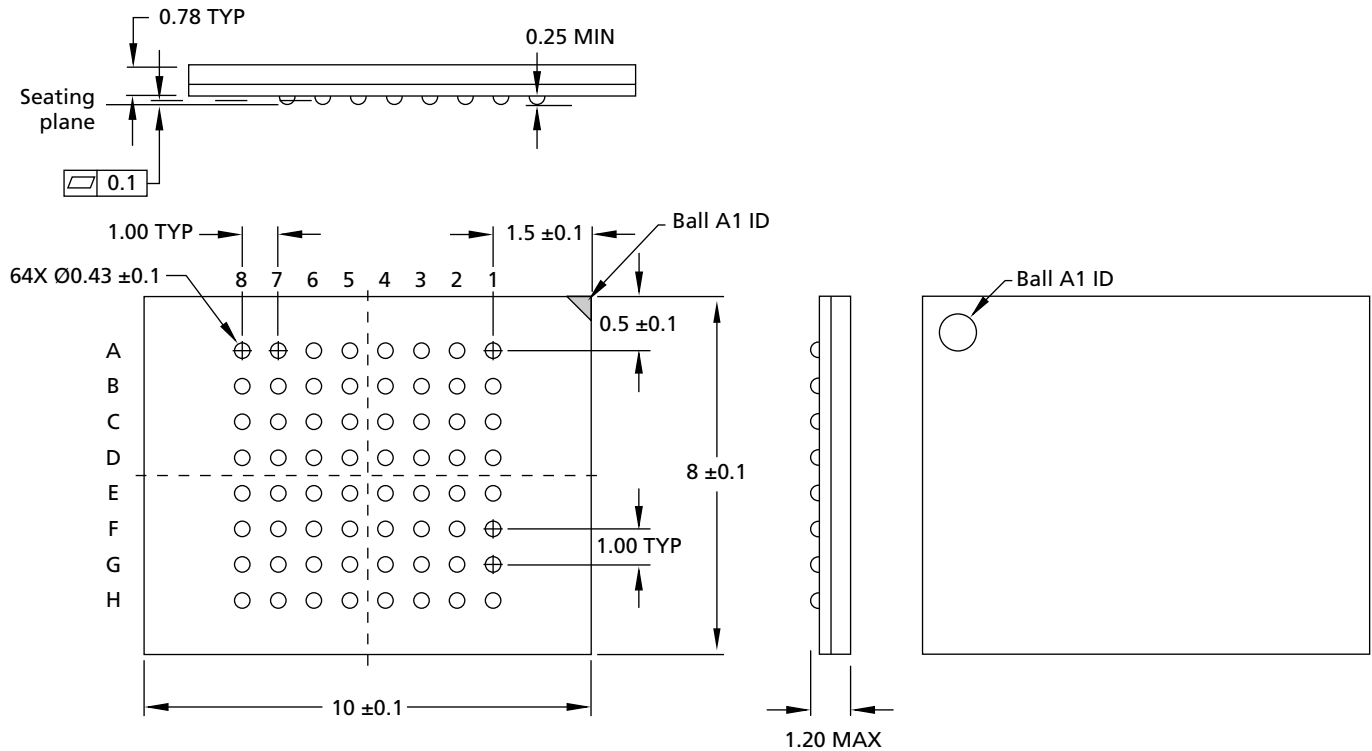
## Package Dimensions

Figure 4: 56-Pin TSOP – 14mm x 20mm



- Notes:
1. All dimensions are in millimeters. Drawing not to scale.
  2. One dimple on package denotes pin 1; if two dimples, then the larger dimple denotes pin 1. Pin 1 will always be in the upper left corner of the package, in reference to the product mark.
  3. For the lead width value of  $0.22 \pm 0.05$ , there is also a legacy value of  $0.15 \pm 0.05$ .

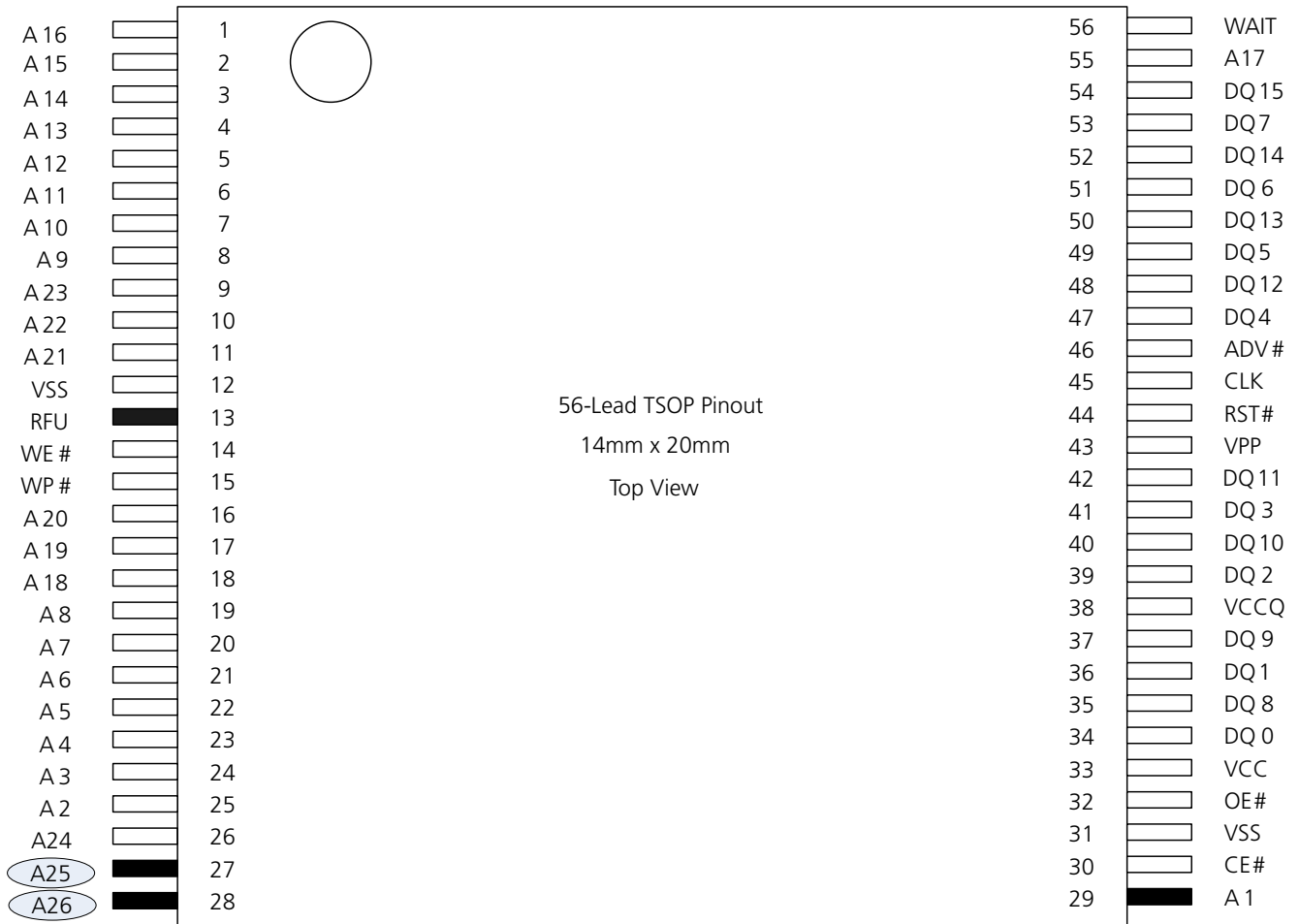
**Figure 5: 64-Ball Easy BGA – 8mm x 10mm x 1.2mm**



- Notes:
1. All dimensions are in millimeters. Drawing not to scale.
  2. The 512Mb device does not contain the A1 ID ball located on the back side of the device.

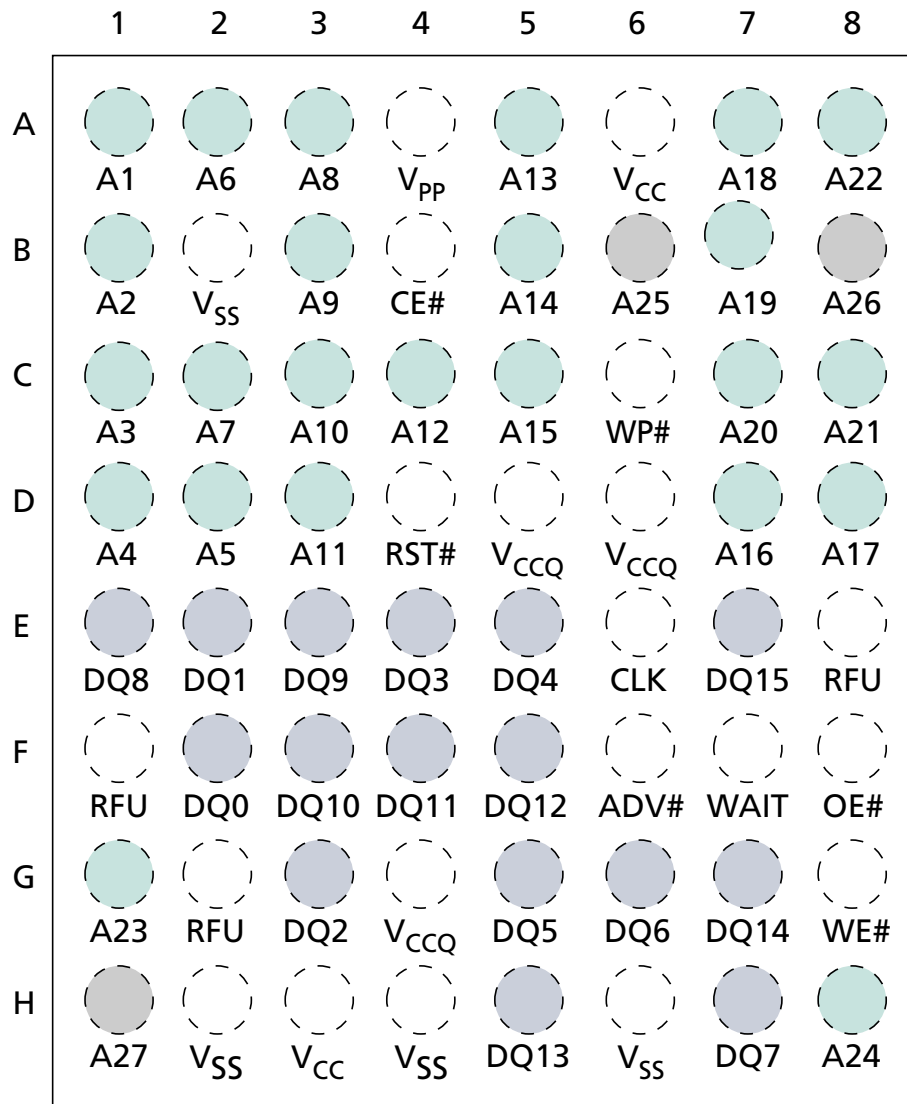
## Pinouts and Ballouts

**Figure 6: 56-Lead TSOP Pinout – 512Mb and 1Gb**



- Notes:**
1. A1 is the least significant address bit.
  2. ADV# must be tied to V<sub>SS</sub> or driven to LOW throughout the asynchronous read mode.
  3. A25 is valid for 512Mb densities and above; otherwise, it is a no connect (NC).
  4. A26 is valid for 1Gb densities and above; otherwise, it is a no connect (NC).
  5. One dimple on package denotes Pin 1 which will always be in the upper left corner of the package, in reference to the product mark.

**Figure 7: 64-Ball Easy BGA (Top View – Balls Down) – 512Mb, 1Gb, and 2Gb**



- Notes:
1. A1 is the least significant address bit.
  2. A25 is valid for 512Mb densities and above; otherwise, it is a no connect (NC).
  3. A26 is valid for 1Gb densities and above; otherwise, it is a no connect (NC).
  4. A27 is valid for 2Gb densities and above; otherwise, it is a no connect (NC).
  5. One dimple on package denotes Pin 1 which will always be in the upper left corner of the package, in reference to the product mark.

## Signal Descriptions

**Table 4: TSOP and Easy BGA Signal Descriptions**

| Symbol   | Type         | Name and Function  |
|----------|--------------|--|
| A[MAX:1] | Input        | <b>Address inputs:</b> Device address inputs.<br><b>Note:</b> Unused active address pins should not be left floating; tie them to $V_{CCQ}$ or $V_{SS}$ according to specific design requirements.   |
| ADV#     | Input        | <b>Address valid:</b> Active LOW input. During synchronous READ operations, addresses are latched on the rising edge of ADV#, or on the next valid CLK edge with ADV# LOW, whichever occurs first. In asynchronous mode, the address is latched when ADV# goes HIGH or continuously flows through if ADV# is held LOW.<br><b>Note:</b> Designs not using ADV# must tie it to $V_{SS}$ to allow addresses to flow through.  |
| CE#      | Input        | <b>Chip enable:</b> Active LOW input. CE# LOW selects the associated die. When asserted, internal control logic, input buffers, decoders, and sense amplifiers are active. When de-asserted, the associated die is deselected, power is reduced to standby levels, data and wait outputs are placed in High-Z.<br><b>Note:</b> CE# must be driven HIGH when device is not in use.  |
| CLK      | Input        | <b>Clock:</b> Synchronizes the device with the system bus frequency in synchronous-read mode. During synchronous READs, addresses are latched on the rising edge of ADV#, or on the next valid CLK edge with ADV# LOW, whichever occurs first.<br><b>Note:</b> Designs not using CLK for synchronous read mode must tie it to $V_{CCQ}$ or $V_{SS}$ .  |
| OE#      | Input        | <b>Output enable:</b> Active LOW input. OE# LOW enables the device's output data buffers during READ cycles. OE# HIGH places the data outputs and WAIT in High-Z.  |
| RST#     | Input        | <b>Reset:</b> Active LOW input. RST# resets internal automation and inhibits WRITE operations. This provides data protection during power transitions. RST# HIGH enables normal operation. Exit from reset places the device in asynchronous read array mode.  |
| WP#      | Input        | <b>Write protect:</b> Active LOW input. WP# LOW enables the lock-down mechanism. Blocks in lock-down cannot be unlocked with the Unlock command. WP# HIGH overrides the lock-down function enabling blocks to be erased or programmed using software commands.<br><b>Note:</b> Designs not using WP# for protection could tie it to $V_{CCQ}$ or $V_{SS}$ without additional capacitor.  |
| WE#      | Input        | <b>Write enable:</b> Active LOW input. WE# controls writes to the device. Address and data are latched on the rising edge of WE# or CE#, whichever occurs first.   |
| $V_{PP}$ | Power/Input  | <b>Erase and program power:</b> A valid voltage on this pin allows erasing or programming. Memory contents cannot be altered when $V_{PP} \leq V_{PPLK}$ . Block erase and program at invalid $V_{PP}$ voltages should not be attempted.<br><br>Set $V_{PP} = V_{PPL}$ for in-system PROGRAM and ERASE operations. To accommodate resistor or diode drops from the system supply, the $V_{IH}$ level of $V_{PP}$ can be as low as $V_{PPL,min}$ . $V_{PP}$ must remain above $V_{PPL,min}$ to perform in-system modification. $V_{PP}$ may be 0V during READ operations.<br><br>$V_{PP}$ can be connected to 9V for a cumulative total not to exceed 80 hours. Extended use of this pin at 9V may reduce block cycling capability. |
| DQ[15:0] | Input/Output | <b>Data input/output:</b> Inputs data and commands during WRITE cycles; outputs data during memory, status register, protection register, and read configuration register reads. Data balls float when the CE# or OE# are de-asserted. Data is internally latched during writes.   |

**Table 4: TSOP and Easy BGA Signal Descriptions (Continued)**

| Symbol    | Type   | Name and Function  |
|-----------|--------|--|
| WAIT      | Output | <p><b>Wait:</b> Indicates data valid in synchronous array or non-array burst reads. Read configuration register bit 10 (RCR.10, WT) determines its polarity when asserted. This signal's active output is <math>V_{OL}</math> or <math>V_{OH}</math> when CE# and OE# are <math>V_{IL}</math>. WAIT is High-Z if CE# or OE# is <math>V_{IH}</math>.</p> <ul style="list-style-type: none"> <li>• In synchronous array or non-array read modes, this signal indicates invalid data when asserted and valid data when de-asserted.</li> <li>• In asynchronous page mode, and all write modes, this signal is de-asserted.</li> </ul> |
| $V_{CC}$  | Power  | <p><b>Device core power supply:</b> Core (logic) source voltage. Writes to the array are inhibited when <math>V_{CC} \leq V_{LKO}</math>. Operations at invalid <math>V_{CC}</math> voltages should not be attempted.</p>  |
| $V_{CCQ}$ | Power  | <p><b>Output power supply:</b> Output-driver source voltage.</p>   |
| $V_{SS}$  | Power  | <p><b>Ground:</b> Connect to system ground. Do not float any <math>V_{SS}</math> connection.</p>   |
| RFU       | —      | <p><b>Reserved for future use:</b> Reserved by Micron for future device functionality and enhancement. These should be treated in the same way as a DU signal.</p>   |
| DU        | —      | <p><b>Do not use:</b> Do not connect to any other signal, or power supply; must be left floating.</p>  |
| NC        | —      | <p><b>No connect:</b> No internal connection; can be driven or floated.</p>  |



## Bus Operations

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

Bus cycles to/from the device conform to standard microprocessor bus operations. Bus operations and the logic levels that must be applied to the device control signal inputs are shown here.

**Table 5: Bus Operations**

| Bus Operation  |              | RST# | CLK          | ADV# | CE# | OE# | WE# | WAIT        | DQ[15:0] | Notes |
|----------------|--------------|------|--------------|------|-----|-----|-----|-------------|----------|-------|
| READ           | Asynchronous | H    | X            | L    | L   | L   | H   | De-asserted | Output   | -     |
|                | Synchronous  | H    | Run-<br>ning | L    | L   | L   | H   | Driven      | Output   | -     |
| WRITE          |              | H    | X            | L    | L   | H   | L   | High-Z      | Input    | 1     |
| OUTPUT DISABLE |              | H    | X            | X    | L   | H   | H   | High-Z      | High-Z   | 2     |
| STANDBY        |              | H    | X            | X    | H   | X   | X   | High-Z      | High-Z   | 2     |
| RESET          |              | L    | X            | X    | X   | X   | X   | High-Z      | High-Z   | 2, 3  |

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

### Read

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

### Write

To perform a WRITE operation, both CE# and WE# are asserted while RST# and OE# are de-asserted. During a WRITE operation, address and data are latched on the rising edge of WE# or CE#, whichever occurs first. The Command Bus Cycles table shows the bus cycle sequence for each of the supported device commands, while the Command Codes and Definitions table describes each command.

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

### Output Disable

When OE# is de-asserted, device outputs DQ[15:0] are disabled and placed in High-Z state, WAIT is also placed in High-Z.

### Standby

When CE# is de-asserted the device is deselected and placed in standby, substantially reducing power consumption. In standby, the data outputs are placed in High-Z, independent of the level placed on OE#. Standby current ( $I_{CCS}$ ) is the average current meas-

ured over any 5ms time interval, 5 $\mu$ s after CE# is de-asserted. During standby, average current is measured over the same time interval 5 $\mu$ s after CE# is de-asserted.

When the device is deselected (while CE# is de-asserted) during a PROGRAM or ERASE operation, it continues to consume active power until the PROGRAM or ERASE operation is completed.

## Reset

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

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

When device returns from a reset (RST# de-asserted), a minimum wait is required before the initial read access outputs valid data. Also, a minimum delay is required after a reset before a write cycle can be initiated. After this wake-up interval passes, normal operation is restored.

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

## Device Command Codes

The system CPU provides control of all in-system READ, WRITE, and ERASE operations of the device via the system bus. The device manages all block-erase and word-program algorithms.

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

**Note:** For a dual device, all setup commands should be re-issued to the device when a different die is selected.

**Table 6: Command Codes and Definitions**

| Mode  | Device Mode                                   | Code | Description  |
|-------|---|------|--|
| Read  | Read array                                    | 0xFF | Places the device in read array mode. Array data is output on DQ[15:0].  |
|       | Read status register                          | 0x70 | Places the device in read status register mode. The device enters this mode after a PROGRAM or ERASE command is issued. Status register data is output on DQ[7:0].   |
|       | Read device ID or read configuration register | 0x90 | Places device in read device identifier mode. Subsequent reads output manufacturer/device codes, configuration register data, block lock status, or protection register data on DQ[15:0].  |
|       | Read CFI                                      | 0x98 | Places the device in read CFI mode. Subsequent reads output CFI information on DQ[7:0].  |
|       | Clear status register                         | 0x50 | The device sets status register error bits. The clear status register command is used to clear the SR error bits.  |
| Write | Word program setup                            | 0x40 | First cycle of a 2-cycle programming command; prepares the CUI for a WRITE operation. On the next write cycle, the address and data are latched and the device executes the programming algorithm at the addressed location. During PROGRAM operations, the device responds only to READ STATUS REGISTER and PROGRAM SUSPEND commands. CE# or OE# must be toggled to update the status register in asynchronous read. CE# or ADV# must be toggled to update the status register data for synchronous non-array reads. The READ ARRAY command must be issued to read array data after programming has finished. |
|       | Buffered program                              | 0xE8 | This command loads a variable number of words up to the buffer size of 512 words onto the program buffer.  |
|       | Buffered program confirm                      | 0xD0 | The CONFIRM command is issued after the data streaming for writing into the buffer is completed. The device then performs the buffered program algorithm, writing the data from the buffer to the memory array.  |
|       | BEFP setup                                    | 0x80 | First cycle of a two-cycle command; initiates buffered enhanced factory program mode (BEFP). The CUI then waits for the BEFP CONFIRM command, 0xD0, that initiates the BEFP algorithm. All other commands are ignored when BEFP mode begins.   |
|       | BEFP confirm                                  | 0xD0 | If the previous command was BEFP SETUP (0x80), the CUI latches the address and data, and prepares the device for BEFP mode.  |

**Table 6: Command Codes and Definitions (Continued)**

| Mode          | Device Mode                                 | Code | Description  |
|---------------|---|------|--|
| Erase         | Block erase setup                           | 0x20 | First cycle of a two-cycle command; prepares the CUI for a BLOCK ERASE operation. The device performs the erase algorithm on the block addressed by the ERASE CONFIRM command. If the next command <i>is not</i> the ERASE CONFIRM (0xD0) command, the CUI sets status register bits SR4 and SR5, and places the device in read status register mode.  |
|               | Block erase confirm                         | 0xD0 | If the first command was BLOCK ERASE SETUP (0x20), the CUI latches the address and data, and the device erases the addressed block. During BLOCK ERASE operations, the device responds only to READ STATUS REGISTER and ERASE SUSPEND commands. CE# or OE# must be toggled to update the status register in asynchronous read. CE# or ADV# must be toggled to update the status register data for synchronous non-array reads. |
| Suspend       | Program or erase suspend                    | 0xB0 | This command issued to any device address initiates a suspend of the currently-executing program or BLOCK ERASE operation. The status register indicates successful suspend operation by setting either SR2 (program suspended) or SR6 (erase suspended), along with SR7 (ready). The device remains in the suspend mode regardless of control signal states (except for RST# asserted).                                       |
|               | Suspend resume                              | 0xD0 | This command issued to any device address resumes the suspended PROGRAM or BLOCK ERASE operation.  |
| Protection    | Block lock setup                            | 0x60 | First cycle of a two-cycle command; prepares the CUI for block lock configuration changes. If the next command is not BLOCK LOCK (0x01), BLOCK UNLOCK (0xD0), or BLOCK LOCK DOWN (0x2F), the CUI sets status register bits SR5 and SR4, indicating a command sequence error.   |
|               | Block lock                                  | 0x01 | If the previous command was BLOCK LOCK SETUP (0x60), the addressed block is locked.  |
|               | Block unlock                                | 0xD0 | If the previous command was BLOCK LOCK SETUP (0x60), the addressed block is unlocked. If the addressed block is in a lock down state, the operation has no effect.   |
|               | Block lock down                             | 0x2F | If the previous command was BLOCK LOCK SETUP (0x60), the addressed block is locked down.   |
|               | OTP register or lock register program setup | 0xC0 | First cycle of a two-cycle command; prepares the device for a OTP REGISTER or LOCK REGISTER PROGRAM operation. The second cycle latches the register address and data, and starts the programming algorithm to program data the OTP array.   |
| Configuration | Read configuration register setup           | 0x60 | First cycle of a two-cycle command; prepares the CUI for device read configuration. If the SET READ CONFIGURATION REGISTER command (0x03) is not the next command, the CUI sets status register bits SR4 and SR5, indicating a command sequence error.   |
|               | Read configuration register                 | 0x03 | If the previous command was READ CONFIGURATION REGISTER SETUP (0x60), the CUI latches the address and writes A[16:1] to the read configuration register for Easy BGA and TSOP, A[15:0] for QUAD+. Following a CONFIGURE READ CONFIGURATION REGISTER command, subsequent READ operations access array data.   |

**Table 6: Command Codes and Definitions (Continued)**

| Mode        | Device Mode                 | Code | Description  |
|-------------|-----------------------------|------|--|
| Blank Check | Block blank check           | 0xBC | First cycle of a two-cycle command; initiates the BLANK CHECK operation on a main block.   |
|             | Block blank check confirm   | 0xD0 | Second cycle of blank check command sequence; it latches the block address and executes blank check on the main array block.   |
| EFI         | Extended function interface | 0xEB | First cycle of a multiple-cycle command; initiate operation using extended function interface. The second cycle is a Sub-Op-Code, the data written on third cycle is one less than the word count; the allowable value on this cycle are 0–511. The subsequent cycles load data words into the program buffer at a specified address until word count is achieved. |

## Device Command Bus Cycles

Device operations are initiated by writing specific device commands to the command user interface (CUI). Several commands are used to modify array data including WORD PROGRAM and BLOCK ERASE commands. Writing either command to the CUI initiates a sequence of internally timed functions that culminate in the completion of the requested task. However, the operation can be aborted by either asserting RST# or by issuing an appropriate suspend command.

