# S29GL064S

# 64-Mbit (8 Mbyte), 3.0 V, Flash Memory

## Distinctive Characteristics

- CMOS 3.0 Volt Core with Versatile I/O

### Architectural Advantages

- Single Power Supply Operation
- Manufactured on 65 nm MirrorBit Process Technology
- Secure Silicon Region
  - 128-word/256-byte sector for permanent, secure identification through an 8-word / 16-byte random Electronic Serial Number, accessible through a command sequence
  - Programmed and locked at the factory or by the customer
- Flexible Sector Architecture
  - 64 Mb (uniform sector models): One hundred twenty-eight 32-kword (64-kB) sectors
  - 64 Mb (boot sector models): One hundred twenty-seven 32-kword (64-kB) sectors + eight 4kword (8kB) boot sectors
- Automatic Error Checking and Correction (ECC) - internal hardware ECC with single bit error correction
- Enhanced VersatileI/O Control
  - All input levels (address, control, and DQ input levels) and outputs are determined by voltage on $V_{IO}$ input. $V_{IO}$ range is 1.65 to $V_{CC}$
- Compatibility with JEDEC Standards
  - Provides pinout and software compatibility for single-power supply flash, and superior inadvertent write protection
- 100,000 Erase Cycles per Sector Minimum
- 20-year Data Retention Typical

### Performance Characteristics

- High Performance
  - 70 ns access time
  - 8-word / 16-byte page read buffer
  - 15 ns page read time
  - 128-word / 256-byte write buffer which reduces overall programming time for multiple-word updates
- Low Power Consumption
  - 25 mA typical initial read current @ 5 MHz
  - 7.5 mA typical page read current @ 33 MHz
  - 50 mA typical erase / program current
  - 40 µA typical standby mode current
- Package Options
  - 48-pin TSOP
  - 56-pin TSOP
  - 64-ball Fortified BGA (LAA064 13 mm × 11 mm × 1.4 mm) (LAE064 9 mm × 9 mm × 1.4 mm)
  - 48-ball fine-pitch BGA (VBK048 8.15 mm × 6.15 mm × 1.0 mm)
- Temperature Range
  - Industrial (–40°C to +85°C)
  - Industrial Plus (–40°C to +105°C)
  - Automotive, AEC-Q100 Grade 3 (–40°C to +85°C)
  - Automotive, AEC-Q100 Grade 2(–40°C to +105°C)

### Software and Hardware Features

- Software Features
  - Advanced Sector Protection: offers Persistent Sector Protection and Password Sector Protection
  - Program Suspend and Resume: read other sectors before programming operation is completed
  - Erase Suspend and Resume: read / program other sectors before an erase operation is completed
  - Data# polling and toggle bits provide status
  - CFI (Common Flash Interface) compliant: allows host system to identify and accommodate multiple flash devices
  - Unlock Bypass Program command reduces overall multiple-word programming time
- Hardware Features
  - WP#/ACC input supports manufacturing programming operations (when high voltage is applied). Protects first or last sector regardless of sector protection settings on uniform sector models
  - Hardware reset input (RESET#) resets device
  - Ready/Busy# output (RY/BY#) detects program or erase cycle completion



Downloaded from Arrow.com.

## General Description

The S29GL-S mid density family of devices are 3.0-volt single-power flash memory manufactured using 65 nm MirrorBit technology. The S29GL064S is a 64-Mb device organized as 4,194,304 words or 8,388,608 bytes. Depending on the model number, the devices have 16-bit wide data bus only, or a 16-bit wide data bus that can also function as an 8-bit wide data bus by using the BYTE# input. The devices can be programmed either in the host system or in standard EPROM programmers.

Access times as fast as 70 ns are available. Note that each access time has a specific operating voltage range ($V_{CC}$) as specified in the Product Selector Guide and Ordering Information. Package offerings include 48-pin TSOP, 56-pin TSOP, 48-ball fine-pitch BGA, and 64-ball Fortified BGA, depending on model number. Each device has separate chip enable (CE#), write enable (WE#) and output enable (OE#) controls.