**Table 7: Command Bus Cycles**

| Mode          | Command   | Bus Cycles | First Bus Cycle |                   |                   | Second Bus Cycle |                   |                   |
|---------------|---|------------|-----------------|-------------------|-------------------|------------------|-------------------|-------------------|
|               |   |            | Op              | Addr <sup>1</sup> | Data <sup>2</sup> | Op               | Addr <sup>1</sup> | Data <sup>2</sup> |
| Read          | READ ARRAY  | 1          | WRITE           | DnA               | 0xFF              | –                | –                 | –                 |
|               | READ DEVICE IDENTIFIER                                | ≥2         | WRITE           | DnA               | 0x90              | READ             | DBA + IA          | ID                |
|               | READ CFI  | ≥2         | WRITE           | DnA               | 0x98              | READ             | DBA + CFI-A       | CFI-D             |
|               | READ STATUS REGISTER                                  | 2          | WRITE           | DnA               | 0x70              | READ             | DnA               | SRD               |
|               | CLEAR STATUS REGISTER                                 | 1          | WRITE           | DnA               | 0x50              | –                | –                 | –                 |
| Program       | WORD PROGRAM  | 2          | WRITE           | WA                | 0x40              | WRITE            | WA                | WD                |
|               | BUFFERED PROGRAM <sup>3</sup>                         | >2         | WRITE           | WA                | 0xE8              | WRITE            | WA                | N - 1             |
|               | BUFFERED ENHANCED FACTORY PROGRAM (BEFP) <sup>4</sup> | >2         | WRITE           | WA                | 0x80              | WRITE            | WA                | 0xD0              |
| Erase         | BLOCK ERASE   | 2          | WRITE           | BA                | 0x20              | WRITE            | BA                | 0xD0              |
| Suspend       | PROGRAM/ERASE SUSPEND                                 | 1          | WRITE           | DnA               | 0xB0              | –                | –                 | –                 |
|               | PROGRAM/ERASE RESUME                                  | 1          | WRITE           | DnA               | 0xD0              | –                | –                 | –                 |
| Protection    | BLOCK LOCK  | 2          | WRITE           | BA                | 0x60              | WRITE            | BA                | 0x01              |
|               | BLOCK UNLOCK  | 2          | WRITE           | BA                | 0x60              | WRITE            | BA                | 0xD0              |
|               | BLOCK LOCK DOWN                                       | 2          | WRITE           | BA                | 0x60              | WRITE            | BA                | 0x2F              |
|               | PROGRAM OTP REGISTER                                  | 2          | WRITE           | PRA               | 0xC0              | WRITE            | OTP-RA            | OTP-D             |
|               | PROGRAM LOCK REGISTER                                 | 2          | WRITE           | LRA               | 0xC0              | WRITE            | LRA               | LRD               |
| Configuration | CONFIGURE READ CONFIGURATION REGISTER                 | 2          | WRITE           | RCD               | 0x60              | WRITE            | RCD               | 0x03              |
| Blank Check   | BLOCK BLANK CHECK                                     | 2          | WRITE           | BA                | 0xBC              | WRITE            | BA                | D0                |
| EFI           | EXTENDED FUNCTION INTERFACE <sup>5</sup>              | >2         | WRITE           | WA                | 0xEB              | Write            | WA                | Sub-Op code       |

- Notes:
1. First command cycle address should be the same as the operation's target address. DBA = Device base address (needed for dual die 512Mb device); DnA = Address within the device; IA = Identification code address offset; CFI-A = Read CFI address offset; WA = Word address of memory location to be written; BA = Address within the block; OTP-RA = Protection register address; LRA = Lock register address; RCD = Read configuration register data on A[16:1] for Easy BGA and TSOP, A[15:0] for QUAD+ package.
  2. ID = Identifier data; CFI-D = CFI data on DQ[15:0]; SRD = Status register data; WD = Word data; N = Word count of data to be loaded into the write buffer; OTP-D = Protection register data; LRD = Lock register data.

3. The second cycle of the BUFFERED PROGRAM command is the word count of the data to be loaded into the write buffer. This is followed by up to 512 words of data. Then the CONFIRM command (0xD0) is issued, triggering the array programming operation.
4. The CONFIRM command (0xD0) is followed by the buffer data.
5. The second cycle is a Sub-Op-Code, the data written on third cycle is N-1;  $1 \leq N \leq 512$ . The subsequent cycles load data words into the program buffer at a specified address until word count is achieved, after the data words are loaded, the final cycle is the confirm cycle (0xD0).

## Read Operations

The device supports two read modes: asynchronous page mode and synchronous burst mode. Asynchronous page mode is the default read mode after device power-up or a reset. Under asynchronous page mode, the device can also perform single word read. The read configuration register must be configured to enable synchronous burst reads of the array.

The device can be in any of four read states: read array, read identifier, read status, or read CFI. Upon power-up, or after a reset, the device defaults to read array. To change the read state, the appropriate READ command must be written to the device.

### Asynchronous Single Word Read

To perform an asynchronous single word read, an address is driven onto the address bus, and CE# is asserted.

**Note:** To perform an asynchronous single word read for a TSOP package, ADV# must be LOW throughout the READ cycle. For an Easy BGA package, ADV# can be driven HIGH to latch the address or be held LOW throughout the READ cycle.

WE# and RST# must already have been de-asserted. WAIT is set to a de-asserted state during single word mode, as determined by bit 10 of the read configuration register. CLK is not used for asynchronous single word reads, and is ignored. If asynchronous reads are to be performed only, CLK should be tied to a valid  $V_{IH}$  or  $V_{SS}$  level, WAIT can be floated, and ADV# must be tied to ground. After OE# is asserted, the data is driven onto DQ[15:0] after an initial access time  $t_{AVQV}$  or  $t_{GLQV}$  delay.

### Asynchronous Page Mode Read (Easy BGA Only)

**Note:** Asynchronous Page Mode Read is supported only in the main array.

Following a device power-up or reset, asynchronous page mode is the default read mode and the device is set to read array. However, to perform array reads after any other device operation (WRITE operation), the READ ARRAY command must be issued in order to read from the array.

Asynchronous page mode reads can only be performed when read configuration register bit RCR15 is set.

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

In asynchronous page mode, 16 data words are “sensed” simultaneously from the array and loaded into an internal page buffer. The buffer word corresponding to the initial address on the address bus is driven onto DQ[15:0] after the initial access delay. The lowest four address bits determine which word of the 16-word page is output from the data buffer at any given time.



## Synchronous Burst Mode Read (Easy BGA Only)

Read configuration register bits RCR[15:0] must be set before synchronous burst operation can be performed. Synchronous burst mode can be performed for both array and non-array reads such as read ID, read status, or read query.

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

During synchronous array and non-array read modes, the first word is output from the data buffer on the next valid CLK edge after the initial access latency delay. Subsequent data is output on valid CLK edges following a minimum delay. However, for a synchronous non-array read, the same word of data will be output on successive clock edges until the burst length requirements are satisfied. Refer to the timing diagrams for more detailed information.

## Read CFI

The READ CFI command instructs the device to output CFI data when read. See Common Flash Interface for details on issuing the READ CFI command, and for details on addresses and offsets within the CFI database.

## Read Device ID

The READ DEVICE IDENTIFIER command instructs the device to output manufacturer code, device identifier code, block lock status, protection register data, or configuration register data.

**Table 8: Device ID Information**

| Item  | Address                    | Data  |
|---|----------------------------|---|
| Manufacturer code   | 0x00                       | 0x89  |
| Device ID code  | 0x01                       | ID (see the Device ID Codes table )   |
| <b>Block lock configuration</b><br>Block is unlocked<br>Block is locked<br>Block is not locked down<br>Block is locked down | Block base address + 0x02  | <b>Lock bit</b><br>DQ <sub>0</sub> = 0b0<br>DQ <sub>0</sub> = 0b1<br>DQ <sub>1</sub> = 0b0<br>DQ <sub>1</sub> = 0b1 |
| Read configuration register   | 0x05                       | RCR contents  |
| General purpose register  | Device base address + 0x07 | General purpose register data   |
| Lock register 0   | 0x80                       | PR-LK0 data   |
| 64-bit factory-programmed OTP register  | 0x81–0x84                  | Factory OTP register data   |
| 64-bit user-programmable OTP register   | 0x85–0x88                  | User OTP register data  |
| Lock register 1   | 0x89                       | PR-LK1 OTP register lock data   |
| 128-bit user-programmable protection registers  | 0x8A–0x109                 | OTP register data   |



## Device ID Codes

Table 9: Device ID codes

| ID Code Type | Device Density | Device Identifier Codes |                       |                           |
|--------------|----------------|-------------------------|-----------------------|---------------------------|
|              |                | -T (Top Parameter)      | -B (Bottom Parameter) | -E/F (Symmetrical Blocks) |
| Device Code  | 512Mb          | 8964                    | 8965                  | 899E                      |
|              | 1Gb            | 8966                    | 8967                  | 899F                      |

Note: 1. The 2Gb devices do not have a unique device ID associated with them. Each die within the stack can be identified by device ID codes.

## Program Operations

Successful programming requires the addressed block to be unlocked. If the block is locked down, WP# must be de-asserted and the block must be unlocked before attempting to program the block. Attempting to program a locked block causes a program error (SR4 and SR1 set) and termination of the operation. See Security Modes for details on locking and unlocking blocks.

### Word Programming (40h)

Word programming operations are initiated by writing the WORD PROGRAM SETUP command to the device (see the Command Codes and Definitions table). This is followed by a second write to the device with the address and data to be programmed. The device outputs status register data when read (see the Word Program Flowchart).  $V_{PP}$  must be above  $V_{PPLK}$ , and within the specified  $V_{PPL}$  MIN/MAX values.

During programming, the device executes a sequence of internally-timed events that program the desired data bits at the addressed location, and verifies that the bits are sufficiently programmed. Programming the array changes 1s to 0s. Memory array bits that are 0s can be changed to 1s only by erasing the block (see Erase Operations).

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

SR7 indicates the programming status while the sequence executes. Commands that can be issued to the device during programming are PROGRAM SUSPEND, READ STATUS REGISTER, READ DEVICE IDENTIFIER, READ CFI, and READ ARRAY (this returns unknown data).

When programming has finished, SR4 (when set) indicates a programming failure. If SR3 is set, the device could not perform the WORD PROGRAMMING operation because  $V_{PP}$  was outside of its acceptable limits. If SR1 is set, the WORD PROGRAMMING operation attempted to program a locked block, causing the operation to abort.

Before issuing a new command, the status register contents should be examined and then cleared using the CLEAR STATUS REGISTER command. Any valid command can follow, when word programming has completed.

### Buffered Programming (E8h, D0h)

The device features a 512-word buffer to enable optimum programming performance. For buffered programming, data is first written to an on-chip write buffer. Then the buffer data is programmed into the array in buffer-size increments. This can improve system programming performance significantly over non-buffered programming.

When the BUFFERED PROGRAMMING SETUP command is issued, status register information is updated and reflects the availability of the buffer. SR7 indicates buffer availability: if set, the buffer is available; if cleared, the buffer is not available.

**Note:** The device default state is to output SR data after the BUFFERED PROGRAMMING SETUP command. CE# and OE# LOW drive device to update status register. It is not allowed to issue 70h to read SR data after E8h command; otherwise, 70h would be counted as word count.

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

On the next write, a device start address is given along with the first data to be written to the flash memory array. Subsequent writes provide additional device addresses and data. All data addresses must lie within the start address plus the word count. Optimum programming performance and lower power usage are obtained by aligning the starting address at the beginning of a 512-word boundary ( $A[9:1] = 0x00$  for Easy BGA and TSOP,  $A[8:0]$  for QUAD+ package; see Part Numbering Information). The maximum buffer size would be 256-word if the misaligned address range is crossing a 512-word boundary during programming.

After the last data is written to the buffer, the BUFFERED PROGRAMMING CONFIRM command must be issued to the original block address. The device begins to program buffer contents to the array. If a command other than the BUFFERED PROGRAMMING CONFIRM command is written to the device, a command sequence error occurs and  $SR[7,5,4]$  are set. If an error occurs while writing to the array, the device stops programming, and  $SR[7,4]$  are set, indicating a programming failure.

When buffered programming has completed, additional buffer writes can be initiated by issuing another BUFFERED PROGRAMMING SETUP command and repeating the buffered program sequence. Buffered programming may be performed with  $V_{PP} = V_{PPL}$  or  $V_{PPH}$  (see Operating Conditions for limitations when operating the device with  $V_{PP} = V_{PPH}$ ).

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

If buffered programming is attempted while  $V_{PP}$  is at or below  $V_{PPLK}$ ,  $SR[4,3]$  are set. If any errors are detected that have set status register bits, the status register should be cleared using the CLEAR STATUS REGISTER command.

## Buffered Enhanced Factory Programming (80h, D0h)

Buffered enhanced factory programming (BEFP) speeds up multilevel cell (MLC) programming. The enhanced programming algorithm used in BEFP eliminates traditional programming elements that drive up overhead in device programmer systems.

BEFP consists of three phases: setup, program/verify, and exit (see the BEFP Flowchart). It uses a write buffer to spread MLC program performance across 512 data words. Verification occurs in the same phase as programming to accurately program the cell to the correct bit state.

A single two-cycle command sequence programs the entire block of data. This enhancement eliminates three write cycles per buffer: two commands and the word count for each set of 512 data words. Host programmer bus cycles fill the device write buffer followed by a status check.  $SR0$  indicates when data from the buffer has been programmed into sequential array locations.

Following the buffer-to-flash array programming sequence, the device increments internal addressing to automatically select the next 512-word array boundary. This aspect of BEFP saves host programming equipment the address bus setup overhead.

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

**Table 10: BEFP Requirements**

| Parameter/Issue   | Requirement  | Notes |
|-------------------|--|-------|
| Case temperature  | $T_C = 30^\circ\text{C} \pm 10^\circ\text{C}$  |       |
| $V_{CC}$          | Nominal $V_{CC}$   |       |
| $V_{PP}$          | Driven to $V_{PPH}$  |       |
| Setup and confirm | Target block must be unlocked before issuing the BEFP Setup and Confirm commands.  |       |
| Programming       | The first-word address (WA0) of the block to be programmed must be held constant from the setup phase through all data streaming into the target block, until transition to the exit phase is desired. |       |
| Buffer alignment  | WA0 must align with the start of an array buffer boundary.   | 1     |

Note: 1. Word buffer boundaries in the array are determined by the lowest 9 address bits (0x000 through 0x1FF). The alignment start point is 0x000.

**Table 11: BEFP Considerations**

| Parameter/Issue       | Requirement   | Notes |
|-----------------------|---|-------|
| Cycling               | For optimum performance, cycling must be limited below 50 ERASE cycles per block.   | 1     |
| Programming blocks    | BEFP programs one block at a time; all buffer data must fall within a single block. | 2     |
| Suspend               | BEFP cannot be suspended.   |       |
| Programming the array | Programming to the array can occur only when the buffer is full.                    | 3     |

- Notes:
1. Some degradation in performance may occur if this limit is exceeded, but the internal algorithm continues to work properly.
  2. If the internal address counter increments beyond the block's maximum address, addressing wraps around to the beginning of the block.
  3. If the number of words is less than 512, remaining locations must be filled with 0xFFFF.

**BEFP Setup Phase:** After receiving the BEFP SETUP and CONFIRM command sequence, SR7 (ready) is cleared, indicating that the device is busy with BEFP algorithm startup. A delay before checking SR7 is required to allow the device enough time to perform all of its setups and checks (block lock status,  $V_{PP}$  level, etc.). If an error is detected, SR4 is set and BEFP operation terminates. If the block was found to be locked, SR1 is also set. SR3 is set if the error occurred due to an incorrect  $V_{PP}$  level.

**Note:** Reading from the device after the BEFP SETUP and CONFIRM command sequence outputs status register data. Do not issue the READ STATUS REGISTER command; it will be interpreted as data to be loaded into the buffer.

**BEFP Program/Verify Phase:** After the BEFP setup phase has completed, the host programming system must check SR[7,0] to determine the availability of the write buffer for data streaming. SR7 cleared indicates the device is busy and the BEFP program/verify phase is activated. SR0 indicates the write buffer is available.

Two basic sequences repeat in this phase: loading of the write buffer, followed by buffer data programming to the array. For BEFP, the count value for buffer loading is always

the maximum buffer size of 512 words. During the buffer-loading sequence, data is stored to sequential buffer locations starting at address 0x00. Programming of the buffer contents to the array starts as soon as the buffer is full. If the number of words is less than 512, the remaining buffer locations must be filled with 0xFFFF.

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

The starting address for data entry must be buffer size aligned; if not, the BEFP algorithm will be aborted, the program fails, and the (SR4) flag will be set.

Data words from the write buffer are directed to sequential memory locations in the array; programming continues from where the previous buffer sequence ended. The host programming system must poll SR0 to determine when the buffer program sequence completes. SR0 cleared indicates that all buffer data has been transferred to the array; SR0 set indicates that the buffer is not available yet for the next fill cycle. The host system may check full status for errors at any time, but it is only necessary on a block basis after BEFP exit. After the buffer fill cycle, no WRITE cycles should be issued to the device until SR0 = 0 and the device is ready for the next buffer fill.

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

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

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

## Program Suspend

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

When a programming operation is executing, issuing the PROGRAM SUSPEND command requests the device to suspend the programming algorithm at predetermined points. The device continues to output status register data after the PROGRAM SUSPEND command is issued. Programming is suspended when SR[7,2] are set.

To read data from the device, the READ ARRAY command must be issued. READ ARRAY, READ STATUS REGISTER, READ DEVICE IDENTIFIER, READ CFI, and PROGRAM RESUME valid commands during a program suspend.

During a program suspend, de-asserting CE# places the device in standby, reducing active current. V<sub>PP</sub> must remain at its programming level, and WP# must remain unchanged while in program suspend. If RST# is asserted, the device is reset.

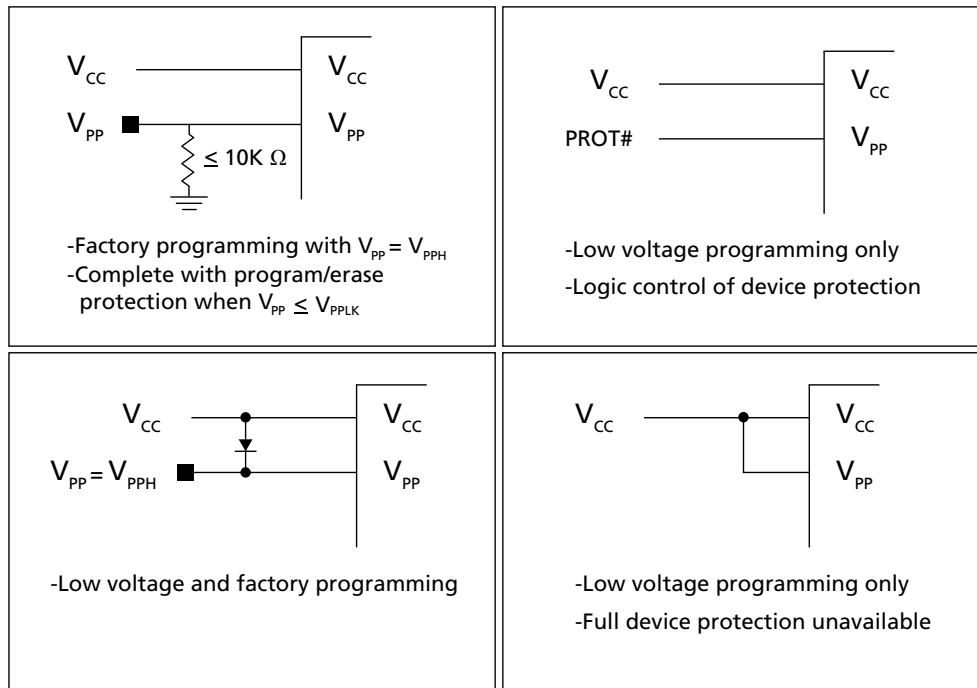
### Program Resume

The RESUME command instructs the device to continue programming, and automatically clears SR[7,2]. This command can be written to any address. If error bits are set, the status register should be cleared before issuing the next command. RST# must remain de-asserted.

### Program Protection

When  $V_{PP} = V_{IL}$ , absolute hardware write protection is provided for all device blocks. If  $V_{PP}$  is at or below  $V_{PPLK}$ , programming operations halt and SR3 is set, indicating a  $V_{PP}$ -level error. Block lock registers are not affected by the voltage level on  $V_{PP}$ ; they may still be programmed and read, even if  $V_{PP}$  is less than  $V_{PPLK}$ .

**Figure 8: Example  $V_{PP}$  Supply Connections**



## Erase Operations

### BLOCK ERASE Command

ERASE operations are performed on a block basis. An entire block is erased each time a BLOCK ERASE command sequence is issued, and only one block is erased at a time. When a block is erased, each bit within that block reads as a logical 1.

A BLOCK ERASE operation is initiated by writing the BLOCK ERASE SETUP command to the address of the block to be erased, followed by the BLOCK ERASE CONFIRM command. If the device is placed in standby (CE# de-asserted) during a BLOCK ERASE operation, the device completes the operation before entering standby. The  $V_{PP}$  value must be above  $V_{PPLK}$  and the block must be unlocked.

During a BLOCK ERASE operation, the device executes a sequence of internally-timed events that conditions, erases, and verifies all bits within the block. Erasing the array changes the value in each cell from a 1 to a 0. Memory block array cells that with a value of 1 can be changed to 0 only by programming the block.

The status register can be examined for block erase progress and errors by reading any address. The device remains in the read status register state until another command is written. SR0 indicates whether the addressed block is erasing. SR7 is set upon erase completion.

SR7 indicates block erase status while the sequence executes. When the BLOCK ERASE operation has completed, SR5 = 1 (set) indicates an erase failure. SR3 = 1 indicates that the device could not perform the BLOCK ERASE operation because  $V_{PP}$  was outside of its acceptable limits. SR1 = 1 indicates that the BLOCK ERASE operation attempted to erase a locked block, causing the operation to abort.

Before issuing a new command, the status register contents should be examined and then cleared using the CLEAR STATUS REGISTER command. Any valid command can follow after the BLOCK ERASE operation has completed.

The BLOCK ERASE operation is aborted by performing a reset or powering down the device. In either case, data integrity cannot be ensured, and it is recommended to erase again the blocks aborted.

### BLANK CHECK Command

The BLANK CHECK operation determines whether a specified main block is blank; that is, completely erased. Other than a BLANK CHECK operation, only a BLOCK ERASE operation can ensure a block is completely erased. BLANK CHECK is especially useful when a BLOCK ERASE operation is interrupted by a power loss event.

A BLANK CHECK operation can apply to only one block at a time. The only operation allowed simultaneously is a READ STATUS REGISTER operation. SUSPEND and RESUME operations and a BLANK CHECK operation are mutually exclusive.

A BLANK CHECK operation is initiated by writing the BLANK CHECK SETUP command to the block address, followed by the CHECK CONFIRM command. When a successful command sequence is entered, the device automatically enters the read status state. The device then reads the entire specified block and determines whether any bit in the block is programmed or over-erased.



BLANK CHECK operation progress and errors are determined by reading the status register at any address within the block being accessed. SR7 = 0 is a BLANK CHECK busy status. SR7 = 1 is a BLANK CHECK operation complete status. The status register should be checked for any errors and then cleared. If the BLANK CHECK operation fails, meaning the block is not completely erased, SR5 = 1. CE# or OE# toggle (during polling) updates the status register.

The READ STATUS REGISTER command must always be followed by a CLEAR STATUS REGISTER command. The device remains in status register mode until another command is written to the device. Any command can follow once the BLANK CHECK command is complete.

## **ERASE SUSPEND Command**

The ERASE SUSPEND command suspends a BLOCK ERASE operation that is in progress, enabling access to data in memory locations other than the one being erased. The ERASE SUSPEND command can be issued to any device address. A BLOCK ERASE operation can be suspended to perform a WORD or BUFFER PROGRAM operation, or a READ operation within any block except the block that is erase suspended.

When a BLOCK ERASE operation is executing, issuing the ERASE SUSPEND command requests the device to suspend the erase algorithm at predetermined points. The device continues to output status register data after the ERASE SUSPEND command is issued. Block erase is suspended when SR[7,6] are set.

To read data from the device (other than an erase-suspended block), the READ ARRAY command must be issued. During erase suspend, a PROGRAM command can be issued to any block other than the erase-suspended block. Block erase cannot resume until program operations initiated during erase suspend complete. READ ARRAY, READ STATUS REGISTER, READ DEVICE IDENTIFIER, READ CFI, and ERASE RESUME are valid commands during erase suspend. Additionally, CLEAR STATUS REGISTER, PROGRAM, PROGRAM SUSPEND, BLOCK LOCK, BLOCK UNLOCK, and BLOCK LOCK DOWN are valid commands during an ERASE SUSPEND operation.

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

## **ERASE RESUME Command**

The ERASE RESUME command instructs the device to continue erasing, and automatically clears SR[7,6]. This command can be written to any address. If status register error bits are set, the status register should be cleared before issuing the next instruction. RST# must remain de-asserted.

## **Erase Protection**

When  $V_{PP} = V_{IL}$ , absolute hardware erase protection is provided for all device blocks. If  $V_{PP}$  is at or below  $V_{PPLK}$ , ERASE operations halt and SR3 is set indicating a  $V_{PP}$ -level error.

## Security Operations

### Block Locking

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

Software-controlled security is implemented using the BLOCK LOCK and BLOCK UNLOCK commands. Hardware-controlled security can be implemented using the BLOCK LOCK DOWN command along with asserting WP#. Also,  $V_{PP}$  data security can be used to inhibit PROGRAM and ERASE operations.

### BLOCK LOCK Command

To lock a block, issue the BLOCK LOCK SETUP command, followed by the BLOCK LOCK command issued to the desired block's address. If the SET READ CONFIGURATION REGISTER command is issued after the BLOCK LOCK SETUP command, the device configures the RCR instead.

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

### BLOCK UNLOCK Command

The BLOCK UNLOCK command is used to unlock blocks. Unlocked blocks can be read, programmed, and erased. Unlocked blocks return to a locked state when the device is reset or powered down. If a block is in a lock-down state, WP# must be de-asserted before it can be unlocked.

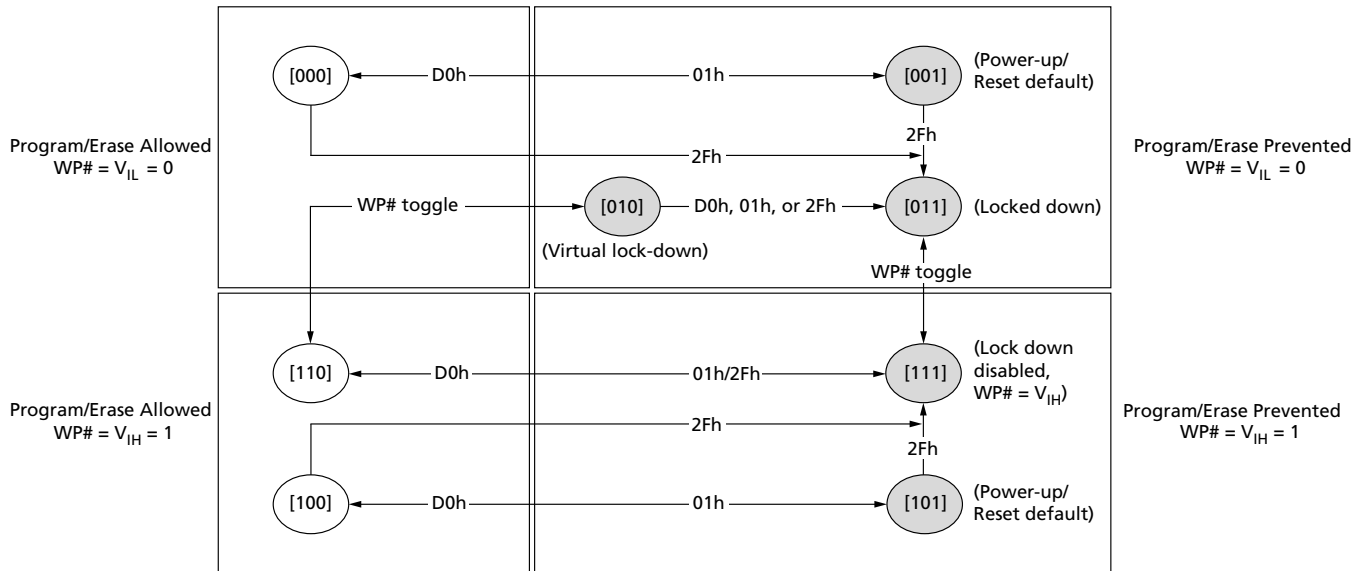
### BLOCK LOCK DOWN Command

A locked or unlocked block can be locked-down by writing the BLOCK LOCK DOWN command sequence. Blocks in a lock-down state cannot be programmed or erased; they can only be read. However, unlike locked blocks, their locked state cannot be changed by software commands alone. A locked-down block can only be unlocked by issuing the BLOCK UNLOCK command with WP# de-asserted. To return an unlocked block to locked-down state, a BLOCK LOCK DOWN command must be issued prior to changing WP# to  $V_{IL}$ . Locked-down blocks revert to the locked state upon reset or power up the device.

### Block Lock Status

The READ DEVICE IDENTIFIER command is used to determine a block's lock status. DQ[1:0] display the addressed block's lock status; DQ0 is the addressed block's lock bit, while DQ1 is the addressed block's lock-down bit.

Figure 9: Block Locking State Diagram



Note: 1. D0h = UNLOCK command; 01h = LOCK command; 60h (not shown) LOCK SETUP command; 2Fh = LOCK DOWN command.

### Block Locking During Suspend

Block lock and unlock changes can be performed during an erase suspend. To change block locking during an ERASE operation, first issue the ERASE SUSPEND command. Monitor the status register until SR7 and SR6 are set, indicating the device is suspended and ready to accept another command.

Next, write the desired lock command sequence to a block, which changes the lock state of that block. After completing BLOCK LOCK or BLOCK UNLOCK operations, resume the ERASE operation using the ERASE RESUME command.

**Note:**

A BLOCK LOCK SETUP command followed by any command other than BLOCK LOCK, BLOCK UNLOCK, or BLOCK LOCK DOWN produces a command sequence error and set SR4 and SR5. If a command sequence error occurs during an erase suspend, SR4 and SR5 remains set, even after the erase operation is resumed. Unless the Status Register is cleared using the CLEAR STATUS REGISTER command before resuming the ERASE operation, possible erase errors may be masked by the command sequence error.

If a block is locked or locked-down during an erase suspend of the *same* block, the lock status bits change immediately. However, the ERASE operation completes when it is resumed. BLOCK LOCK operations cannot occur during a program suspend.

## Selectable OTP Blocks

The OTP security feature on the device is backward-compatible to the earlier generation devices. Contact your local Micron representative for details about its implementation.

## Password Access

The password access is a security enhancement offered on the device. This feature protects information stored in array blocks by preventing content alteration or reads until a valid 64-bit password is received. The password access may be combined with nonvolatile protection and/or volatile protection to create a multi-tiered solution.

Contact your Micron sales office for further details concerning password access.

## Status Register

### Read Status Register

To read the status register, issue the READ STATUS REGISTER command at any address. Status register information is available at the address that the READ STATUS REGISTER, WORD PROGRAM, or BLOCK ERASE command is issued to. Status register data is automatically made available following a word program, block erase, or block lock command sequence. Reads from the device after any of these command sequences will output the devices status until another valid command is written (e.g. READ ARRAY command).

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

The device write status bit (SR7) provides the overall status of the device. SR[6:1] present status and error information about the PROGRAM, ERASE, SUSPEND, V<sub>PP</sub>, and BLOCK LOCK operations.

**Note:** Reading the status register is a nonarray READ operation. When the operation occurs in asynchronous page mode, only the first data is valid and all subsequent data are undefined. When the operation occurs in synchronous burst mode, the same data word requested will be output on successive clock edges until the burst length requirements are satisfied.

**Table 12: Status Register Description**

Notes apply to entire table

| Bits | Name   | Bit Settings   | Description   |
|------|--|--|---|
| 7    | Device write status (DWS)                            | 0 = Busy<br>1 = Ready  | <b>Status bit:</b> Indicates whether a PROGRAM or ERASE command cycle is in progress.   |
| 6    | Erase Suspend Status (ESS)                           | 0 = Not in effect<br>1 = In effect   | <b>Status bit:</b> Indicates whether an ERASE operation has been or is going to be suspended.   |
| 5:4  | Erase/Blank check status (ES)<br>Program status (PS) | 00 = PROGRAM/ERASE successful<br>01 = PROGRAM error<br>10 = ERASE/BLANK CHECK error<br>11 = Command sequence error | <b>Status/Error bit:</b> Indicates whether an ERASE/BLANK CHECK or PROGRAM operation was successful. When an error is returned, the operation is aborted. |
| 3    | V <sub>PP</sub> status (VPPS)                        | 0 = Within limits<br>1 = Exceeded limits (V <sub>PP</sub> ≤ V <sub>PPLK</sub> )                                    | <b>Status bit:</b> Indicates whether a PROGRAM/ERASE operation is within acceptable voltage range limits.   |
| 2    | Program suspend status (PSS)                         | 0 = Not in effect<br>1 = In effect   | <b>Status bit:</b> Indicates whether a PROGRAM operation has been or is going to be suspended.  |
| 1    | Block lock status (BLS)                              | 0 = Not locked<br>1 = Locked (operation aborted)   | <b>Status bit:</b> Indicates whether a block is locked when a PROGRAM or ERASE operation is initiated.  |
| 0    | BEFP status (BWS)                                    | 0 = BEFP complete<br>1 = BEFP in progress  | <b>Status bit:</b> Indicates whether BEFP data has completed loading into the buffer.   |

- Notes:
1. Default value = 0x80.
  2. Always clear the status register prior to resuming ERASE operations. This eliminates status register ambiguity when issuing commands during ERASE SUSPEND. If a command

sequence error occurs during an ERASE SUSPEND, the status register contains the command sequence error status (SR[7,5,4] set). When the ERASE operation resumes and finishes, possible errors during the operation cannot be detected via the status register because it contains the previous error status.

3. When bits 5:4 indicate a PROGRAM/ERASE operation error, either a CLEAR STATUS REGISTER (50h) or a RESET command must be issued with a 15 $\mu$ s delay.

## Clear Status Register

The CLEAR STATUS REGISTER command clears the status register. It functions independently of  $V_{pp}$ . The device sets and clears SR[7,6,2], but it sets bits SR[5:3,1] without clearing them. The status register should be cleared before starting a command sequence to avoid any ambiguity. A device reset also clears the status register.

## Configuration Register

### Read Configuration Register

The read configuration register (RCR) is a 16-bit read/write register used to select bus read mode (synchronous or asynchronous) and to configure device synchronous burst read characteristics. To modify RCR settings, use the CONFIGURE READ CONFIGURATION REGISTER command. RCR contents can be examined using the READ DEVICE IDENTIFIER command and then reading from offset 0x05. On power-up or exit from reset, the RCR defaults to asynchronous mode. RCR bits are described in more detail below.

**Note:** Reading the configuration register is a nonarray READ operation. When the operation occurs in asynchronous page mode, only the first data is valid, and all subsequent data are undefined. When the operation occurs in synchronous burst mode, the same word of data requested will be output on successive clock edges until the burst length requirements are satisfied.

**Table 13: Read Configuration Register**

| Bits  | Name                    | Settings/Description  |  |  |
|-------|-------------------------|---|--|--|
| 15    | Read mode (RM)          | 0 = Synchronous burst mode read<br>1 = Asynchronous page mode read (default)  |  |  |
| 14:11 | Latency count (LC[3:0]) | 0000 = Code 0 (reserved)<br>0001 = Code 1 (reserved)<br>0010 = Code 2 (reserved)<br>0011 = Code 3<br>0100 = Code 4<br>0101 = Code 5       | 0110 = Code 6<br>0111 = Code 7<br>1000 = Code 8<br>1001 = Code 9<br>1010 = Code 10           | 1011 = Code 11<br>1100 = Code 12<br>1101 = Code 13<br>1110 = Code 14<br>1111 = Code 15 (default) |
| 10    | WAIT polarity (WP)      | 0 = WAIT signal is active LOW (default)<br>1 = WAIT signal is active HIGH   |  |  |
| 9     | Reserved (R)            | Default 0, Nonchangeable  |  |  |
| 8     | WAIT delay (WD)         | 0 = WAIT de-asserted with valid data<br>1 = WAIT de-asserted one data cycle before valid data (default)                                   |  |  |
| 7     | Burst sequence (BS)     | Default 0, Nonchangeable  |  |  |
| 6     | Clock edge (CE)         | 0 = Falling edge<br>1 = Rising edge (default)   |  |  |
| 5:4   | Reserved (R)            | Default 0, Nonchangeable  |  |  |
| 3     | Burst wrap (BW)         | 0 = Wrap; Burst accesses wrap within burst length set by BL[2:0]<br>1 = No Wrap; Burst accesses do not wrap within burst length (default) |  |  |
| 2:0   | Burst length (BL[2:0])  | 001 = 4-word burst<br>010 = 8-word burst  | 011 = 16-word burst<br>111 = Continuous burst (default)<br>(Other bit settings are reserved) |  |

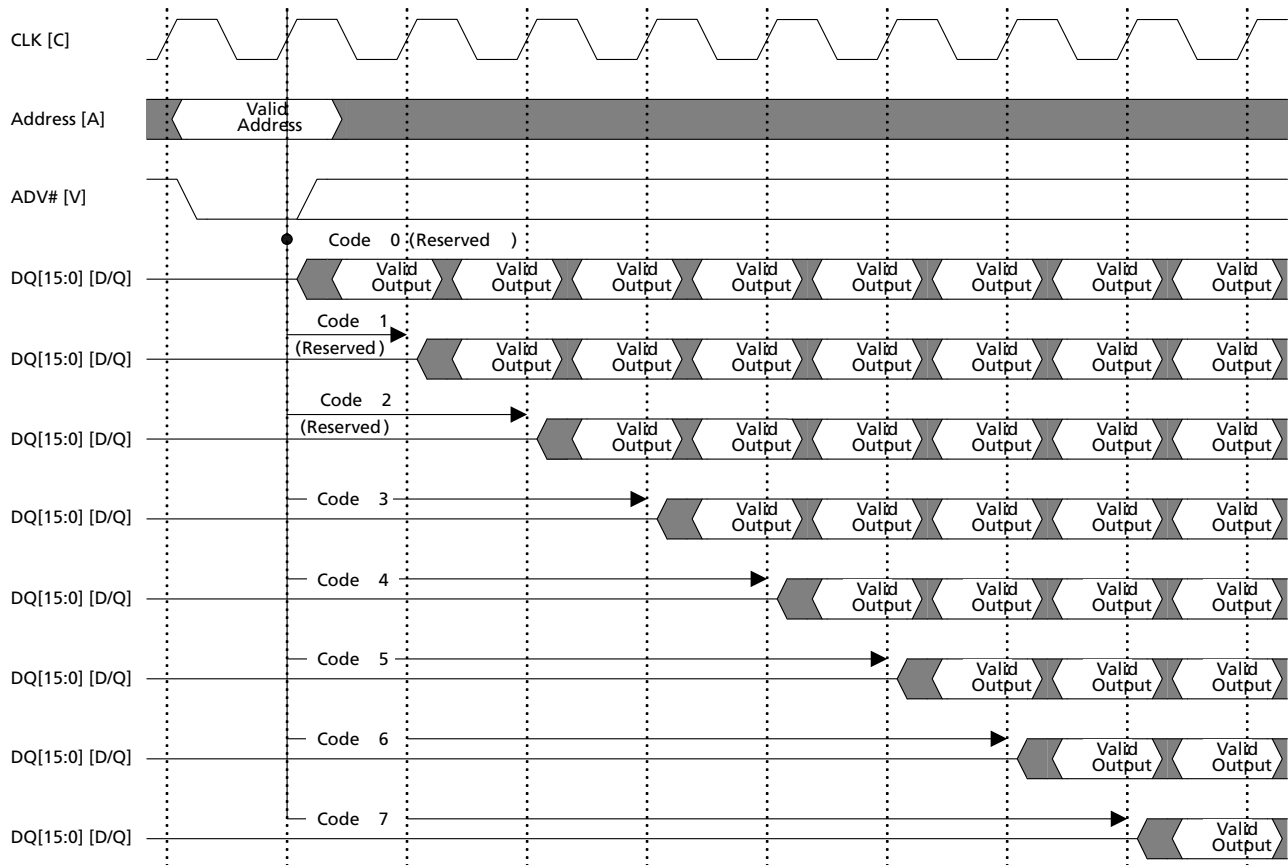
### Read Mode

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

## Latency Count

The latency count (LC) bits tell the device how many clock cycles must elapse from the rising edge of ADV# (or from the first valid clock edge after ADV# is asserted) until the first valid data word is driven to DQ[15:0]. The input clock frequency is used to determine this value. The First Access Latency Count figure shows the data output latency for different LC settings.

**Figure 10: First Access Latency Count**

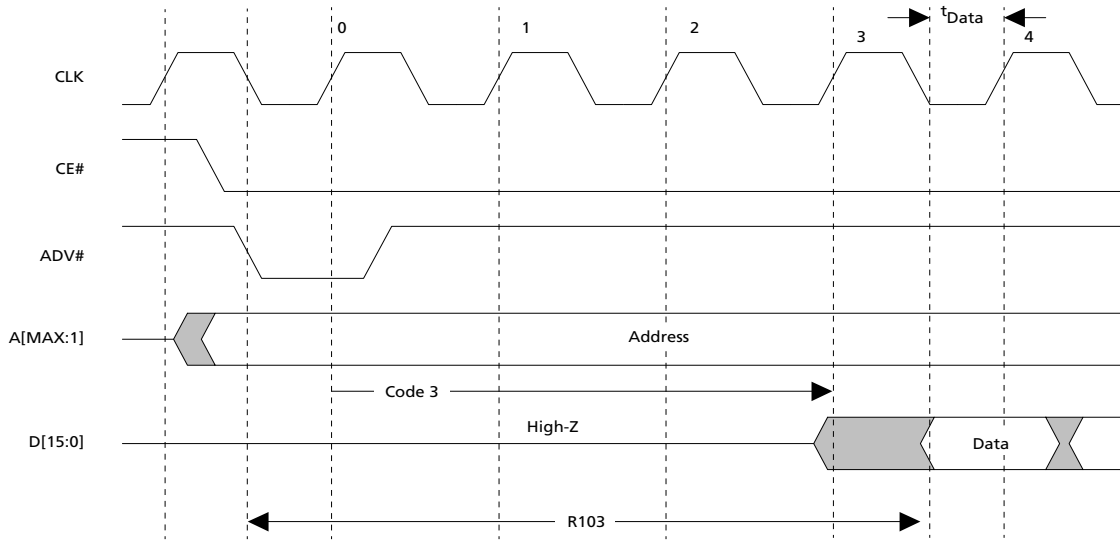


Note: 1. First Access Latency Count Calculation:

- $1 / \text{CLK frequency} = \text{CLK period (ns)}$
- $n \times (\text{CLK period}) \geq t_{\text{AVQV}} (\text{ns}) - t_{\text{CHQV}} (\text{ns})$
- Latency Count = n



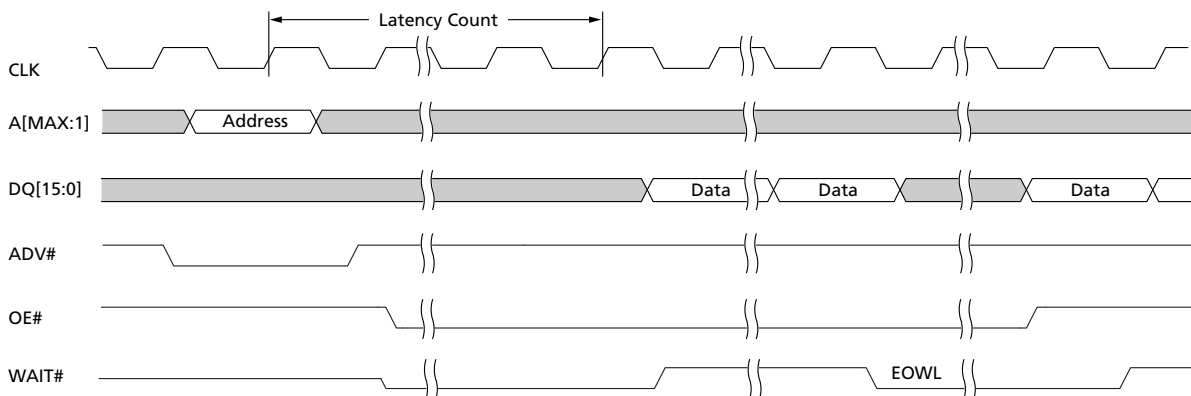
Figure 11: Example Latency Count Setting Using Code 3



### End of Wordline Considerations

End of wordline (EOWL) wait states can result when the starting address of the burst operation is not aligned to a 16-word boundary; that is, A[4:1] of the start address does not equal 0x0. The figure below illustrates the end of wordline wait state(s) that occur after the first 16-word boundary is reached. The number of data words and wait states is summarized in the table below.

Figure 12: End of Wordline Timing Diagram



**Table 14: End of Wordline Data and WAIT State Comparison**

| Latency Count | 130nm         |               | 65nm          |               |
|---------------|---------------|---------------|---------------|---------------|
|               | Data Words    | WAIT States   | Data Words    | WAIT States   |
| 1             | Not Supported | Not Supported | Not Supported | Not Supported |
| 2             | 4             | 0 to 1        | Not Supported | Not Supported |
| 3             | 4             | 0 to 2        | Not Supported | Not Supported |
| 4             | 4             | 0 to 3        | Not Supported | Not Supported |
| 5             | 4             | 0 to 4        | 16            | 0 to 4        |
| 6             | 4             | 0 to 5        | 16            | 0 to 5        |
| 7             | 4             | 0 to 6        | 16            | 0 to 6        |
| 8             | Not Supported | Not Supported | 16            | 0 to 7        |
| 9             |               |               | 16            | 0 to 8        |
| 10            |               |               | 16            | 0 to 9        |
| 11            |               |               | 16            | 0 to 10       |
| 12            |               |               | 16            | 0 to 11       |
| 13            |               |               | 16            | 0 to 12       |
| 14            |               |               | 16            | 0 to 13       |
| 15            |               |               | 16            | 0 to 14       |

### WAIT Signal Polarity and Functionality

The WAIT polarity (WP) bit, RCR10 determines the asserted level ( $V_{OH}$  or  $V_{OL}$ ) of WAIT. When WP is set, WAIT is asserted HIGH (default). When WP is cleared, WAIT is asserted LOW. The WAIT signal changes state on valid clock edges during active bus cycles (CE# asserted, OE# asserted, RST# de-asserted).

The WAIT signal indicates data valid when the device is operating in synchronous mode (RCR15 = 0). The WAIT signal is only de-asserted when data is valid on the bus. When the device is operating in synchronous nonarray read mode, such as read status, read ID, or read CFI, the WAIT signal is also de-asserted when data is valid on the bus. WAIT behavior during synchronous nonarray reads at the end of wordline works correctly only on the first data access. When the device is operating in asynchronous page mode, asynchronous single word read mode, and all write operations, WAIT is set to a de-asserted state as determined by RCR10.

**Table 15: WAIT Functionality Table**

| Condition                            | WAIT        | Notes |
|--------------------------------------|-------------|-------|
| CE# = 1, OE# = X or CE# = 0, OE# = 1 | High-Z      | 1     |
| CE# = 0, OE# = 0                     | Active      | 1     |
| Synchronous Array Reads              | Active      | 1     |
| Synchronous Nonarray Reads           | Active      | 1     |
| All Asynchronous Reads               | De-asserted | 1     |

**Table 15: WAIT Functionality Table (Continued)**

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

- Notes: 1. Active means that WAIT is asserted until data becomes valid, then deasserts.  
 2. When OE# = V<sub>IH</sub> during writes, WAIT = High-Z.

## WAIT Delay

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

## Burst Sequence

The burst sequence (BS) bit selects linear burst sequence (default). Only linear burst sequence is supported. The synchronous burst sequence for all burst lengths, as well as the effect of the burst wrap (BW) setting are shown below.

**Table 16: Burst Sequence Word Ordering**

| Start Address (DEC) | Burst Wrap (RCR3) | Burst Addressing Sequence (DEC) |                                |                                 |                                    |
|---------------------|-------------------|---------------------------------|--------------------------------|---------------------------------|------------------------------------|
|                     |                   | 4-Word Burst (BL[2:0] = 0b001)  | 8-Word Burst (BL[2:0] = 0b010) | 16-Word Burst (BL[2:0] = 0b011) | Continuous Burst (BL[2:0] = 0b111) |
| 0                   | 0                 | 0-1-2-3                         | 0-1-2-3-4-5-6-7                | 0-1-2-3-4...14-15               | 0-1-2-3-4-5-6-...                  |
| 1                   | 0                 | 1-2-3-0                         | 1-2-3-4-5-6-7-0                | 1-2-3-4-5...15-0                | 1-2-3-4-5-6-7-...                  |
| 2                   | 0                 | 2-3-0-1                         | 2-3-4-5-6-7-0-1                | 2-3-4-5-6...15-0-1              | 2-3-4-5-6-7-8-...                  |
| 3                   | 0                 | 3-0-1-2                         | 3-4-5-6-7-0-1-2                | 3-4-5-6-7...15-0-1-2            | 3-4-5-6-7-8-9-...                  |
| 4                   | 0                 |                                 | 4-5-6-7-0-1-2-3                | 4-5-6-7-8...15-0-1-2-3          | 4-5-6-7-8-9-10...                  |
| 5                   | 0                 |                                 | 5-6-7-0-1-2-3-4                | 5-6-7-8-9...15-0-1-2-3-4        | 5-6-7-8-9-10-11...                 |
| 6                   | 0                 |                                 | 6-7-0-1-2-3-4-5                | 6-7-8-9-10...15-0-1-2-3-4-5     | 6-7-8-9-10-11-12-...               |
| 7                   | 0                 |                                 | 7-0-1-2-3-4-5-6                | 7-8-9-10...15-0-1-2-3-4-5-6     | 7-8-9-10-11-12-13...               |
| ⋮                   | ⋮                 | ⋮                               | ⋮                              | ⋮                               | ⋮                                  |
| 14                  | 0                 |                                 |                                | 14-15-0-1-2...12-13             | 14-15-16-17-18-19-20-...           |
| 15                  | 0                 |                                 |                                | 15-0-1-2-3...13-14              | 15-16-17-18-19-20-21-...           |
| ⋮                   | ⋮                 | ⋮                               | ⋮                              | ⋮                               | ⋮                                  |
| 0                   | 1                 | 0-1-2-3                         | 0-1-2-3-4-5-6-7                | 0-1-2-3-4...14-15               | 0-1-2-3-4-5-6-...                  |
| 1                   | 1                 | 1-2-3-4                         | 1-2-3-4-5-6-7-8                | 1-2-3-4-5...15-16               | 1-2-3-4-5-6-7-...                  |
| 2                   | 1                 | 2-3-4-5                         | 2-3-4-5-6-7-8-9                | 2-3-4-5-6...16-17               | 2-3-4-5-6-7-8-...                  |
| 3                   | 1                 | 3-4-5-6                         | 3-4-5-6-7-8-9-10               | 3-4-5-6-7...17-18               | 3-4-5-6-7-8-9-...                  |

**Table 16: Burst Sequence Word Ordering (Continued)**

| Start Address (DEC) | Burst Wrap (RCR3) | Burst Addressing Sequence (DEC) |                                |                                 |                                    |
|---------------------|-------------------|---------------------------------|--------------------------------|---------------------------------|------------------------------------|
|                     |                   | 4-Word Burst (BL[2:0] = 0b001)  | 8-Word Burst (BL[2:0] = 0b010) | 16-Word Burst (BL[2:0] = 0b011) | Continuous Burst (BL[2:0] = 0b111) |
| 4                   | 1                 |                                 | 4-5-6-7-8-9-10-11              | 4-5-6-7-8...18-19               | 4-5-6-7-8-9-10...                  |
| 5                   | 1                 |                                 | 5-6-7-8-9-10-11-12             | 5-6-7-8-9...19-20               | 5-6-7-8-9-10-11...                 |
| 6                   | 1                 |                                 | 6-7-8-9-10-11-12-13            | 6-7-8-9-10...20-21              | 6-7-8-9-10-11-12-...               |
| 7                   | 1                 |                                 | 7-8-9-10-11-12-13-14           | 7-8-9-10-11...21-22             | 7-8-9-10-11-12-13...               |
| ⋮                   | ⋮                 | ⋮                               | ⋮                              | ⋮                               | ⋮                                  |
| 14                  | 1                 |                                 |                                | 14-15-16-17-18...28-29          | 14-15-16-17-18-19-20-...           |
| 15                  | 1                 |                                 |                                | 15-16-17-18-19...29-30          | 15-16-17-18-19-20-21-...           |

### Clock Edge

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

### Burst Wrap

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

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

### Burst Length

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

Continuous burst accesses are linear only and do not wrap within any word length boundaries. When a burst cycle begins, the device outputs synchronous burst data until it reaches the end of the "burstable" address space.

## One-Time Programmable Registers

### Read OTP Registers

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

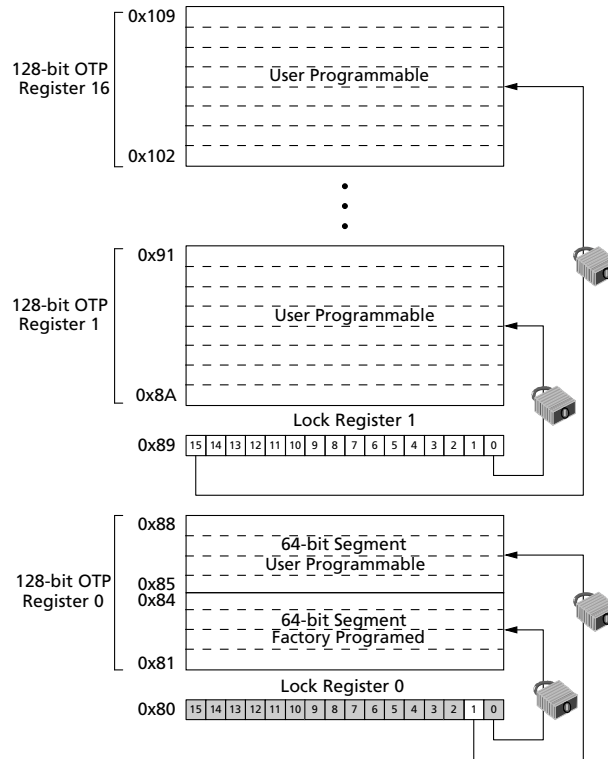
The first 128-bit OTP register is comprised of two 64-bit (8-word) segments. The lower 64-bit segment is preprogrammed at the Micron factory with a unique 64-bit number. The upper 64-bit segment, as well as the other sixteen 128-bit OTP registers, are blank. Users can program them as needed. Once programmed, users can also lock the OTP register(s) to prevent additional bit programming (see the OTP Register Map figure below).

The OTP registers contain OTP bits; when programmed, PR bits cannot be erased. Each OTP register can be accessed multiple times to program individual bits, as long as the register remains unlocked.

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

The OTP registers can be read from an OTP-RA address. To read the OTP register, a READ DEVICE IDENTIFIER command is issued at an OTP-RA address to place the device in the read device identifier state. Next, a READ operation is performed using the address offset corresponding to the register to be read. The Device Identifier Information table shows the address offsets of the OTP registers and lock registers. PR data is read 16 bits at a time.

**Figure 13: OTP Register Map**



## Program OTP Registers

To program an OTP register, a PROGRAM OTP REGISTER command is issued at the parameter's base address plus the offset of the desired OTP register location. Next, the desired OTP register data is written to the same OTP register address.

The device programs the 64-bit and 128-bit user-programmable OTP register data 16 bits at a time. Issuing the PROGRAM OTP REGISTER command outside of the OTP register's address space causes a program error (SR4 set). Attempting to program a locked OTP register causes a program error (SR4 set) and a lock error (SR1 set).

## Lock OTP Registers

Each OTP register can be locked by programming its respective lock bit in the lock register. The corresponding bit in the lock register is programmed by issuing the PROGRAM LOCK REGISTER command, followed by the desired lock register data. The physical addresses of the lock registers are 0x80 for register 0 and 0x89 for register 1; these addresses are used when programming the lock registers.

Bit 0 of lock register 0 is programmed during the manufacturing process, locking the lower-half segment of the first 128-bit OTP register. Bit 1 of lock register 0, which corresponds to the upper-half segment of the first 128-bit OTP register, can be programmed by the user. When programming bit 1 of lock register 0, all other bits need to be left as 1 such that the data programmed is 0xFFFFD.

Lock register 1 controls the the upper sixteen 128-bit OTP registers. Each bit of lock register 1 corresponds to a specific 128-bit OTP register. Programming a bit in lock register 1 locks the corresponding 128-bit OTP register; e.g., programming LR1.0 locks the corresponding OTP register 1.

**Note:** Once locked, the OTP registers cannot be unlocked.

## Common Flash Interface

The CFI is part of an overall specification for multiple command-set and control-interface descriptions. System software can parse the CFI database structure to obtain information about the device, such as block size, density, bus width, and electrical specifications. The system software determines which command set to use to properly perform a WRITE command, a BLOCK ERASE or READ command, and to otherwise control the device. Information in the CFI database can be viewed by issuing the READ CFI command.

### READ CFI Structure Output

The READ CFI command obtains CFI database structure information and always outputs it on the lower byte, DQ[7:0], for a word-wide (x16) device. This CFI-compliant device always outputs 00h data on the upper byte (DQ[15:8]).

The numerical offset value is the address relative to the maximum bus width the device supports. For this device family, the starting address is a 10h, which is a word address for x16 devices. For example, at this starting address of 10h, a READ CFI command outputs an ASCII Q in the lower byte and 00h in the higher byte as shown here.

In all the CFI tables shown here, address and data are represented in hexadecimal notation. In addition, because the upper byte of word-wide devices is always 00h, as shown in the example here, the leading 00 has been dropped and only the lower byte value is shown. Following is a table showing the CFI output for a x16 device, beginning at address 10h and a table showing an overview of the CFI database sections with their addresses.

**Table 17: Example of CFI Output (x16 device) as a Function of Device and Mode**

| Device  | Hex Offset | Hex Code           | ASCII Value (DQ[15:8]) | ASCII Value (DQ[7:0])        |
|---------|------------|--------------------|------------------------|------------------------------|
| Address | 00010:     | 51                 | 00                     | Q                            |
|         | 00011:     | 52                 | 00                     | R                            |
|         | 00012:     | 59                 | 00                     | Y                            |
|         | 00013:     | P_ID <sub>LO</sub> | 00                     | Primary vendor ID            |
|         | 00014:     | P_ID <sub>HI</sub> | 00                     |                              |
|         | 00015:     | P <sub>LO</sub>    | 00                     | Primary vendor table address |
|         | 00016:     | P <sub>HI</sub>    | 00                     |                              |
|         | 00017:     | A_ID <sub>LO</sub> | 00                     | Alternate vendor ID          |
|         | 00018:     | A_ID <sub>HI</sub> | 00                     |                              |
|         |            | :                  | :                      | :                            |
|         | :          | :                  | :                      | :                            |



**Table 18: CFI Database: Addresses and Sections**

| Address  | Section Name                           | Description   |
|----------|--|---|
| 00001:Fh | Reserved                               | Reserved for vendor-specific information  |
| 00010h   | CFI ID string                          | Flash device command set ID (identification) and vendor data offset   |
| 0001Bh   | System interface information           | Flash device timing and voltage   |
| 00027h   | Device geometry definition             | Flash device layout   |
| P        | Primary Micron-specific extended query | Vendor-defined informaton specific to the primary vendor algorithm (offset 15 defines <i>P</i> which points to the primary Micron-specific extended query table.) |

**Table 19: CFI ID String**

| Hex Offset | Length | Description  | Address | Hex Code | ASCII Value (DQ[7:0])                             |
|------------|--------|--|---------|----------|---|
| 10h        | 3      | Query unique ASCII string "QRY"  | 10:     | - -51    | Q   |
|            |        |  | 11:     | - -52    | R   |
|            |        |  | 12:     | - -59    | Y   |
| 13h        | 2      | Primary vendor command set and control interface ID code. 16-bit ID code for vendor-specified algorithms.            | 13:     | - -01    | Primary vendor ID number                          |
|            |        |  | 14:     | - -00    |   |
| 15h        | 2      | Extended query table primary algorithm address.  | 15:     | - -0A    | Primary vendor table address, primary algorithm   |
|            |        |  | 16:     | - -01    |   |
| 17h        | 2      | Alternate vendor command set and control interface ID code. 0000h means no second vendor-specified algorithm exists. | 17:     | - -00    | Alternate vendor ID number                        |
|            |        |  | 18:     | - -00    |   |
| 19h        | 2      | Secondary algorithm extended query table address. 0000h means none exists.   | 19:     | - -00    | Primary vendor table address, secondary algorithm |
|            |        |  | 1A:     | - -00    |   |

Note: 1. The CFI ID string provides verification that the device supports the CFI specification. It also indicates the specification version and supported vendor-specific command sets.

**Table 20: System Interface Information**

| Hex Offset | Length | Description  | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|--|---------|----------|-----------------------|
| 1Bh        | 1      | V <sub>CC</sub> logic supply minimum program/erase voltage.<br>bits 0 - 3 BCD 100 mV<br>bits 4 - 7 BCD volts         | 1Bh     | --23     | 2.3V                  |
| 1Ch        | 1      | V <sub>CC</sub> logic supply maximum program/erase voltage.<br>bits 0 - 3 BCD 100 mV<br>bits 4 - 7 BCD volts         | 1Ch     | --36     | 3.6V                  |
| 1Dh        | 1      | V <sub>PP</sub> [programming] supply minimum program/erase voltage.<br>bits 0 - 3 BCD 100 mV<br>bits 4 - 7 hex volts | 1Dh     | --85     | 8.5V                  |
| 1Eh        | 1      | V <sub>PP</sub> [programming] supply maximum program/erase voltage.<br>bits 0 - 3 BCD 100 mV<br>bits 4 - 7 hex volts | 1Eh     | --95     | 9.5V                  |
| 1Fh        | 1      | "n" such that typical single word program timeout = 2 <sup>n</sup> μs.   | 1Fh     | --09     | 512μs                 |
| 20h        | 1      | "n" such that typical full buffer write timeout = 2 <sup>n</sup> μs.   | 20h     | --0A     | 1024μs                |
| 21h        | 1      | "n" such that typical block erase timeout = 2 <sup>n</sup> ms.   | 21h     | --0A     | 1s                    |
| 22h        | 1      | "n" such that typical full chip erase timeout = 2 <sup>n</sup> ms.   | 22h     | --00     | NA                    |
| 23h        | 1      | "n" such that maximum word program timeout = 2 <sup>n</sup> times typical.   | 23h     | --01     | 1024μs                |
| 24h        | 1      | "n" such that maximum buffer write timeout = 2 <sup>n</sup> times typical.   | 24h     | --02     | 4096μs                |
| 25h        | 1      | "n" such that maximum block erase timeout = 2 <sup>n</sup> times typical.  | 25h     | --02     | 4s                    |
| 26h        | 1      | "n" such that maximum chip erase timeout = 2 <sup>n</sup> times typical.   | 26h     | --00     | NA                    |

**Table 21: Device Geometry**

| Hex Offset | Length | Description   | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|---|---------|----------|-----------------------|
| 27h        | 1      | n such that device size in bytes = 2 <sup>n</sup> .   | 27:     |          | See Note 1            |
| 28h        | 2      | Flash device interface code assignment: n such that n + 1 specifies the bit field that represents the flash device width capabilities as described here:<br>bit 0: x8<br>bit 1: x16<br>bit 2: x32<br>bit 3: x64<br>bits 4 - 7: –<br>bits 8 - 15: –  | 28:     | --01     | x16                   |
|            |        |   | 29:     | --00     |                       |
| 2Ah        | 2      | n such that maximum number of bytes in write buffer = 2 <sup>n</sup> .  | 2Ah     | --0A     | 1024                  |
|            |        |   | 2Bh     | --00     |                       |
| 2Ch        | 1      | Number of erase block regions (x) within the device:<br>1) x = 0 means no erase blocking; the device erases in bulk.<br>2) x specifies the number of device regions with one or more contiguous, same-size erase blocks.<br>3) Symmetrically blocked partitions have one blocking region. | 2Ch     |          | See Note 1            |
| 2Dh        | 4      | Erase block region 1 information:<br>bits 0 - 15 = y, y + 1 = number of identical-size erase blocks.<br>bits 16 - 31 = z, region erase block(s) size are z x 256 bytes.   | 2D:     |          | See Note 1            |
|            |        |   | 2E:     |          |                       |
|            |        |   | 2F:     |          |                       |
|            |        |   | 30:     |          |                       |
| 31h        | 4      | Erase block region 2 information:<br>bits 0 - 15 = y, y + 1 = number of identical-size erase blocks.<br>bits 16 - 31 = z, region erase block(s) size are z x 256 bytes.   | 31:     |          | See Note 1            |
|            |        |   | 32:     |          |                       |
|            |        |   | 33:     |          |                       |
|            |        |   | 34:     |          |                       |
| 35h        | 4      | Reserved for future erase block region information.   | 35:     |          | See Note 1            |
|            |        |   | 36:     |          |                       |
|            |        |   | 37:     |          |                       |
|            |        |   | 38:     |          |                       |

Note: 1. See Block Region Map Information table.

**Table 22: Block Region Map Information**

| Address | 512Mb |        |             | 1Gb  |        |             | 2Gb         |
|---------|-------|--------|-------------|------|--------|-------------|-------------|
|         | Top   | Bottom | Symmetrical | Top  | Bottom | Symmetrical | Symmetrical |
| 27:     | --1A  | --1A   | --1A        | --1B | --1B   | --1B        | --1B        |
| 28:     | --01  | --01   | --01        | --01 | --01   | --01        | --01        |
| 29:     | --00  | --00   | --00        | --00 | --00   | --00        | --00        |
| 2A:     | --0A  | --0A   | --0A        | --0A | --0A   | --0A        | --0A        |
| 2B:     | --00  | --00   | --00        | --00 | --00   | --00        | --00        |
| 2C:     | --02  | --02   | --01        | --02 | --02   | --01        | --01        |
| 2D:     | --FE  | --03   | --FF        | --FE | --03   | --FF        | --FF        |

**Table 22: Block Region Map Information (Continued)**

| Address | 512Mb |        |             | 1Gb  |        |             | 2Gb         |
|---------|-------|--------|-------------|------|--------|-------------|-------------|
|         | Top   | Bottom | Symmetrical | Top  | Bottom | Symmetrical | Symmetrical |
| 2E:     | --01  | --00   | --01        | --03 | --00   | --03        | --03        |
| 2F:     | --00  | --80   | --00        | --00 | --80   | --00        | --00        |
| 30:     | --02  | --00   | --02        | --02 | --00   | --02        | --02        |
| 31:     | --03  | --FE   | --00        | --03 | --FE   | --00        | --00        |
| 32:     | --00  | --01   | --00        | --00 | --03   | --00        | --00        |
| 33:     | --80  | --00   | --00        | --80 | --00   |             | --00        |
| 34:     | --00  | --02   | --00        | --00 | --02   |             | --00        |
| 35:~38: | --00  | --00   | --00        | --00 | --00   |             | --00        |

**Table 23: Primary Vendor-Specific Extended Query**

| Hex Offset<br>P = 10Ah     | Length | Description  | Address | Hex Code | ASCII Value<br>(DQ[7:0]) |
|----------------------------|--------|--|---------|----------|--------------------------|
| (P+0)h<br>(P+1)h<br>(P+2)h | 3      | Primary extended query table, unique ASCII string: PRI | 10A:    | - -50    | P                        |
|                            |        |  | 10B:    | - -52    | R                        |
|                            |        |  | 10C:    | - -49    | I                        |
| (P+3)h                     | 1      | Major version number, ASCII                            | 10D:    | - -31    | 1                        |
| (P+4)h                     | 1      | Minor version number, ASCII                            | 10E:    | - -35    | 5                        |

**Table 23: Primary Vendor-Specific Extended Query (Continued)**

| Hex Offset<br>P = 10Ah   | Length     | Description  | Address                      | Hex Code   | ASCII Value<br>(DQ[7:0]) |
|--|------------|--|------------------------------|------------|--------------------------|
| (P+5)h<br>(P+6)h<br>(P+7)h<br>(P+8)h                                   | 4          | Optional feature and command support (1 = yes; 0 = no)   | 10F:                         | --E6       | –                        |
|  |            | Bits 11 - 29 are reserved; undefined bits are 0<br>If bit 31 = 1, then another 31-bit field of optional features follows at the end of the bit 30 field. | 110:                         | --01       | –                        |
|  |            |  | 111:                         | --00       | –                        |
|  |            |  | 112:                         | See Note 1 | –                        |
|  |            |  | Bit 0: Chip erase supported. | bit 0 = 0  | No                       |
|  |            | Bit 1: Suspend erase supported.  | bit 1 = 1                    | Yes        |                          |
|  |            | Bit 2: Suspend program supported.  | bit 2 = 1                    | Yes        |                          |
|  |            | Bit 3: Legacy lock/unlock supported.   | bit 3 = 0                    | No         |                          |
|  |            | Bit 4: Queued erase supported.   | bit 4 = 0                    | No         |                          |
|  |            | Bit 5: Instant individual block locking supported.   | bit 5 = 1                    | Yes        |                          |
|  |            | Bit 6: OTP bits supported.   | bit 6 = 1                    | Yes        |                          |
|  |            | Bit 7: Page mode read supported.   | bit 7 = 1                    | Yes        |                          |
|  |            | Bit 8: Synchronous read supported.   | bit 8 = 1                    | Yes        |                          |
|  |            | Bit 9: Simultaneous operations supported.  | bit 9 = 0                    | No         |                          |
|  |            | Bit 10: Extended Flash array block supported.  | bit 10 = 0                   | No         |                          |
|  |            | Bit 11: Permanent block locking of up to full main array supported.  | bit 11 = 0                   | Yes        |                          |
| Bit 12: Permanent block locking of up to partial main array supported. | bit 12 = 0 | No   |                              |            |                          |
| Bit 30: CFI links to follow:   | bit 30     | See Note 1   |                              |            |                          |
| Bit 31: Another optional features field to follow.                     | bit 31     |  |                              |            |                          |
| (P+9)h   | 1          | Supported functions after SUSPEND: READ ARRAY, STATUS, QUERY. Other supported options include:<br>Bits 1 - 7: Reserved; undefined bits are 0.            | 113:                         | --01       | –                        |
|  |            | Bit 0: Program supported after ERASE SUSPEND.  | bit 0 = 1                    | Yes        |                          |
| (P+A)h<br>(P+B)h   | 2          | Block Status Register mask:<br>Bits 2 - 15 are reserved; undefined bits are 0.   | 114:                         | --03       | –                        |
|  |            | Bit 0: Block lock-bit status register active.  | 115:                         | --00       | –                        |
|  |            |  | bit 0 = 1                    | Yes        |                          |
|  |            |  | bit 1 = 1                    | Yes        |                          |
|  |            |  | bit 4 = 0                    | No         |                          |
| Bit 5: EFA block lock-bit status active.                               | bit 5 = 0  | No   |                              |            |                          |
| (P+C)h   | 1          | V <sub>CC</sub> logic supply highest performance program/erase voltage.<br>bits 0 - 3 BCD 100 mV<br>bits 4 - 7 hex value in volts                        | 116:                         | --30       | 3.0V                     |

**Table 23: Primary Vendor-Specific Extended Query (Continued)**

| Hex Offset<br>P = 10Ah | Length | Description   | Address | Hex Code | ASCII Value<br>(DQ[7:0]) |
|------------------------|--------|---|---------|----------|--------------------------|
| (P+D)h                 | 1      | V <sub>PP</sub> optimum program/erase voltage.<br>bits 0 - 3 BCD 100mV<br>bits 4 - 7 hex value in volts | 117:    | --90     | 9.0V                     |

Note: 1. See Optional Features Fields table.

**Table 24: Optional Features Field**

| Address | Discrete |      | 2Gb       |           |           |           |
|---------|----------|------|-----------|-----------|-----------|-----------|
|         | Bottom   | Top  | Bottom    |           | Top       |           |
|         | –        | –    | die 1 (B) | die 2 (T) | die 1 (T) | die 2 (B) |
| 112:    | --00     | --00 | 40:       | --00      | --40      | --00      |

**Table 25: One Time Programmable (OTP) Space Information**

| Hex Offset<br>P = 10Ah                  | Length | Description  | Address | Hex Code | ASCII Value<br>(DQ[7:0]) |
|---|--------|--|---------|----------|--------------------------|
| (P+E)h                                  | 1      | Number of OTP block fields in JEDEC ID space.<br>00h indicates that 256 OTP fields are available.  | 118:    | --02     | 2                        |
| (P+F)h<br>(P+10)h<br>(P+11)h<br>(P+12)h | 4      | OTP Field 1: OTP Description:<br>This field describes user-available OTP bytes.<br>Some are preprogrammed with device-unique serial numbers. Others are user-programmable.<br>Bits 0-15 point to the OTP Lock byte (the first byte).<br>The following bytes are factory preprogrammed and user-programmable:<br>Bits 0 - 7 = Lock/bytes JEDEC plane physical low address.<br>Bits 8 - 15 = Lock/bytes JEDEC plane physical high address.<br>Bits 16 - 23 = n where 2 <sup>n</sup> equals factory preprogrammed bytes.<br>Bits 24 - 31 = n where 2 <sup>n</sup> equals user-programmable bytes. | 119:    | --80     | 80h                      |
|   |        |  | 11A:    | --00     | 00h                      |
|   |        |  | 1B:     | --03     | 8 byte                   |
|   |        |  | 11C:    | --03     | 8 byte                   |

**Table 25: One Time Programmable (OTP) Space Information (Continued)**

| Hex Offset<br>P = 10Ah                   | Length | Description   | Address | Hex Code | ASCII Value<br>(DQ[7:0]) |
|--|--------|---|---------|----------|--------------------------|
| (P+13)h<br>(P+14)h<br>(P+15)h<br>(P+16)h | 10     | Protection field 2: protection description<br>Bits 0 - 31 point to the protection register physical lock word address in the JEDEC plane.<br>The bytes that follow are factory or user-programmable.                      | 11D:    | - -89    | 89h                      |
| 11E:                                     |        |   | - -00   | 00h      |                          |
| 11F:                                     |        |   | - -00   | 00h      |                          |
| 120:                                     |        |   | - -00   | 00h      |                          |
| (P+17)h<br>(P+18)h<br>(P+19)h            |        | Bits 32 - 39 = n where n equals factory programmed groups (low byte).<br>Bits 40 - 47 = n where n equals factory programmed groups (high byte).<br>Bits 48 - 55 = n where 2n equals factory programmed bytes/groups.      | 121:    | - -00    | 0                        |
| 122:                                     |        |   | - -00   | 0        |                          |
| 123:                                     |        |   | - -00   | 0        |                          |
| (P+1A)h<br>(P+1B)h<br>(P+1C)h            |        | Bits 56 - 63 = n where n equals user programmed groups (low byte).<br>Bits 64 - 71 = n where n equals user programmed groups (high byte).<br>Bits 72 - 79 = n where 2 <sup>n</sup> equals user programmable bytes/groups. | 124:    | - -10    | 16                       |
| 125:                                     |        |   | - -00   | 0        |                          |
| 126:                                     |        |   | - -04   | 16       |                          |

**Table 26: Burst Read Information**

| Hex Offset<br>P = 10Ah | Length | Description   | Address | Hex Code | ASCII Value<br>(DQ[7:0]) |
|------------------------|--------|---|---------|----------|--------------------------|
| (P+1D)h                | 1      | Page Mode Read capability:<br>Bits 7 - 0 = n where 2 <sup>n</sup> hex value represents the number of read-page bytes. See offset 28h for device word width to determine page-mode data output width. 00h indicates no read page buffer. | 127:    | - -05    | 32 byte                  |
| (P+1E)h                | 1      | Number of synchronous mode read configuration fields that follow. 00h indicates no burst capability.  | 128:    | - -04    | 4                        |

**Table 26: Burst Read Information (Continued)**

| Hex Offset<br>P = 10Ah | Length | Description  | Address | Hex Code | ASCII Value<br>(DQ[7:0]) |
|------------------------|--------|--|---------|----------|--------------------------|
| (P+1F)h                | 1      | Synchronous mode read capability configuration 1:<br>Bits 3 - 7 = Reserved.<br>Bits 0 - 2 = n where $2^{n+1}$ hex value represents the maximum number of continuous synchronous reads when the device is configured for its maximum word width.<br>A value of 07h indicates that the device is capable of continuous linear bursts that will output data until the internal burst counter reaches the end of the device's burstable address space.<br>This field's 3-bit value can be written directly to the Read Configuration Register bits 0 - 2 if the device is configured for its maximum word width. See offset 28h for word width to determine the burst data output width. | 129:    | --01     | 4                        |
| (P+20)h                | 1      | Synchronous mode read capability configuration 2.  | 12A:    | --02     | 8                        |
| (P+21)h                | 1      | Synchronous mode read capability configuration 3.  | 12B:    | --03     | 16                       |
| (P+22)                 | 1      | Synchronous mode read capability configuration 4.  | 12C:    | --07     | Continued                |

**Table 27: Partition and Block Erase Region Information**

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



**Table 28: Partition Region 1 Information: Top and Bottom Offset/Address**

| Hex Offset<br>P = 10Ah |                    | Description<br>Optional Flash features and commands  | Length | Address      |              |
|------------------------|--------------------|--|--------|--------------|--------------|
| Bottom                 | Top                |  |        | Bottom       | Top          |
| (P+24)h<br>(P+25)h     | (P+24)h<br>(P+25)h | Data size of this Partition Region information field (number of addressable locations, including this field).  | 2      | 12E:<br>12F: | 12E:<br>12F: |
| (P+26)h<br>(P+27)h     | (P+26)h<br>(P+27)h | Number of identical partitions within the partition region.  | 2      | 130:<br>131: | 130:<br>131: |
| (P+28)h                | (P+28)h            | Number of program or erase operations allowed in a partition:<br>Bits 0 - 3 = Number of simultaneous program operations.<br>Bits 4 - 7 = Number of simultaneous erase operations.  | 1      | 132:         | 132:         |
| (P+29)h                | (P+29)h            | Simultaneous program or erase operations allowed in other partitions while a partition in this region is in program mode:<br>Bits 0 - 3 = Number of simultaneous program operations.<br>Bits 4 - 7 = Number of simultaneous erase operations.  | 1      | 133:         | 133:         |
| (P+2A)h                | (P+2A)h            | Simultaneous program or erase operations allowed in other partitions while a partition in this region is in erase mode:<br>Bits 0 - 3 = Number of simultaneous program operations.<br>Bits 4 - 7 = Number of simultaneous erase operations.  | 1      | 134:         | 134:         |
| (P+2B)h                | (P+2B)h            | Types of erase block regions in this partition region:<br>x=0: No erase blocking; the partition region erases in bulk.<br>x = Number of erase block regions with contiguous, same-size erase blocks.<br>Symmetrically blocked partitions have one blocking region.<br>Partition size = (Type 1 blocks) x (Type 1 block sizes) + (Type 2 blocks) x (Type 2 block sizes) +...+ (Type n blocks) x (Type n block sizes). | 1      | 135:         | 135:         |

**Table 29: Partition Region 1 Information**

| Hex Offset<br>P = 10Ah<br>Bottom/Top     | Description<br>Optional Flash features and commands   | Length | Address<br>Bottom/Top        |
|--|---|--------|------------------------------|
| (P+2C)h<br>(P+2D)h<br>(P+2E)h<br>(P+2F)h | Partition region 1 erase block type 1 information:<br>Bits 0-15 = y, y+1 = Number of identical-sized erase blocks in a partition.<br>Bits 16-31 = z, where region erase block(s) size is z x 256 bytes. | 4      | 136:<br>137:<br>138:<br>139: |

**Table 29: Partition Region 1 Information (Continued)**

| Hex Offset<br>P = 10Ah<br>Bottom/Top                           | Description<br>Optional Flash features and commands   | Length | Address<br>Bottom/Top                        |
|--|---|--------|--|
| (P+30)h<br>(P+31)h   | Partition 1 (erase block type 1):<br>Minimum block erase cycles x 1000  | 2      | 13A:<br>13B:                                 |
| (P+32)h  | Partition 1 (erase block type 1) bits per cell; internal ECC:<br>Bits 0 - 3 = bits per cell in erase region<br>Bit 4 = reserved for "internal ECC used" (1=yes, 0=no)<br>Bit 5 - 7 = reserved for future use  | 1      | 13C:   |
| (P+33)h  | Partition 1 (erase block type 1) page mode and synchronous mode capabilities:<br>Bits 0 = page-mode host reads permitted (1=yes, 0=no)<br>Bit 1 = synchronous host reads permitted (1=yes, 0=no)<br>Bit 2 = synchronous host writes permitted (1=yes, 0=no)<br>Bit 3 - 7 = reserved for future use  | 1      | 13D:   |
| (P+34)h<br>(P+35)h<br>(P+36)h<br>(P+37)h<br>(P+38)h<br>(P+39)h | Partition 1 (erase block type 1) programming region information:<br>Bits 0 - 7 = x, 2 <sup>x</sup> : programming region aligned size (bytes)<br>Bit 8-14 = reserved for future use<br>Bit 15 = legacy flash operation; ignore 0:7<br>Bit 16 - 23 = y: control mode valid size (bytes)<br>Bit 24 - 31 = reserved for future use<br>Bit 32 - 39 = z: control mode invalid size (bytes)<br>Bit 40 - 46 = reserved for future use<br>Bit 47 = legacy flash operation (ignore 23:16 and 39:32) | 6      | 13E:<br>13F:<br>140:<br>141:<br>142:<br>143: |
| (P+3A)h<br>(P+3B)h<br>(P+3C)h<br>(P+3D)h                       | Partition 1 erase block type 2 information:<br>Bits 0-15 = y, y+1 = Number of identical-size erase blocks in a partition.<br>Bits 16 - 31 = z, where region erase block(s) size is z x 256 bytes.<br>(bottom parameter device only)   | 4      | 144:<br>145:<br>146:<br>147:                 |
| (P+3E)h<br>(P+3F)h   | Partition 1 (erase block type 2)<br>Minimum block erase cycles x 1000   | 2      | 148:<br>149:                                 |
| (P+40)h  | Partition 1 (erase block type 2) bits per cell, internal EDAC:<br>Bits 0 - 3 = bits per cell in erase region<br>Bit 4 = reserved for "internal ECC used" (1=yes, 0=no)<br>Bits 5 - 7 = reserved for future use  | 1      | 14A:   |
| (P+41)h  | Partition 1 (erase block type 2) page mode and synchronous mode capabilities:<br>Bit 0 = page-mode host reads permitted (1=yes, 0=no)<br>Bit 1 = synchronous host reads permitted (1=yes, 0=no)<br>Bit 2 = synchronous host writes permitted (1=yes, 0=no)<br>Bits 3-7 = reserved for future use  | 1      | 14B:   |

**Table 29: Partition Region 1 Information (Continued)**

| <b>Hex Offset<br/>P = 10Ah<br/>Bottom/Top</b> | <b>Description<br/>Optional Flash features and commands</b>  | <b>Length</b> | <b>Address<br/>Bottom/Top</b> |
|---|--|---------------|-------------------------------|
| (P+42)h                                       | Partition 1 (erase block type 2) programming region information:<br>Bits 0-7 = x, 2 <sup>n</sup> x = Programming region aligned size (bytes)<br>Bits 8-14 = reserved for future use<br>Bit 15 = legacy flash operation (ignore 0:7)<br>Bits 16 - 23 = y = Control mode valid size in bytes Bits 24 - 31 = reserved<br>Bits 32 - 39 = z = Control mode invalid size in bytes<br>Bits 40 - 46 = reserved<br>Bit 47 = legacy flash operation (ignore 23:16 and 39:32) | 6             | 14C:                          |
| (P+43)h                                       |  |               | 14D:                          |
| (P+44)h                                       |  |               | 14E:                          |
| (P+45)h                                       |  |               | 14F:                          |
| (P+46)h                                       |  |               | 150:                          |
| (P+47)h                                       |  |               | 151:                          |



**Table 30: Partition Region 1: Partition and Erase Block Map Information**

| Add.  | 512Mb |        |       | 1Gb  |        |       | 2Gb             |                 |
|-------|-------|--------|-------|------|--------|-------|-----------------|-----------------|
|       | Top   | Bottom | Symm. | Top  | Bottom | Symm. | Symm. Upper Die | Symm. Lower Die |
| 12D:  | --01  | --01   | --01  | --01 | --01   | --01  | --01            | --01            |
| 12E:  | --24  | --24   | --14  | --24 | --24   | --14  | --14            | --14            |
| 12F:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 130:  | --01  | --01   | --01  | --01 | --01   | --01  | --01            | --01            |
| 131:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 132:  | --11  | --11   | --11  | --11 | --11   | --11  | --11            | --11            |
| 133:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 134:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 135:  | --02  | --02   | --01  | --02 | --02   | --01  | --01            | --01            |
| 136:  | --FE  | --03   | --FF  | --FE | --03   | --FF  | --FF            | --FF            |
| 137:  | --01  | --00   | --01  | --03 | --00   | --03  | --03            | --03            |
| 138:  | --00  | --80   | --00  | --00 | --80   | --00  | --00            | --00            |
| 139:  | --02  | --00   | --02  | --02 | --00   | --02  | --02            | --02            |
| 13A:  | --64  | --64   | --64  | --64 | --64   | --64  | --64            | --64            |
| 13B:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 13C:  | --02  | --02   | --02  | --02 | --02   | --02  | --02            | --02            |
| 13D*: | --03  | --03   | --03  | --03 | --03   | --03  | --03            | --03            |
| 13E:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 13F:  | --80  | --80   | --80  | --80 | --80   | --80  | --80            | --80            |
| 140:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 141:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 142:  | --00  | --00   | --00  | --00 | --00   | --00  | --00            | --00            |
| 143:  | --80  | --80   | --80  | --80 | --80   | --80  | --80            | --80            |
| 144:  | --03  | --FE   | --FF  | --03 | --FE   | --FF  | --FF            | --10            |
| 145:  | --00  | --01   | --FF  | --00 | --03   | --FF  | --FF            | --C8            |
| 146:  | --80  | --00   | --FF  | --80 | --00   | --FF  | --FF            | --00            |
| 147:  | --00  | --02   | --FF  | --00 | --02   | --FF  | --FF            | --00            |
| 148:  | --64  | --64   | --FF  | --64 | --64   | --FF  | --FF            | --10            |
| 149:  | --00  | --00   | --FF  | --00 | --00   | --FF  | --FF            | --FF            |
| 14A:  | --02  | --02   | --FF  | --02 | --02   | --FF  | --FF            | --FF            |
| 14B:  | --03  | --03   | --FF  | --03 | --03   | --FF  | --FF            | --FF            |
| 14C:  | --00  | --00   | --FF  | --00 | --00   | --FF  | --FF            | --FF            |
| 14D:  | --80  | --80   | --FF  | --80 | --80   | --FF  | --FF            | --FF            |
| 14E:  | --00  | --00   | --FF  | --00 | --00   | --FF  | --FF            | --FF            |
| 14F:  | --00  | --00   | --FF  | --00 | --00   | --FF  | --FF            | --FF            |
| 150:  | --00  | --00   | --FF  | --00 | --00   | --FF  | --FF            | --FF            |

**Table 30: Partition Region 1: Partition and Erase Block Map Information (Continued)**

| Add. | 512Mb |        |       | 1Gb  |        |       | 2Gb             |                 |
|------|-------|--------|-------|------|--------|-------|-----------------|-----------------|
|      | Top   | Bottom | Symm. | Top  | Bottom | Symm. | Symm. Upper Die | Symm. Lower Die |
| 151: | --80  | --80   | --FF  | --80 | --80   | --FF  | --FF            | --FF            |

**Table 31: CFI Link Information – 2Gb**

| Length | Description  | Address                     | ASCII Value (DQ[7:0]) |
|--------|--|-----------------------------|-----------------------|
| 4      | CFI Link field bit definitions:<br>Bits 0 - 9 = Address offset (within 32Mb segment of referenced CFI table)<br>Bits 10 - 27 = nth 32Mb segment of referenced CFI table<br>Bits 28 - 30 = Memory Type<br>Bit 31 = Another CFI link field immediately follows | 144:<br>145:<br>146:<br>147 | See Note 1            |
| 1      | CFI Link field quantity subfield definitions:<br>Bits 0 - 3 = Quantity field (n such that n+1 equals quantity)<br>Bit 4 = Table and die relative location<br>Bit 5 = Link field and table relative location<br>Bits 6 - 7 = Reserved                         | 148:                        |                       |

Note: 1. See "Partition Region 1: Partition and Erase Block Map Information" table.

## Flowcharts

Figure 14: Word Program Procedure

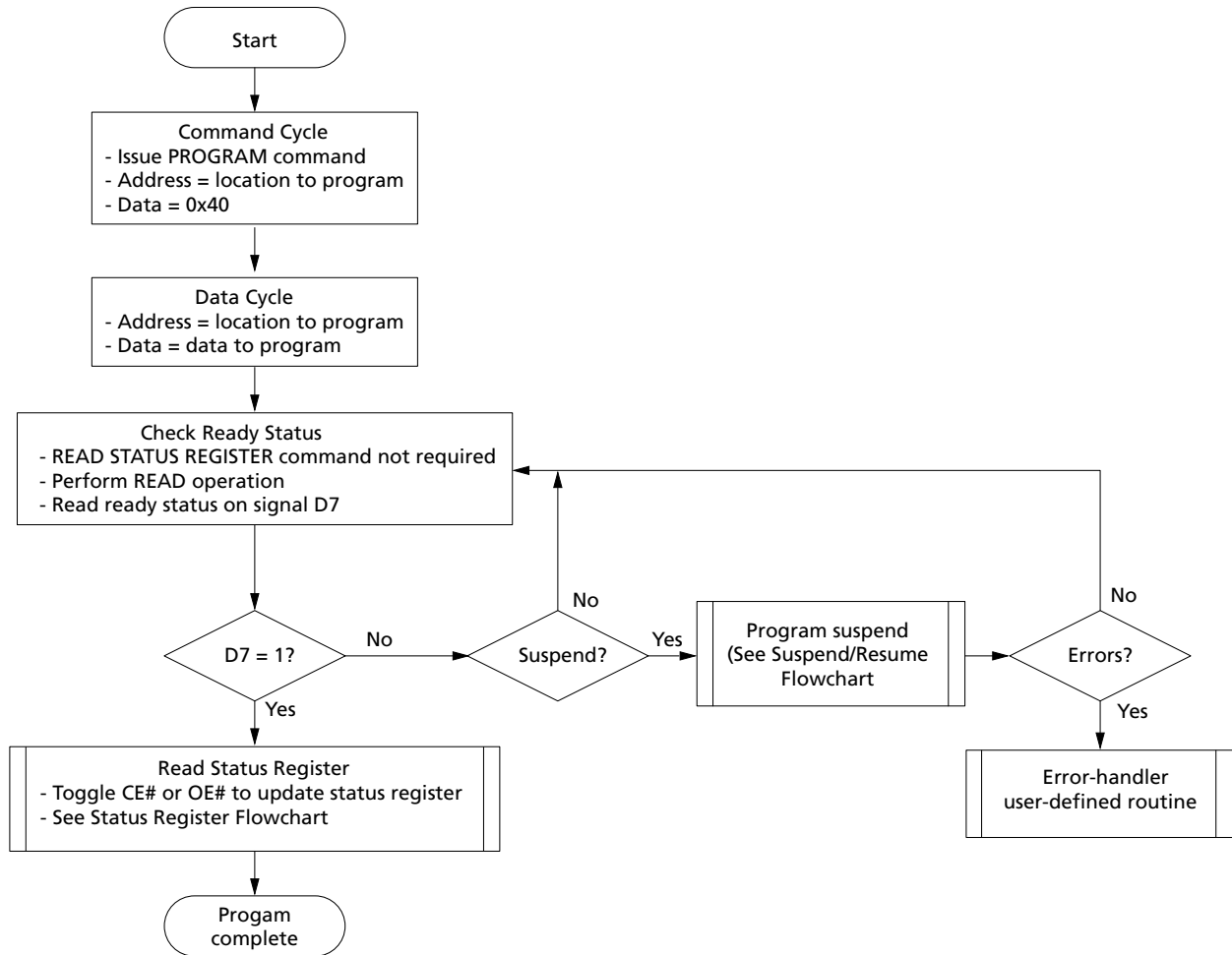
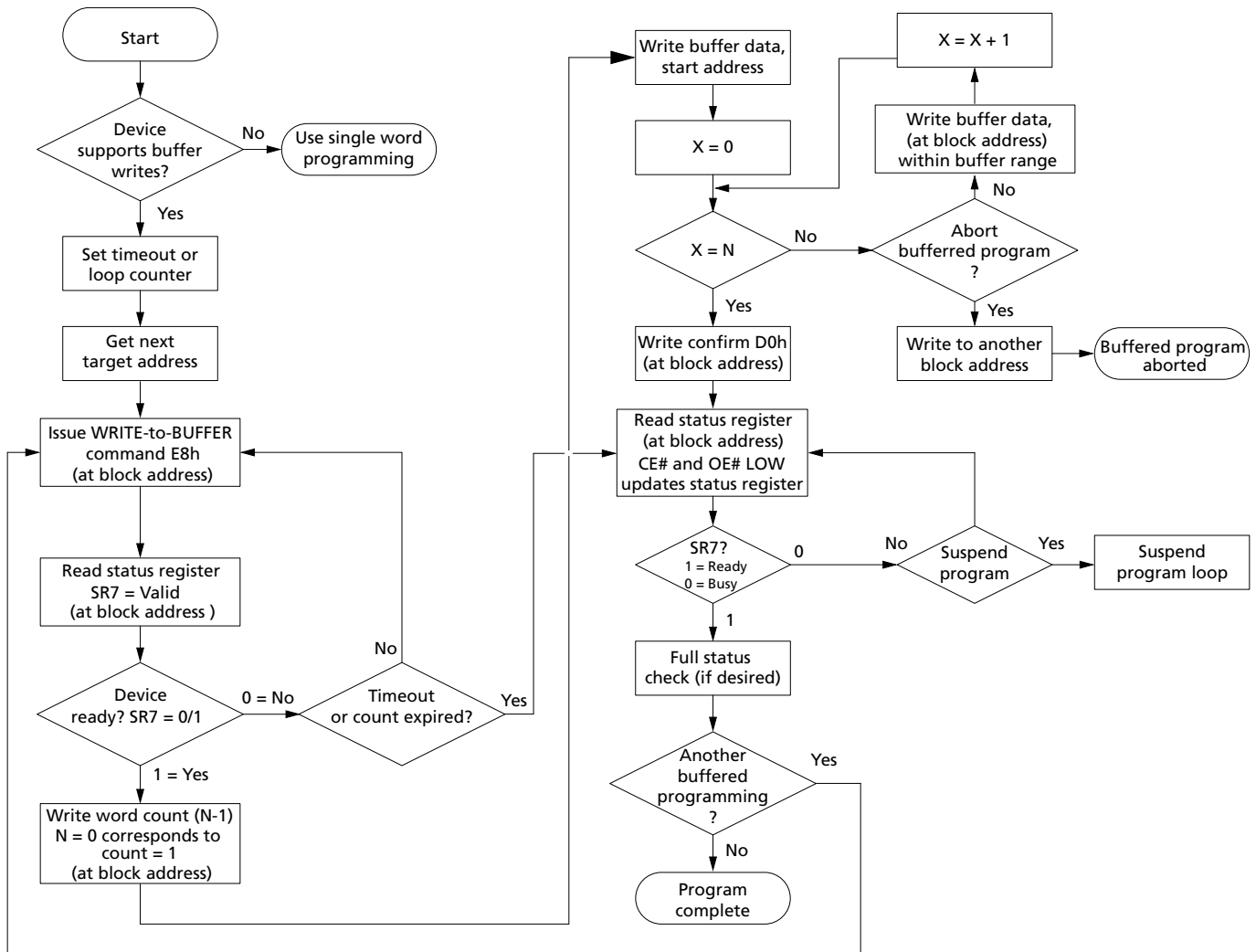


Figure 15: Buffer Program Procedure



- Notes:
1. Word count values on DQ0:DQ15 are loaded into the count register. Count ranges for this device are N = 0000h to 01FFh.
  2. Device outputs the status register when read.
  3. Write buffer contents will be programmed at the device start or destination address.
  4. Align the start address on a write buffer boundary for maximum programming performance; that is, A[9:1] of the start address = 0).
  5. Device aborts the BUFFERED PROGRAM command if the current address is outside the original block address.
  6. Status register indicates an improper command sequence if the BUFFERED PROGRAM command is aborted. Follow this with a CLEAR STATUS REGISTER command.
  7. Device defaults to SR output data after BUFFERED PROGRAMMING SETUP command (E8h) is issued. CE# or OE# must be toggled to update the status register. Don't issue the READ SR command (70h); it is interpreted by the device as buffer word count.
  8. Full status check can be done after erase and write sequences complete. Write FFh after the last operation to reset the device to read array mode.

Figure 16: Buffered Enhanced Factory Programming (BEFP) Procedure

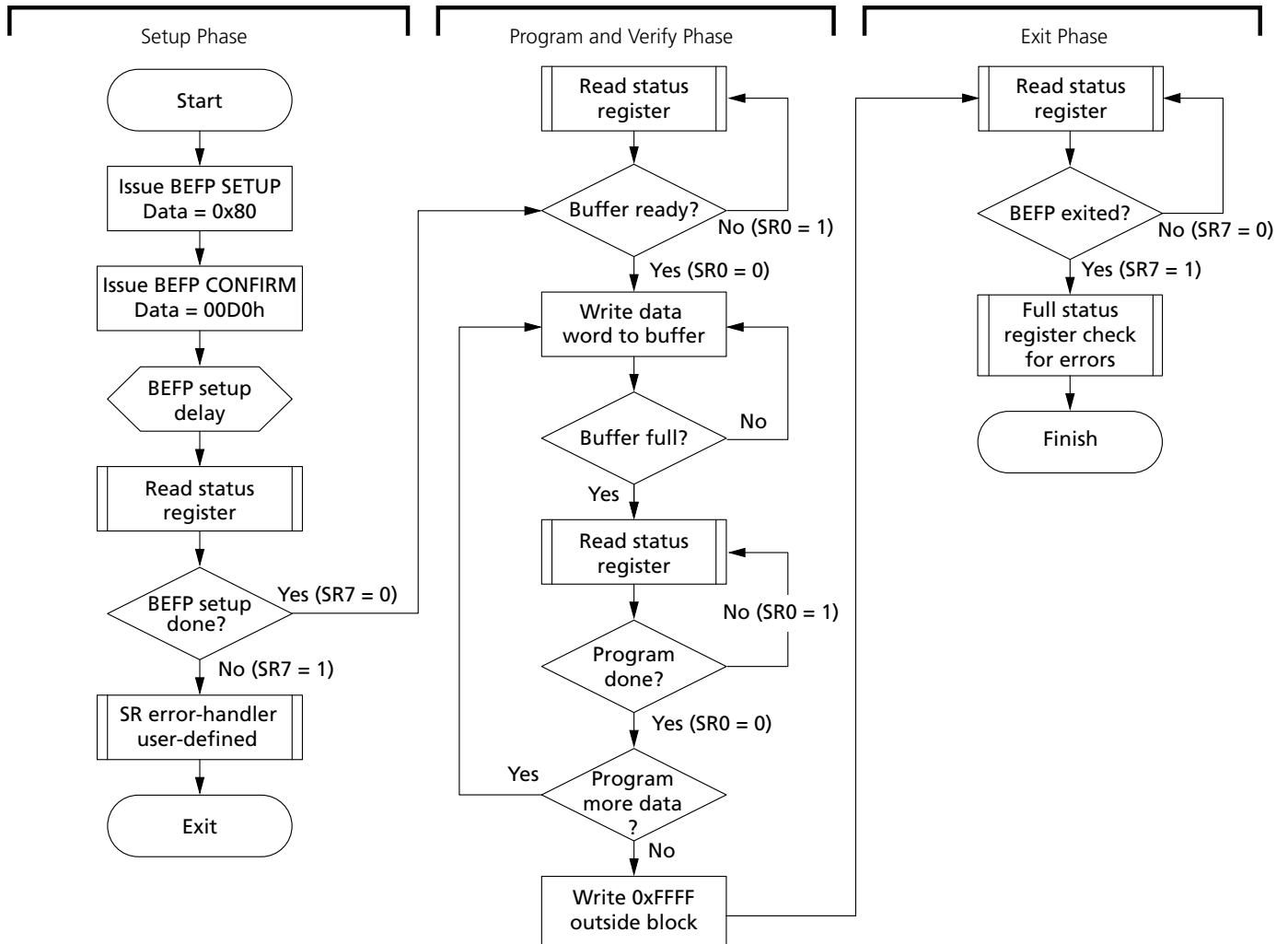




Figure 17: Block Erase Procedure

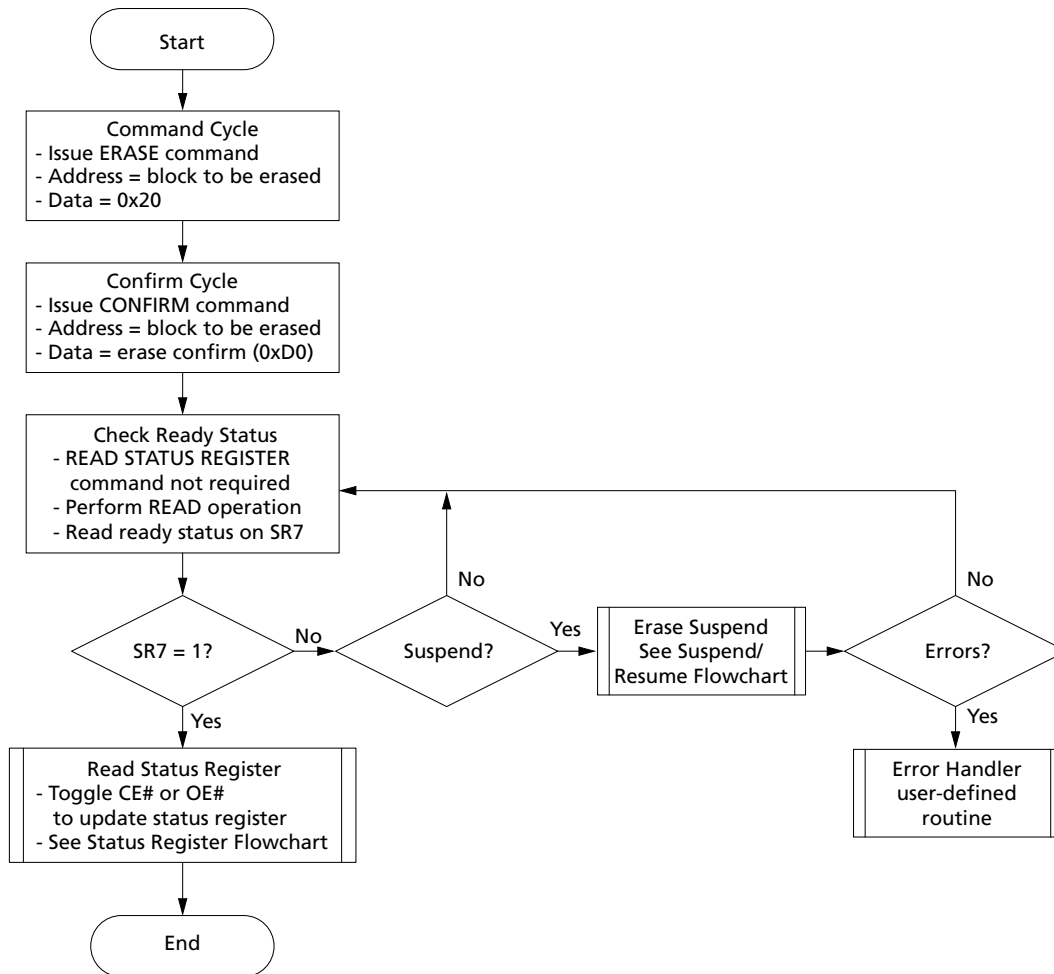


Figure 18: Program Suspend/Resume Procedure

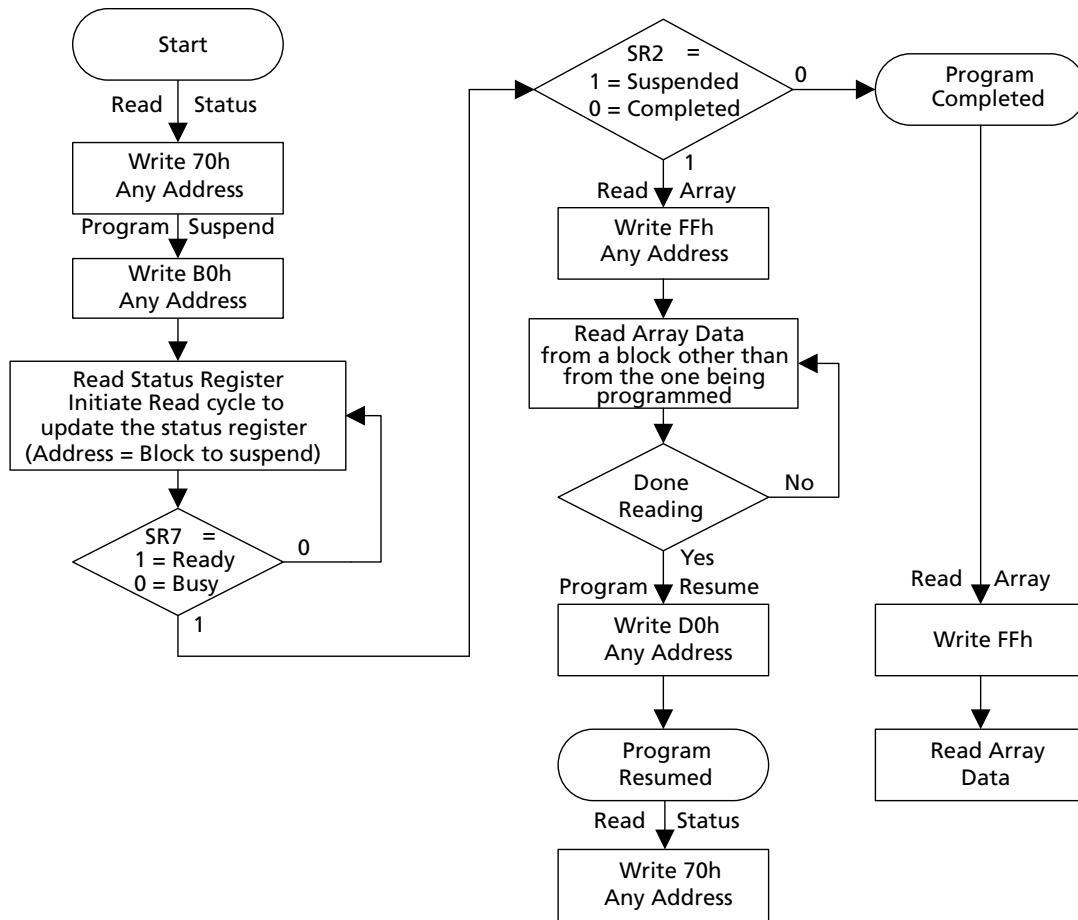
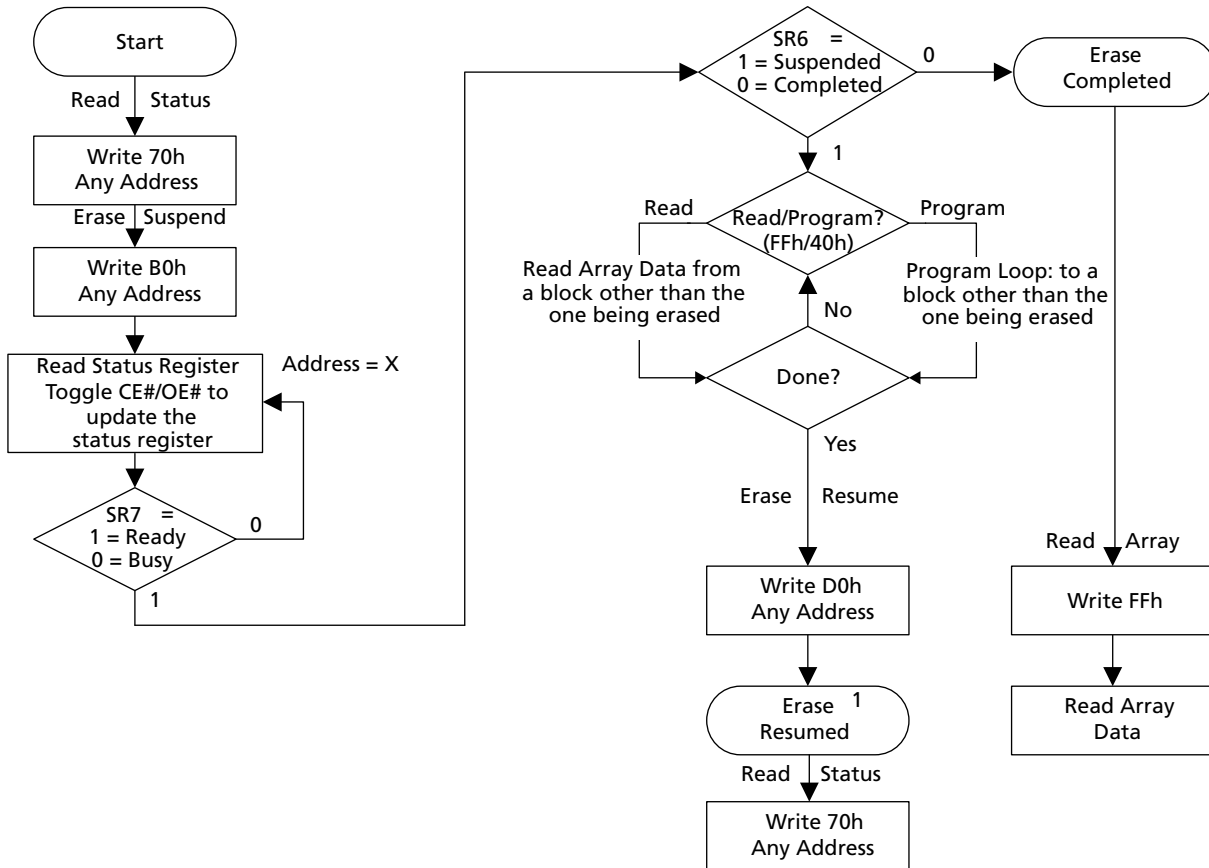


Figure 19: Erase Suspend/Resume Procedure



Note: 1. The  $t_{ERS/SUSP}$  timing between the initial BLOCK ERASE or ERASE RESUME command and a subsequent ERASE SUSPEND command should be followed.

Figure 20: Block Lock Operations Procedure

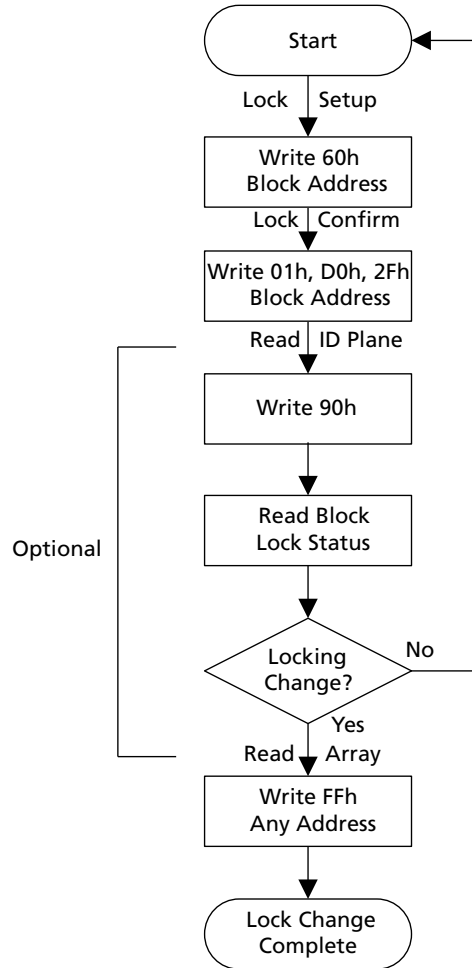


Figure 21: OTP Register Programming Procedure

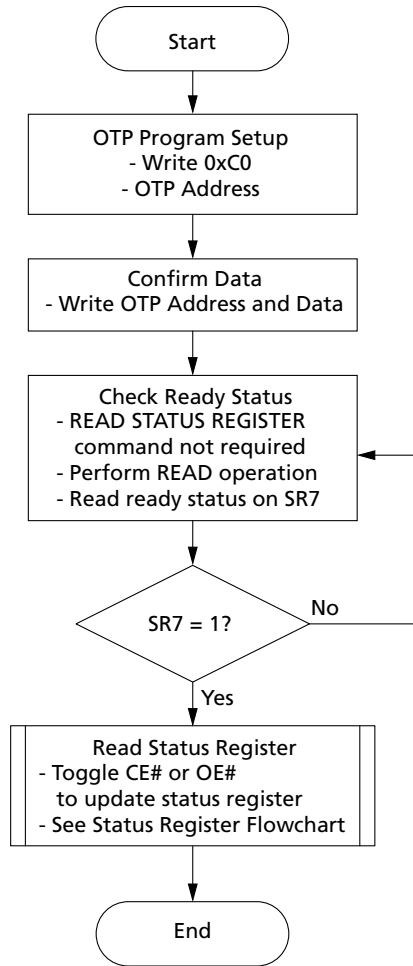
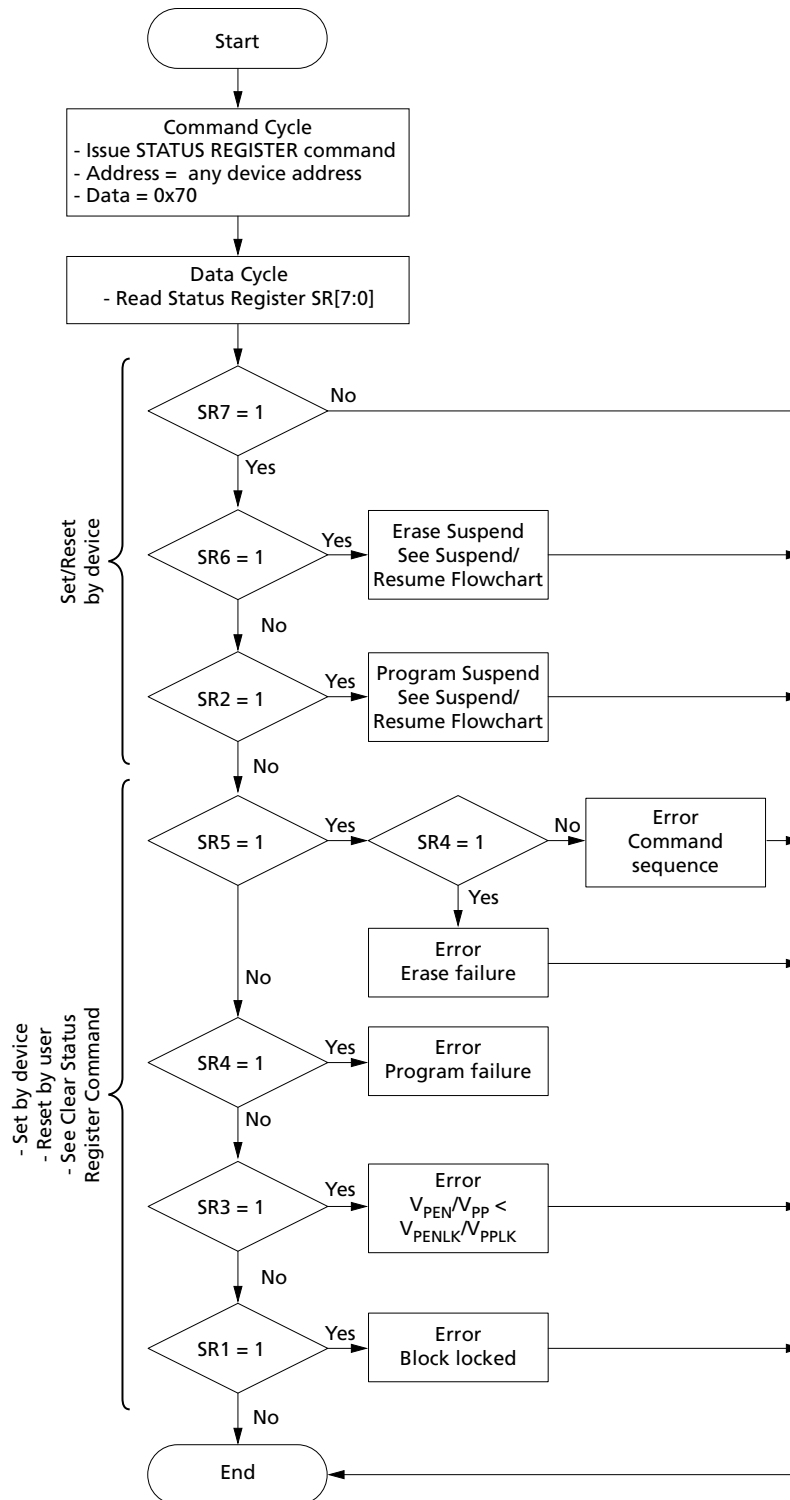


Figure 22: Status Register Procedure



## Power and Reset Specifications

$V_{CC}$  should attain  $V_{CCmin}$  from  $V_{SS}$  simultaneously with or before applying  $V_{CCQ}$ ,  $V_{PP}$  during power up.  $V_{CC}$  should attain  $V_{SS}$  during power down. Device inputs should not be driven before supply voltage =  $V_{CCmin}$ .

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

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

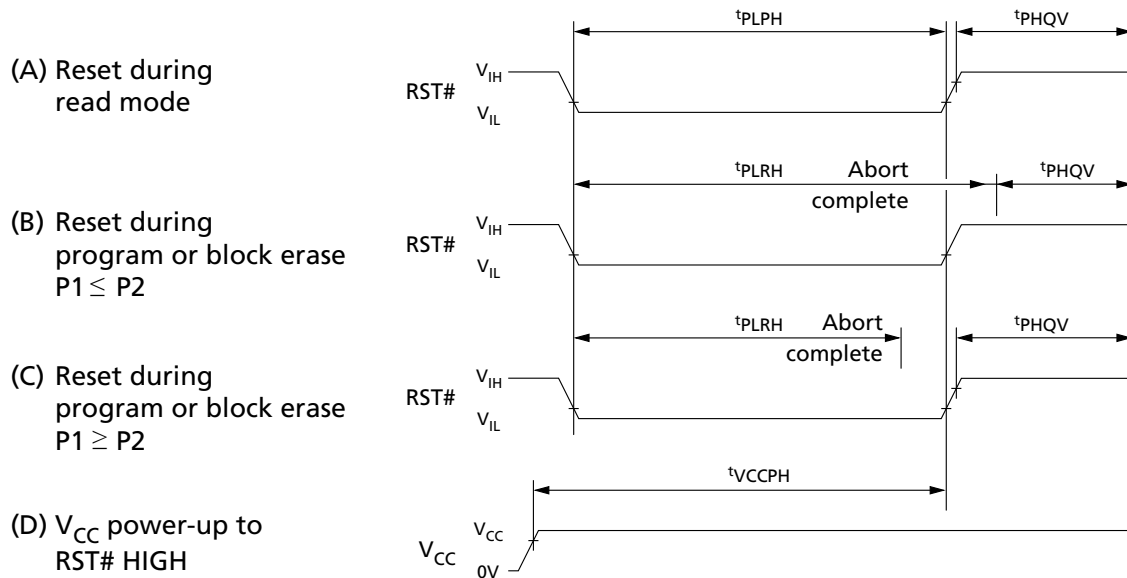
Because the device is disabled when RST# is asserted, it ignores its control inputs during power-up/down. Invalid bus conditions are masked, providing a level of memory protection.

**Table 32: Power and Reset**

| Parameter  | Symbol      | Min | Max | Unit | Notes      |
|--|-------------|-----|-----|------|------------|
| RST# pulse width LOW                             | $t_{PLPH}$  | 100 | –   | ns   | 1, 2, 3, 4 |
| RST# LOW to device reset during erase            | $t_{PLPH}$  | –   | 25  | us   | 1, 3, 4, 7 |
| RST# LOW to device reset during program          |             | –   | 25  |      |            |
| $V_{CC}$ Power valid to RST# de-assertion (HIGH) | $t_{VCCPH}$ | 300 | –   |      | 1, 4, 5, 6 |

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

**Figure 23: Reset Operation Waveforms**



## Power Supply Decoupling

The device requires careful power supply de-coupling. Three basic power supply current considerations are 1) standby current levels, 2) active current levels, and 3) transient peaks produced when CE# and OE# are asserted and de-asserted.

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

Because the devices draw their power from V<sub>CC</sub>, V<sub>PP</sub>, and V<sub>CCQ</sub>, each power connection should have a 0.1 $\mu$ F and a 0.01 $\mu$ F ceramic capacitor to ground. High-frequency, inherently low-inductance capacitors should be placed as close as possible to package leads.

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



## Maximum Ratings and Operating Conditions

Stresses greater than those listed can cause permanent damage to the device. This is stress rating only, and functional operation of the device at these or any other conditions above those indicated is not guaranteed.

**Table 33: Maximum Ratings**

| Parameter   | Maximum Rating    | Notes |
|---|-------------------|-------|
| Temperature under bias  | -40°C to + 85 °C  |       |
| Storage temperature   | -65°C to + 125 °C |       |
| Voltage on any signal (except $V_{CC}$ , $V_{PP}$ , and $V_{CCQ}$ ) | -2V to +5.6V      | 1     |
| $V_{PP}$ voltage  | -2V to +11.5V     | 1, 2  |
| $V_{CC}$ voltage  | -2V to +5.6V      | 1     |
| $V_{CCQ}$ voltage   | -2V to +5.6V      | 1     |
| Output short circuit current  | 100mA             | 3     |

- Notes:
1. Voltages shown are specified with respect to  $V_{SS}$ . During infrequent nonperiodic transitions, the level may undershoot to -2V for periods less than 20ns or overshoot to  $V_{CC} + 2V$  or  $V_{CCQ} + 2V$  or  $V_{PP} + 2V$  for periods less than 20ns.
  2. Program/erase voltage is typically 2.3–3.6V. 9.0V can be applied for 80 hours maximum total, to any blocks for 1000 cycles maximum. 9.0V program/erase voltage may reduce block cycling capability.
  3. Output is shorted for no more than one second, and more than one output is not shorted at one time.

**Table 34: Operating Conditions**

| Symbol                | Parameter                                      |                    | Min     | Max  | Unit  | Notes |
|-----------------------|--|--------------------|---------|------|-------|-------|
| $T_A$                 | Operating temperature                          |                    | -40     | +85  | °C    | 1     |
| $V_{CC}$              | $V_{CC}$ supply voltage                        |                    | 2.3     | 3.6  | V     |       |
| $V_{CCQ}$             | I/O supply voltage                             | CMOS inputs        | 2.3     | 3.6  |       |       |
|                       |  | TTL inputs         | 2.4     | 3.6  |       |       |
| $V_{PPL}$             | $V_{PP}$ voltage supply (logic level)          |                    | 1.5     | 3.6  | Hours | 2     |
| $V_{PPH}$             | Buffered enhanced factory programming $V_{PP}$ |                    | 8.5     | 9.5  |       |       |
| $t_{PPH}$             | Maximum $V_{PP}$ hours                         | $V_{PP} = V_{PPH}$ | –       | 80   |       |       |
| BLOCK<br>ERASE cycles | Array blocks                                   | $V_{PP} = V_{PPL}$ | 100,000 | –    |       |       |
|                       |  | $V_{PP} = V_{PPH}$ | –       | 1000 |       |       |

- Notes:
1.  $T_A$  = ambient temperature.
  2. In typical operation,  $V_{PP}$  program voltage is  $V_{PPL}$ .

## DC Electrical Specifications

**Table 35: DC Current Characteristics**

| Parameter   |   | Symbol                     | CMOS Inputs<br>( $V_{CCQ} = 2.3\text{--}3.6\text{V}$ ) |         | TTL Inputs<br>( $V_{CCQ} = 2.4\text{--}3.6\text{V}$ ) |          | Unit            | Test Conditions  | Notes  |   |
|---|---|----------------------------|--|---------|---|----------|-----------------|--|--|---|
|   |   |                            | Typ  | Max     | Typ   | Max      |                 |  |  |   |
| Input load current  | 512Mb<br>1Gb  | $I_{LI}$                   | –  | $\pm 1$ | –   | $\pm 2$  | $\mu\text{A}$   | $V_{CC} = V_{CC}(\text{MAX})$<br>$V_{CCQ} = V_{CCQ}(\text{MAX})$<br>$V_{IN} = V_{CCQ}$ or $V_{SS}$   | 1, 6   |   |
|   | 2Gb   |                            | –  | $\pm 2$ | –   | $\pm 4$  |                 |  |  |   |
| Output leakage current<br>DQ[15:0], WAIT                                    | 512Mb<br>1Gb  | $I_{LO}$                   | –  | $\pm 1$ | –   | $\pm 10$ | $\mu\text{A}$   | $V_{CC} = V_{CC}(\text{MAX})$<br>$V_{CCQ} = V_{CCQ}(\text{MAX})$<br>$V_{IN} = V_{CCQ}$ or $V_{SS}$   |  |   |
|   | 2Gb   |                            | –  | $\pm 2$ | –   | $\pm 20$ |                 |  |  |   |
| $V_{CC}$ standby,<br>power-down   | 512Mb   | $I_{CCS}$                  | 70   | 225     | 70  | 225      | $\mu\text{A}$   | $V_{CC} = V_{CC}(\text{MAX})$<br>$V_{CCQ} = V_{CCQ}(\text{MAX})$<br>$CE\# = V_{CCQ}$<br>$RST\# = V_{CCQ}$ (for $I_{CCS}$ )<br>$RST\# = V_{SS}$ (for $I_{CCD}$ )<br>$WP\# = V_{IH}$ | 1, 2   |   |
|   | 1Gb   | $I_{CCD}$                  | 75   | 240     | 75  | 240      |                 |  |  |   |
|   | 2Gb   |                            | 150  | 480     | 150   | 480      |                 |  |  |   |
| Average<br>$V_{CC}$ read<br>current   | Asynchronous<br>single-word<br>$f = 5\text{ MHz}$ (1 CLK) | $I_{CCR}$                  | 26   | 31      | 26  | 31       | mA              | 16-word read   | $V_{CC} = V_{CC}(\text{MAX})$<br>$CE\# = V_{IL}$<br>$OE\# = V_{IH}$<br>Inputs:<br>$V_{IL}$ or $V_{IH}$ | 1 |
|   |   |                            | 12   | 16      | 12  | 16       | mA              | 16-word read   |  |   |
|   | 19  |                            | 22   | 19      | 22  | mA       | 8-word read     |  |  |   |
|   | 16  |                            | 18   | 16      | 18  | mA       | 16-word read    |  |  |   |
|   | 21  |                            | 24   | 21      | 24  | mA       | Continuous read |  |  |   |
| $V_{CC}$ program current,<br>$V_{CC}$ erase current                         |   | $I_{CCW}$ ,<br>$I_{CCE}$   | 35   | 50      | 35  | 50       | mA              | $V_{PP} = V_{PPL}$ ,<br>program/erase in progress  | 1, 3, 5  |   |
|   |   |                            | 35   | 50      | 35  | 50       |                 | $V_{PP} = V_{PPH}$ ,<br>program/erase in progress  |  |   |
| $V_{CC}$ program sus-<br>pend current,<br>$V_{CC}$ erase suspend<br>current | 512Mb   | $I_{CCWS}$ ,<br>$I_{CCES}$ | 70   | 225     | 70  | 225      | $\mu\text{A}$   | $CE\# = V_{CCQ}$ , suspend in progress   | 1, 3, 4  |   |
|   | 1Gb<br>2Gb  |                            | 75   | 240     | 75  | 240      |                 |  |  |   |
| $V_{PP}$ standby current  | 512Mb   | $I_{PPS}$                  | 0.2  | 5       | 0.2   | 5        | $\mu\text{A}$   | $V_{PP} = V_{PPL}$ ,<br>in standby mode  | 1, 3, 7  |   |
|   | 1Gb   |                            | 0.2  | 5       | 0.2   | 5        |                 |  |  |   |
|   | 2Gb   |                            | 0.4  | 10      | 0.4   | 10       |                 |  |  |   |
| $V_{PP}$ program suspend current,<br>$V_{PP}$ erase suspend current         |   | $I_{PPWS}$ ,<br>$I_{PPES}$ | 0.2  | 5       | 0.2   | 5        | $\mu\text{A}$   | $V_{PP} = V_{PPL}$ ,<br>suspend in progress  | 1, 3, 7  |   |
| $V_{PP}$ read   |   | $I_{PPR}$                  | 2  | 15      | 2   | 15       | $\mu\text{A}$   | $V_{PP} = V_{PPL}$   | 1, 3   |   |
| $V_{PP}$ program current  |   | $I_{PPW}$                  | 0.05   | 0.1     | 0.05  | 0.1      | mA              | $V_{PP} = V_{PPL}$ , program in progress   | 3  |   |
|   |   |                            | 0.05   | 0.1     | 0.05  | 0.1      |                 | $V_{PP} = V_{PPH}$ , program in progress   |  |   |

**Table 35: DC Current Characteristics (Continued)**

| Parameter              | Symbol     | CMOS Inputs<br>( $V_{CCQ} = 2.3\text{--}3.6\text{V}$ ) |     | TTL Inputs<br>( $V_{CCQ} = 2.4\text{--}3.6\text{V}$ ) |     | Unit | Test Conditions                        | Notes |
|------------------------|------------|--|-----|---|-----|------|--|-------|
|                        |            | Typ  | Max | Typ   | Max |      |  |       |
| $V_{PP}$ erase current | $I_{PPE}$  | 0.05   | 0.1 | 0.05  | 0.1 | mA   | $V_{PP} = V_{PPL}$ , erase in progress | 3     |
|                        |            | 0.05   | 0.1 | 0.05  | 0.1 |      | $V_{PP} = V_{PPH}$ , erase in progress |       |
| $V_{PP}$ blank check   | $I_{PPBC}$ | 0.05   | 0.1 | 0.05  | 0.1 | mA   | $V_{PP} = V_{PPL}$                     | 3     |
|                        |            | 0.05   | 0.1 | 0.05  | 0.1 |      | $V_{PP} = V_{PPH}$                     |       |

- Notes:
1. All currents are RMS unless noted. Typical values at TYP  $V_{CC}$ ,  $T_C = +25^\circ\text{C}$ .
  2.  $I_{CCS}$  is the average current measured over any 5ms time interval  $5\mu\text{s}$  after CE# is de-asserted.
  3. Sampled, not 100% tested.
  4.  $I_{CCES}$  is specified with the device deselected. If device is read while in erase suspend, current is  $I_{CCES}$  plus  $I_{CCR}$ .
  5.  $I_{CCW}$ ,  $I_{CCE}$  measured over TYP or MAX times specified in (page 0 ).
  6. if  $V_{IN} > V_{CC}$ , the input load current increases to  $10\mu\text{A}$  MAX.
  7. the  $I_{PPS}$ ,  $I_{PPWS}$ ,  $I_{PPES}$  will increase to  $200\mu\text{A}$  when  $V_{PP}/WP\#$  is at  $V_{PPH}$ .

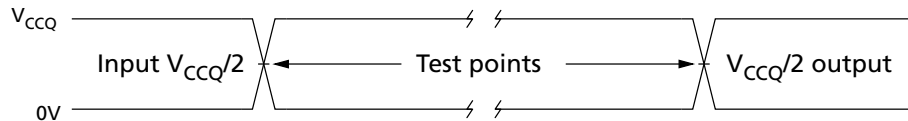
**Table 36: DC Voltage Characteristics**

| Parameter                 | Symbol     | CMOS Inputs<br>( $V_{CCQ} = 2.3\text{--}3.6\text{V}$ ) |                 | TTL Inputs <sup>1</sup><br>( $V_{CCQ} = 2.4\text{--}3.6\text{V}$ ) |                 | Unit | Test Conditions  | Notes |
|---------------------------|------------|--|-----------------|--|-----------------|------|--|-------|
|                           |            | Min  | Max             | Min  | Max             |      |  |       |
| Input low voltage         | $V_{IL}$   | -0.5   | 0.4             | -0.5   | 0.6             | V    |  | 2     |
| Input high voltage        | $V_{IH}$   | $V_{CCQ} - 0.4$  | $V_{CCQ} + 0.5$ | 2  | $V_{CCQ} + 0.5$ | V    |  |       |
| Output low voltage        | $V_{OL}$   | -  | 0.2             | -  | 0.2             | V    | $V_{CC} = V_{CC}(\text{MIN})$<br>$V_{CCQ} = V_{CCQ}(\text{MIN})$<br>$I_{OL} = 100\mu\text{A}$  |       |
| Output high voltage       | $V_{OH}$   | $V_{CCQ} - 0.2$  | -               | $V_{CCQ} - 0.2$  | -               | V    | $V_{CC} = V_{CC}(\text{MIN})$<br>$V_{CCQ} = V_{CCQ}(\text{MIN})$<br>$I_{OH} = -100\mu\text{A}$ |       |
| $V_{PP}$ lock out voltage | $V_{PPLK}$ | -  | 0.4             | -  | 0.4             | V    |  | 3     |

- Notes:
1. Synchronous read mode is not supported with TTL inputs.
  2.  $V_{IL}$  can undershoot to  $-1.0\text{V}$  for durations of 2ns or less and  $V_{IH}$  can overshoot to  $V_{CCQ} + 1.0\text{V}$  for durations of 2ns or less.
  3.  $V_{PP} \leq V_{PPLK}$  inhibits ERASE and PROGRAM operations. Do not use  $V_{PPL}$  and  $V_{PPH}$  outside their valid ranges.

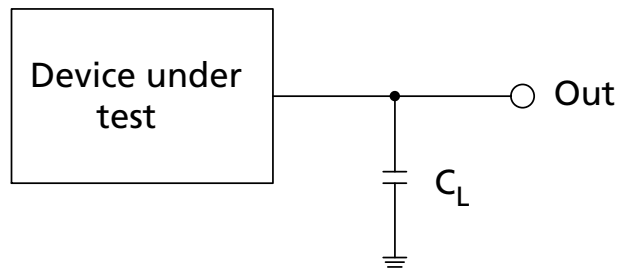
## AC Test Conditions and Capacitance

**Figure 24: AC Input/Output Reference Timing**



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

**Figure 25: Transient Equivalent Load Circuit**

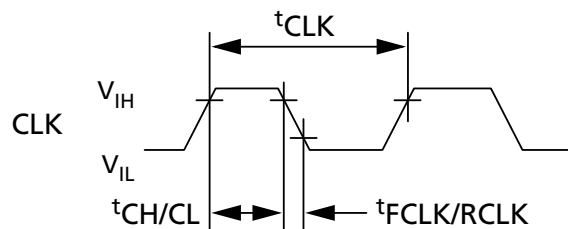


Notes: 1. See the Test Configuration for Worst-Case Speed Conditions table for component values.  
2.  $C_L$  includes jig capacitance.

**Table 37: Test Configuration: Worst-Case Speed Condition**

| Test Configuration                  | $C_L$ (pF) |
|-------------------------------------|------------|
| $V_{CCQ}(\text{MIN})$ standard test | 30         |

**Figure 26: Clock Input AC Waveform**



**Table 38: Capacitance**

| Parameter             | Sym-<br>bol      | Signal  | Density | Min | Typ | Max | Unit | Condition  | Notes |
|-----------------------|------------------|---|---------|-----|-----|-----|------|--|-------|
| Input<br>Capacitance  | C <sub>IN</sub>  | Address, Data,<br>CE#, WE#, OE#,<br>RST#, CLK,<br>ADV#, WP# | 512Mb   | 3   | 7   | 8   | pF   | TYP temp = 25°C; MAX<br>temp = 85°C<br>V <sub>CC</sub> = 0–2.0V, V <sub>CCQ</sub> = 0–<br>3.6V<br>Discrete silicon die | 1     |
|                       |                  |   | 1Gb     | 4   | 8   | 9   |      |  |       |
|                       |                  |   | 2Gb     | 6   | 16  | 18  |      |  |       |
| Output<br>Capacitance | C <sub>OUT</sub> | Data, WAIT  | 512Mb   | 3   | 5   | 7   |      |  |       |
|                       |                  |   | 1Gb     | 3   | 5   | 6   |      |  |       |
|                       |                  |   | 2Gb     | 6   | 10  | 12  |      |  |       |

Note: 1. Sampled, but not 100% tested.



## AC Read Specifications

Table 39: AC Read Specifications

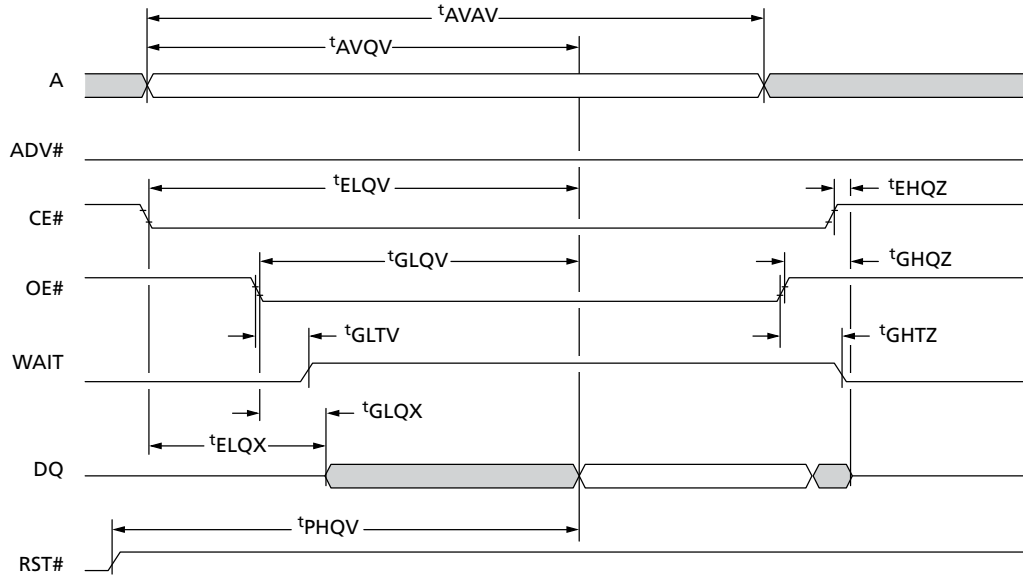
| Parameter  | Symbol     |          | Min       | Max | Unit | Notes   |   |
|--|------------|----------|-----------|-----|------|---------|---|
| <b>Asynchronous Specifications</b>                           |            |          |           |     |      |         |   |
| READ cycle time  | $t_{AVAV}$ | Easy BGA | 512Mb/1Gb | 95  | –    | ns      | – |
|  |            |          | 2Gb       | 100 | –    |         |   |
|  |            | TSOP     | 512Mb/1Gb | 105 | –    |         |   |
| Address to output valid                                      | $t_{AVQV}$ | Easy BGA | 512Mb/1Gb | –   | 95   | ns      | – |
|  |            |          | 2Gb       | –   | 100  |         |   |
|  |            | TSOP     | 512Mb/1Gb | –   | 105  |         |   |
| CE# LOW to output valid                                      | $t_{ELQV}$ | Easy BGA | 512Mb/1Gb | –   | 95   | ns      | – |
|  |            |          | 2Gb       | –   | 100  |         |   |
|  |            | TSOP     | 512Mb/1Gb | –   | 105  |         |   |
| OE# LOW to output valid                                      | $t_{GLQV}$ |          | –         | 25  | ns   | 1, 2    |   |
| RST# HIGH to output valid                                    | $t_{PHQV}$ |          | –         | 150 | ns   | 1       |   |
| CE# LOW to output in Low-Z                                   | $t_{ELQX}$ |          | 0         | –   | ns   | 1, 3    |   |
| OE# LOW to output in Low-Z                                   | $t_{GLQX}$ |          | 0         | –   | ns   | 1, 2, 3 |   |
| CE# HIGH to output in High-Z                                 | $t_{EHQZ}$ |          | –         | 20  | ns   | 1, 3    |   |
| OE# HIGH to output in High-Z                                 | $t_{GHQZ}$ |          | –         | 15  | ns   |         |   |
| Output hold from first occurring address, CE#, or OE# change | $t_{OH}$   |          | 0         | –   | ns   |         |   |
| CE# pulse width HIGH   | $t_{EHEL}$ |          | 17        | –   | ns   | 1       |   |
| CE# LOW to WAIT valid  | $t_{ELTV}$ |          | –         | 17  | ns   |         |   |
| CE# HIGH to WAIT High-Z                                      | $t_{EHTZ}$ |          | –         | 20  | ns   | 1, 3    |   |
| OE# LOW to WAIT valid  | $t_{GLTV}$ |          | –         | 17  | ns   | 1       |   |
| OE# LOW to WAIT in Low-Z                                     | $t_{GLTX}$ |          | 0         | –   | ns   | 1, 3    |   |
| OE# HIGH to WAIT in High-Z                                   | $t_{GHTZ}$ |          | –         | 20  | ns   |         |   |
| <b>Latching Specifications (Easy BGA)</b>                    |            |          |           |     |      |         |   |
| Address setup to ADV# HIGH                                   | $t_{AVVH}$ |          | 10        | –   | ns   | 1       |   |
| CE# LOW to ADV# HIGH   | $t_{ELVH}$ |          | 10        | –   | ns   |         |   |
| ADV# LOW to output valid                                     | $t_{VLQV}$ | Easy BGA | 512Mb/1Gb | –   | 95   | ns      |   |
|  |            |          | 2Gb       | –   | 100  |         |   |
|  |            | TSOP     | 512Mb/1Gb | –   | 105  |         |   |
| ADV# pulse width LOW   | $t_{VLVH}$ |          | 10        | –   | ns   |         |   |
| ADV# pulse width HIGH  | $t_{VHVL}$ |          | 10        | –   | ns   |         |   |
| Address hold from ADV# HIGH                                  | $t_{VHAX}$ |          | 9         | –   | ns   | 1, 4    |   |
| Page address access  | $t_{APA}$  |          | –         | 25  | ns   | 1       |   |
| RST# HIGH to ADV# HIGH                                       | $t_{PHVH}$ |          | 30        | –   | ns   |         |   |
| <b>Clock Specifications (Easy BGA)</b>                       |            |          |           |     |      |         |   |

**Table 39: AC Read Specifications (Continued)**

| Parameter  | Symbol                                   |  | Min  | Max | Unit | Notes      |
|--|--|--|------|-----|------|------------|
| CLK frequency  | $t^{\text{CLK}}$                         |  | –    | 52  | MHz  | 1, 3, 5, 6 |
| CLK period   | $t^{\text{CLK}}$                         |  | 19.2 | –   | ns   |            |
| CLK HIGH/LOW time  | $t^{\text{CH/CL}}$                       |  | 5    | –   | ns   |            |
| CLK fall/rise time                                       | $t^{\text{FCLK/RCLK}}$                   |  | 0.3  | 3   | ns   |            |
| <b>Synchronous Specifications (Easy BGA)<sup>5</sup></b> |  |  |      |     |      |            |
| Address setup to CLK                                     | $t^{\text{AVCH/L}}$                      |  | 9    | –   | ns   | 1, 6       |
| ADV# LOW setup to CLK                                    | $t^{\text{VLCH/L}}$                      |  | 9    | –   | ns   |            |
| CE# LOW setup to CLK                                     | $t^{\text{ELCH/L}}$                      |  | 9    | –   | ns   |            |
| CLK to output valid                                      | $t^{\text{CHQV}}$ /<br>$t^{\text{CLQV}}$ |  | –    | 17  | ns   |            |
| Output hold from CLK                                     | $t^{\text{CHQX}}$                        |  | 3    | -   | ns   | 1, 6       |
| Address hold from CLK                                    | $t^{\text{CHAX}}$                        |  | 10   | -   | ns   | 1, 4, 6    |
| CLK to WAIT valid  | $t^{\text{CHTV}}$                        |  | –    | 17  | ns   | 1, 6       |
| CLK valid to ADV# setup                                  | $t^{\text{CHVL}}$                        |  | 3    | –   | ns   | 1          |
| WAIT hold from CLK                                       | $t^{\text{CHTX}}$                        |  | 3    | –   | ns   | 1, 6       |

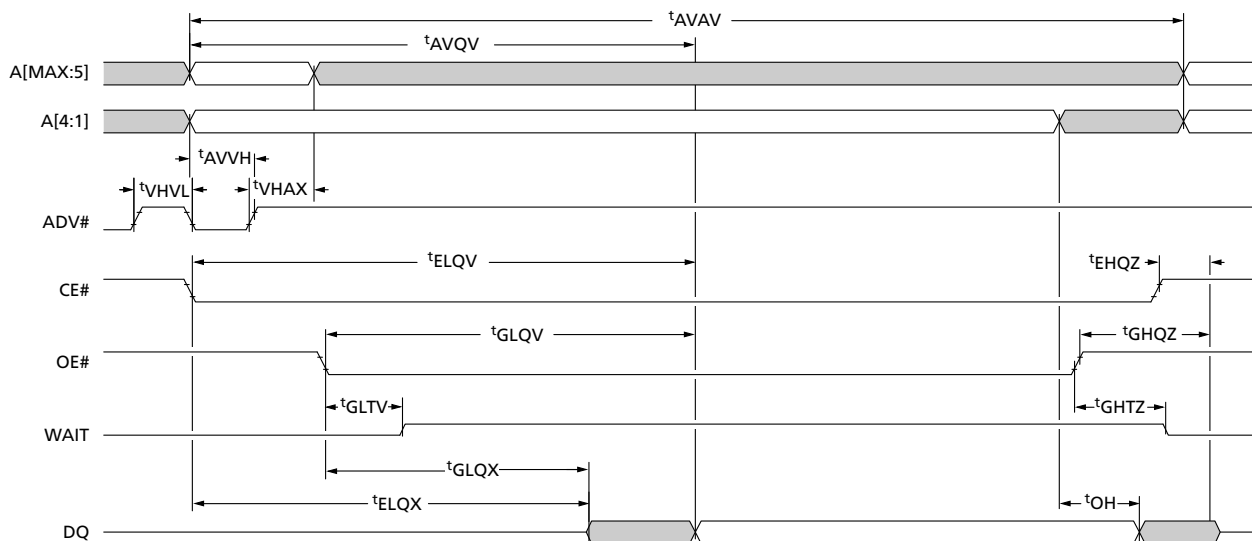
- Notes:
1. See AC Test Conditions for timing measurements and maximum allowable input slew rate.
  2. OE# may be delayed by up to  $t^{\text{ELQV}} - t^{\text{GLQV}}$  after CE#'s falling edge without impact to  $t^{\text{ELQV}}$ .
  3. Sampled, not 100% tested.
  4. Address hold in synchronous burst mode is  $t^{\text{CHAX}}$  or  $t^{\text{VHAX}}$ , whichever timing specification is satisfied first.
  5. Synchronous read mode is not supported with TTL level inputs.
  6. Applies only to subsequent synchronous reads.

Figure 27: Asynchronous Single-Word Read (ADV# LOW)



Note: 1. WAIT shown deasserted during asynchronous read mode (RCR10 = 0, WAIT asserted LOW).

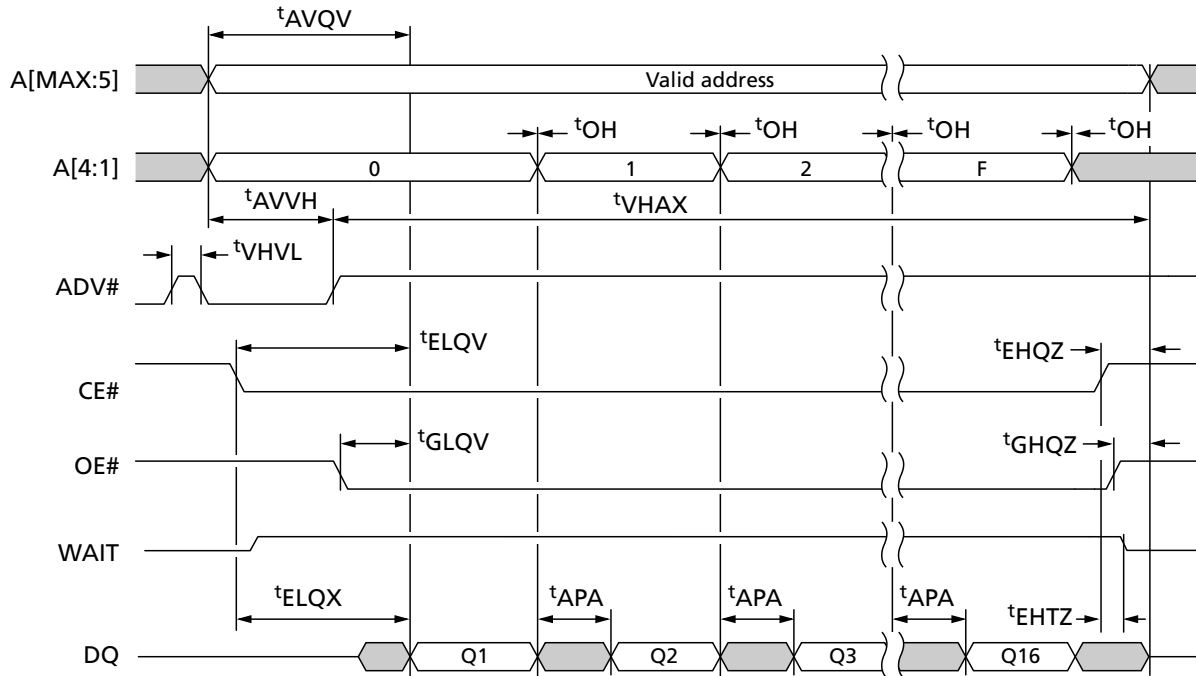
Figure 28: Asynchronous Single-Word Read (ADV# Latch)



Note: 1. WAIT shown deasserted during asynchronous read mode (RCR10 = 0, WAIT asserted LOW).

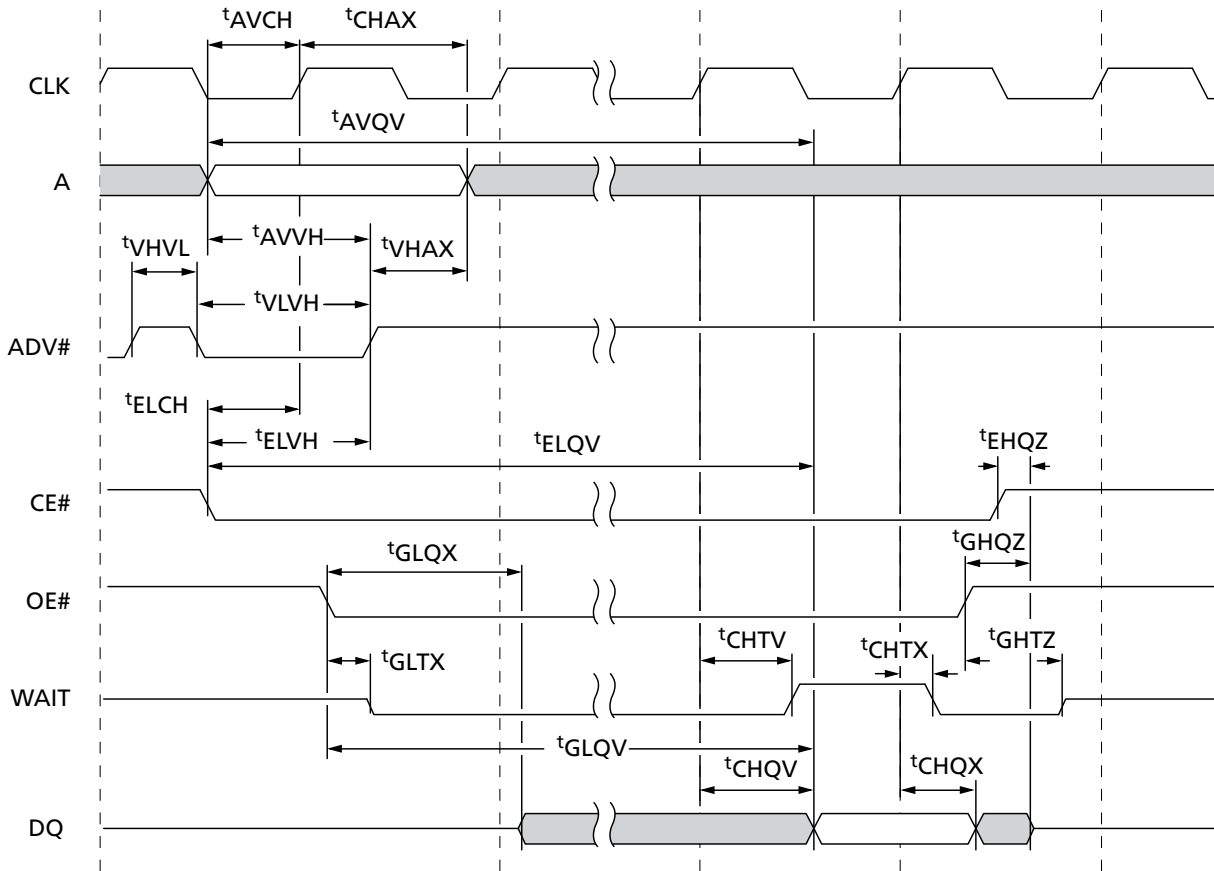


Figure 29: Asynchronous Page Mode Read



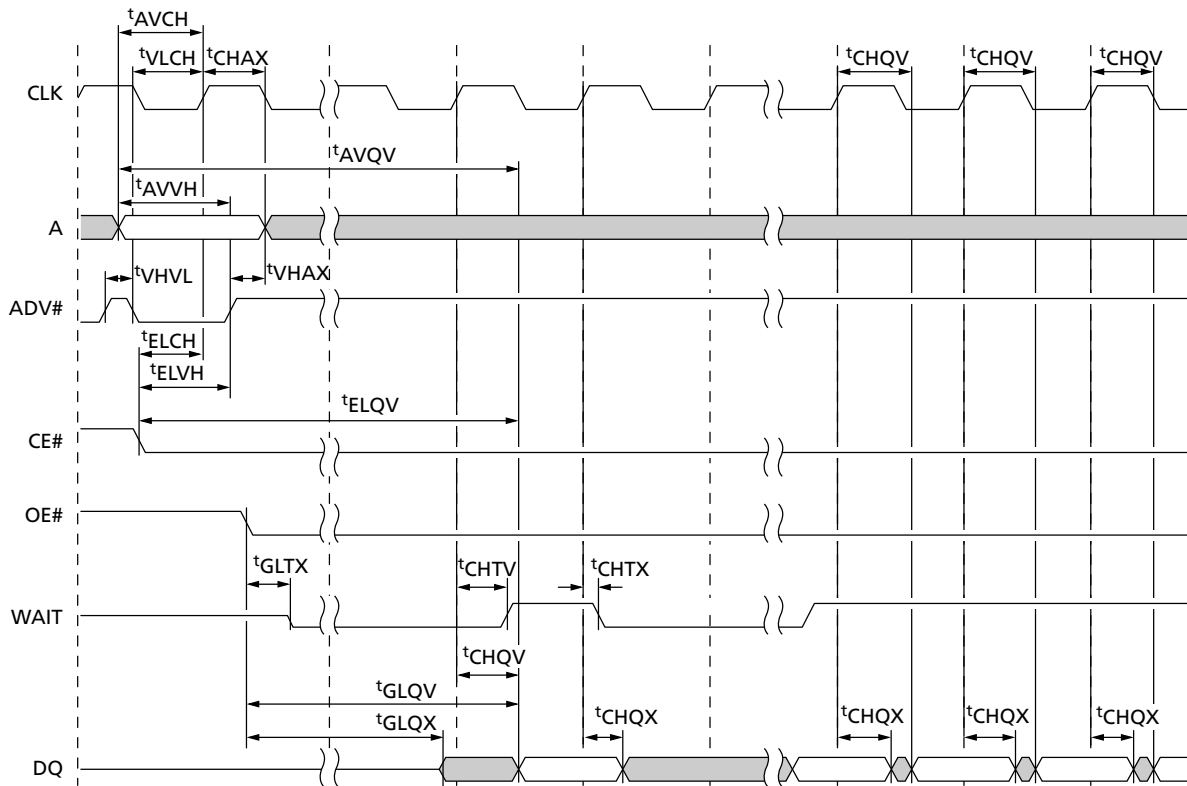
Note: 1. WAIT shown deasserted during asynchronous read mode (RCR10 = 0, WAIT asserted LOW).

Figure 30: Synchronous Single-Word Array or Nonarray Read



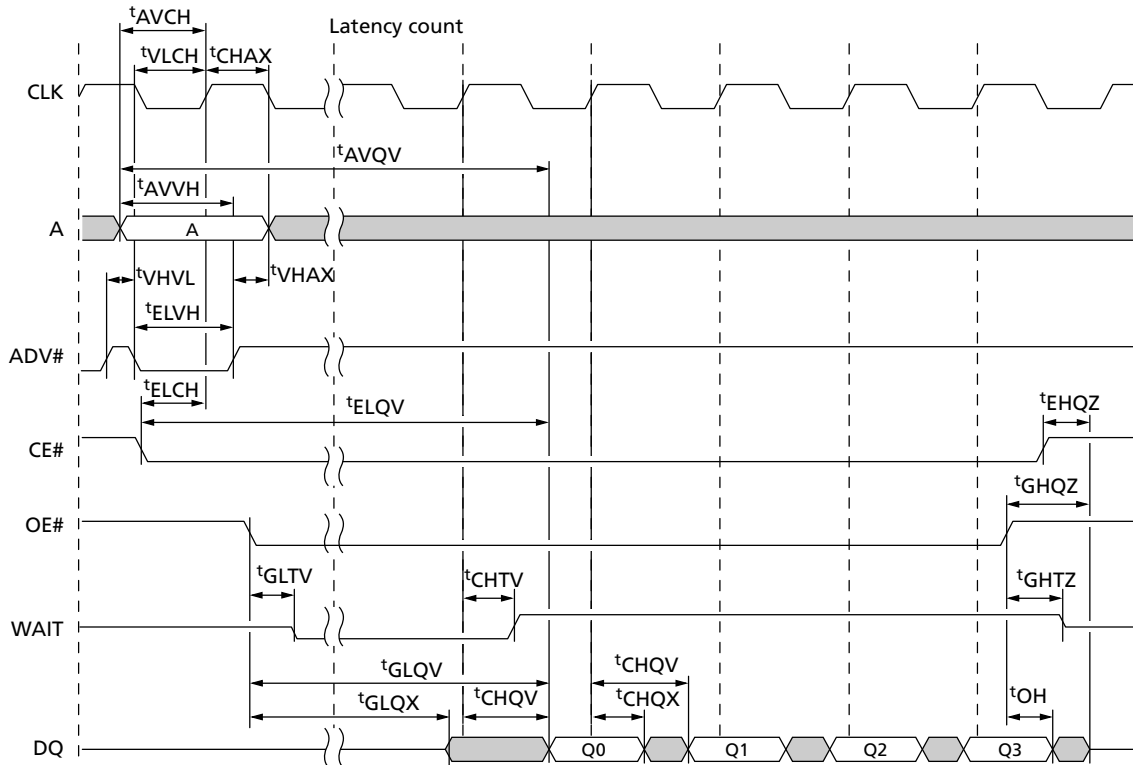
- Notes:
1. WAIT is driven per OE# assertion during synchronous array or nonarray read and can be configured to assert either during or one data cycle before valid data.
  2. In this example, an  $n$ -word burst is initiated to the flash memory array and is terminated by CE# deassertion after the first word in the burst.

Figure 31: Continuous Burst Read with Output Delay



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

Figure 32: Synchronous Burst Mode 4-Word Read



Note: 1. WAIT is driven per OE# assertion during synchronous array or nonarray read. WAIT asserted during initial latency and deasserted during valid data (RCR10 = 0, WAIT asserted LOW).

## AC Write Specifications

**Table 40: AC Write Specifications**

| Parameter  | Symbol       | Min             | Max | Unit | Notes          |
|--|--------------|-----------------|-----|------|----------------|
| RST# HIGH recovery to WE# LOW                    | $t_{PHWL}$   | 150             | -   | ns   | 1, 2, 3        |
| CE# setup to WE# LOW                             | $t_{ELWL}$   | 0               | -   | ns   | 1, 2, 3        |
| WE# write pulse width LOW                        | $t_{WLWH}$   | 50              | -   | ns   | 1, 2, 4        |
| Data setup to WE# HIGH                           | $t_{DVWH}$   | 50              | -   | ns   | 1, 2, 12       |
| Address setup to WE# HIGH                        | $t_{AVWH}$   | 50              | -   | ns   | 1, 2           |
| CE# hold from WE# HIGH                           | $t_{WHEH}$   | 0               | -   | ns   |                |
| Data hold from WE# HIGH                          | $t_{WHDX}$   | 0               | -   | ns   |                |
| Address hold from WE# HIGH                       | $t_{WHAX}$   | 0               | -   | ns   |                |
| WE# pulse width HIGH                             | $t_{WHWL}$   | 20              | -   | ns   | 1, 2, 5        |
| $V_{PP}$ setup to WE# HIGH                       | $t_{VPWH}$   | 200             | -   | ns   | 1, 2, 3, 7     |
| $V_{PP}$ hold from status read                   | $t_{QVVL}$   | 0               | -   | ns   |                |
| WP# hold from status read                        | $t_{QVBL}$   | 0               | -   | ns   | 1, 2, 3, 7     |
| WP# setup to WE# HIGH                            | $t_{BHWH}$   | 200             | -   | ns   |                |
| WE# HIGH to OE# LOW                              | $t_{WHGL}$   | 0               | -   | ns   | 1, 2, 9        |
| WE# HIGH to read valid                           | $t_{WHQV}$   | $t_{AVQV} + 35$ | -   | ns   | 1, 2, 3, 6, 10 |
| <b>Write to Asynchronous Read Specifications</b> |              |                 |     |      |                |
| WE# HIGH to address valid                        | $t_{WHAV}$   | 0               | -   | ns   | 1, 2, 3, 6, 8  |
| <b>Write to Synchronous Read Specifications</b>  |              |                 |     |      |                |
| WE# HIGH to clock valid                          | $t_{WHCH/L}$ | 19              | -   | ns   | 1, 2, 3, 6, 10 |
| WE# HIGH to ADV# HIGH                            | $t_{WHVH}$   | 19              | -   | ns   |                |
| WE# HIGH to ADV# LOW                             | $t_{WHVL}$   | 7               | -   | ns   |                |
| <b>Write Specification with Clock Active</b>     |              |                 |     |      |                |
| ADV# HIGH to WE# LOW                             | $t_{VHWL}$   | -               | 20  | ns   | 1, 2, 3, 11    |
| Clock HIGH to WE# LOW                            | $t_{CHWL}$   | -               | 20  | ns   |                |

- Notes:
1. Write timing characteristics during erase suspend are the same as WRITE-only operations.
  2. A WRITE operation can be terminated with either CE# or WE#.
  3. Sampled, not 100% tested.
  4. Write pulse width LOW ( $t_{WLWH}$  or  $t_{ELEH}$ ) is defined from CE# or WE# LOW (whichever occurs last) to CE# or WE# HIGH (whichever occurs first). Thus,  $t_{WLWH} = t_{ELEH} = t_{WLEH} = t_{ELWH}$ .
  5. Write pulse width HIGH ( $t_{WHWL}$  or  $t_{EHLE}$ ) is defined from CE# or WE# HIGH whichever occurs first) to CE# or WE# LOW whichever occurs last). Thus,  $t_{WHWL} = t_{EHLE} = t_{WHEL} = t_{EHWL}$ .
  6.  $t_{WHVH}$  or  $t_{WHCH/L}$  must be met when transitioning from a WRITE cycle to a synchronous BURST read.
  7.  $V_{PP}$  and WP# should be at a valid level until erase or program success is determined.
  8. This specification is only applicable when transitioning from a WRITE cycle to an asynchronous read. See spec  $t_{WHCH/L}$  and  $t_{WHVH}$  for synchronous read.

9. When doing a READ STATUS operation following any command that alters the status register,  $t_{WHGL}$  is 20ns.
10. Add 10ns if the WRITE operation results in an RCR or block lock status change, for the subsequent READ operation to reflect this change.
11. These specs are required only when the device is in a synchronous mode and the clock is active during an address setup phase.
12. This specification must be complied with customer's writing timing. The result would be unpredictable if there is any violation to this timing specification.

Figure 33: Write to Write Timing

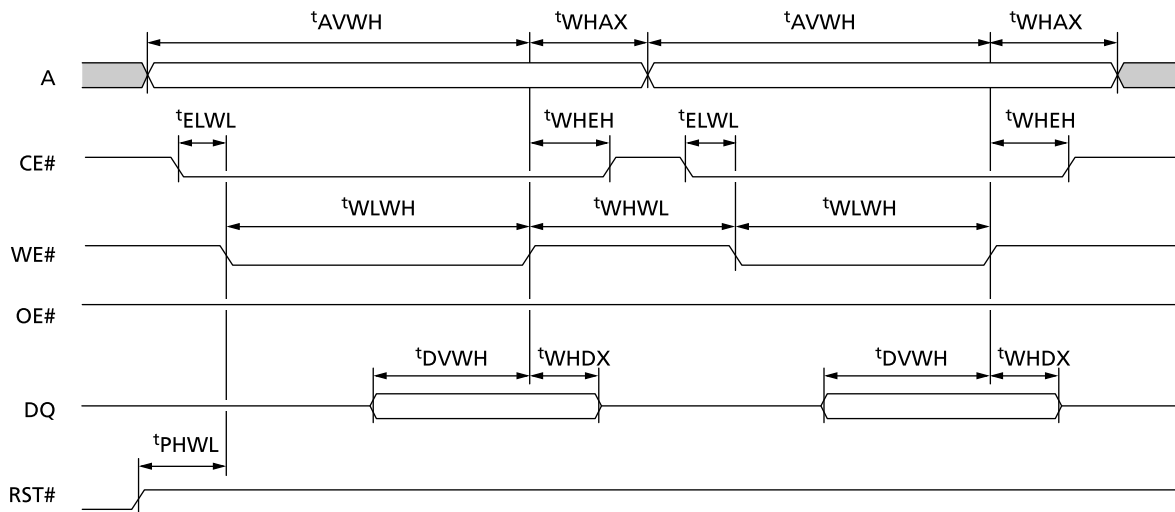
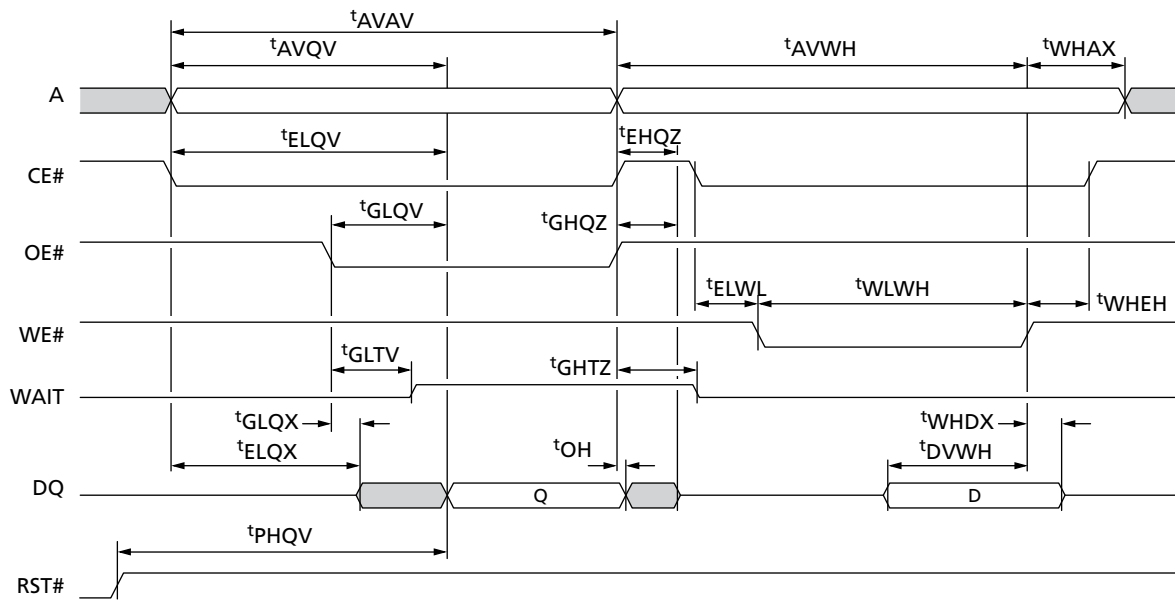


Figure 34: Asynchronous Read to Write Timing



Note: 1. WAIT de-asserted during asynchronous read and during write. WAIT High-Z during write per OE# deasserted.

Figure 35: Write to Asynchronous Read Timing

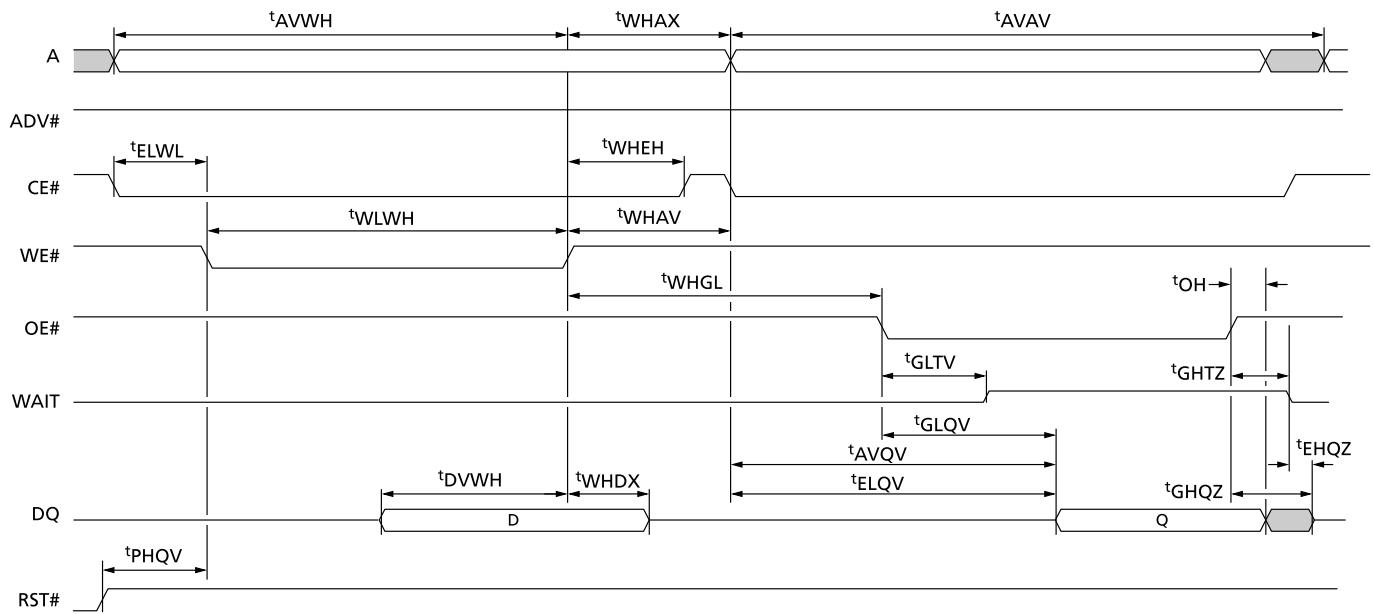
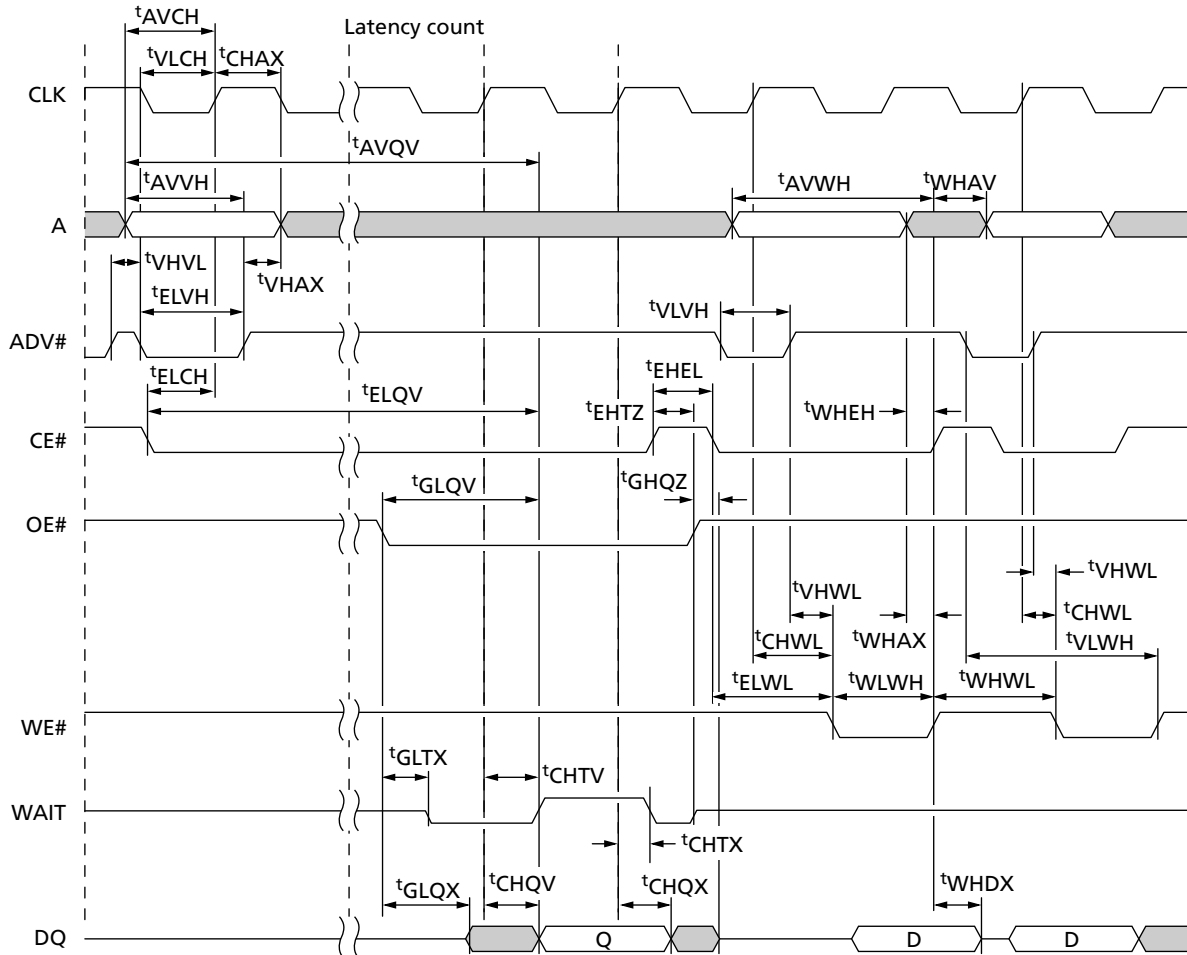


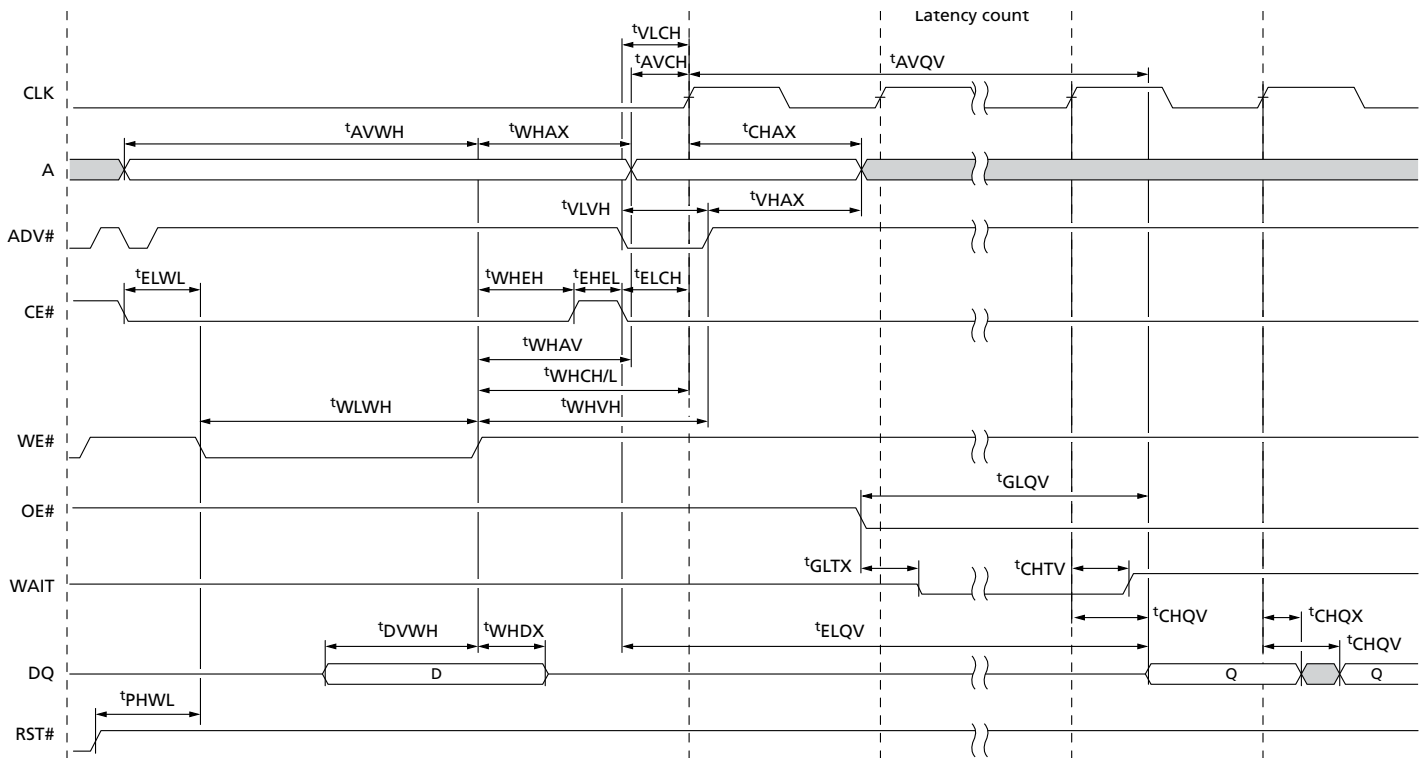


Figure 36: Synchronous Read to Write Timing



Note: 1. WAIT shown de-asserted and High-Z per OE# de-assertion during WRITE operation (RCR10 = 0, WAIT asserted LOW). Clock is ignored during WRITE operation.

Figure 37: Write to Synchronous Read Timing



Note: 1. WAIT shown de-asserted and High-Z per OE# de-assertion during WRITE operation (RCR10 = 0, WAIT asserted LOW).

## Program and Erase Characteristics

**Table 41: Program and Erase Specifications**

| Parameter                                    |                                      | Symbol                  | V <sub>PPL</sub> |     |      | V <sub>PPH</sub> |     |      | Unit | Notes |
|--|--------------------------------------|-------------------------|------------------|-----|------|------------------|-----|------|------|-------|
|  |                                      |                         | Min              | Typ | Max  | Min              | Typ | Max  |      |       |
| <b>Conventional Word Programming</b>         |                                      |                         |                  |     |      |                  |     |      |      |       |
| Program time                                 | Single word                          | <sup>t</sup> PROG/W     | –                | 270 | 456  | –                | 270 | 456  | μs   | 1     |
| <b>Buffered Programming</b>                  |                                      |                         |                  |     |      |                  |     |      |      |       |
| Program time                                 | Aligned, BP time (32 words)          | <sup>t</sup> PROG       | –                | 310 | 716  | –                | 310 | 716  | μs   | 1     |
|  | Aligned, BP time (64 words)          |                         | –                | 310 | 900  | –                | 310 | 900  |      |       |
|  | Aligned, BP time (128 words)         |                         | –                | 375 | 1140 | –                | 375 | 1140 |      |       |
|  | Aligned, BP time (256 words)         |                         | –                | 505 | 1690 | –                | 505 | 1690 |      |       |
|  | One full buffer, BP time (512 words) |                         | –                | 900 | 3016 | –                | 900 | 3016 |      |       |
| <b>Buffered Enhanced Factory Programming</b> |                                      |                         |                  |     |      |                  |     |      |      |       |
| Program                                      | Single byte                          | <sup>t</sup> BEFP/B     | N/A              | N/A | N/A  | –                | 0.5 | –    | μs   | 1, 2  |
|  | BEFP Setup                           | <sup>t</sup> BEFP/SETUP | N/A              | N/A | N/A  | 20               | –   | –    |      | 1     |
| <b>Erase and Suspend</b>                     |                                      |                         |                  |     |      |                  |     |      |      |       |
| Erase time                                   | 32KB parameter                       | <sup>t</sup> ERS/PB     | –                | 0.8 | 4.0  | –                | 0.8 | 4.0  | s    | 1     |
|  | 128KB main                           | <sup>t</sup> ERS/MB     | –                | 0.8 | 4.0  | –                | 0.8 | 4.0  |      |       |
| Suspend latency                              | Program suspend                      | <sup>t</sup> SUSP/P     | –                | 25  | 30   | –                | 25  | 30   | μs   | 1, 3  |
|  | Erase suspend                        | <sup>t</sup> SUSP/E     | –                | 25  | 30   | –                | 25  | 30   |      |       |
|  | Erase-to-suspend                     | <sup>t</sup> ERS/SUSP   | –                | 500 | –    | –                | 500 | –    |      |       |
| <b>Blank Check</b>                           |                                      |                         |                  |     |      |                  |     |      |      |       |
| Blank check                                  | Main array block                     | <sup>t</sup> BC/MB      | –                | 3.2 | –    | –                | 3.2 | –    | ms   |       |

- Notes:
1. Typical values measured at T<sub>C</sub> = +25°C and nominal voltages. Performance numbers are valid for all speed versions. Excludes system overhead. Sampled, but not 100% tested.
  2. Averaged over entire device.
  3. <sup>t</sup>ERS/SUSP is the typical time between an initial BLOCK ERASE or ERASE RESUME command and the a subsequent ERASE SUSPEND command. Violating the specification repeatedly during any particular block erase may cause erase failures.

## **Revision History**

### **Rev. C – 12/13**

- On cover page, corrected erase suspend (TYP) from 30 $\mu$ s to 25 $\mu$ s.
- Revised timings

### **Rev. B – 11/13**

- Added the following part number disclaimer: "Not all part numbers listed here are available for ordering."

### **Rev. A – 8/13**

- Initial Micron brand release

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900  
[www.micron.com/productsupport](http://www.micron.com/productsupport) Customer Comment Line: 800-932-4992  
Micron and the Micron logo are trademarks of Micron Technology, Inc.  
All other trademarks are the property of their respective owners.

This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein. Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.