Each device requires only a **single 3.0-volt power supply** for both read and write functions. In addition to a $V_{CC}$ input, a high-voltage **accelerated program (ACC)** feature is supported through increased voltage on the WP#/ACC or ACC input. This feature is intended to facilitate system production.

The device is entirely command set compatible with the **JEDEC single-power-supply flash standard**. Commands are written to the device using standard microprocessor write timing. Write cycles also internally latch addresses and data needed for the programming and erase operations.

The **sector erase architecture** allows memory sectors to be erased and reprogrammed without affecting the data contents of other sectors. The device is fully erased when shipped from the factory.

The **Advanced Sector Protection** features several levels of sector protection, which can disable both the program and erase operations in certain sectors. Persistent Sector Protection is a method that replaces the previous 12-volt controlled protection method. Password Sector Protection is a highly sophisticated protection method that requires a password before changes to certain sectors are permitted.

Device programming and erasure are initiated through command sequences. Once a program or erase operation begins, the host system need only poll the DQ7 (Data# Polling) or DQ6 (toggle) **status bits** or monitor the **Ready/Busy# (RY/BY#)** output to determine whether the operation is complete. To facilitate programming, an **Unlock Bypass** mode reduces command sequence overhead by requiring only two write cycles to program data instead of four.

**Hardware data protection** measures include a low $V_{CC}$ detector that automatically inhibits write operations during power transitions. The hardware sector protection feature disables both program and erase operations in any combination of sectors of memory. This can be achieved in-system or via programming equipment.

The **Erase Suspend / Erase Resume** feature allows the host system to pause an erase operation in a given sector to read or program any other sector and then complete the erase operation. The **Program Suspend / Program Resume** feature enables the host system to pause a program operation in a given sector to read any other sector and then complete the program operation.

The **hardware RESET# pin** terminates any operation in progress and resets the device, after which it is then ready for a new operation. The RESET# pin may be tied to the system reset circuitry. A system reset would thus also reset the device, enabling the host system to read boot-up firmware from the flash memory device.

The device reduces power consumption in the **standby mode** when it detects specific voltage levels on CE# and RESET#, or when addresses are stable for a specified period of time.

The **Write Protect (WP#)** feature protects the first or last sector by asserting a logic low on the WP#/ACC pin or WP# pin, depending on model number. The protected sector is still protected even during accelerated programming.

The **Secure Silicon Region** provides a 128-word / 256-byte area for code or data that can be permanently protected. Once this sector is protected, no further changes within the sector can occur.

Cypress MirrorBit flash technology combines years of flash memory manufacturing experience to produce the highest levels of quality, reliability and cost effectiveness. The device electrically erases all bits within a sector simultaneously via hot-hole assisted erase. The data is programmed using hot electron injection.



Downloaded from Arrow.com.

# Contents


**1. Product Selector Guide** ..... 4
**2. Block Diagram** ..... 5
**3. Connection Diagrams** ..... 6
**4. Pin Description** ..... 9
**5. S29GL064S Logical Symbols** ..... 10
**6. Ordering Information** ..... 11
6.1 Valid Combinations ..... 12
**7. Other Resources** ..... 13
7.1 Cypress Flash Memory Roadmap ..... 13
7.2 Links to Software ..... 13
7.3 Links to Application Notes ..... 13
**8. Device Bus Operations** ..... 14
8.1 Word / Byte Configuration ..... 14
8.2 Requirements for Reading Array Data ..... 15
8.3 Writing Commands / Command Sequences ..... 15
8.4 Automatic ECC ..... 16
8.5 Standby Mode ..... 17
8.6 Automatic Sleep Mode ..... 17
8.7 RESET#: Hardware Reset Pin ..... 17
8.8 Output Disable Mode ..... 18
8.9 Memory Map ..... 18
8.10 Autoselect Mode ..... 19
8.11 Advanced Sector Protection ..... 20
8.12 Lock Register ..... 21
8.13 Persistent Sector Protection ..... 22
8.14 Password Sector Protection ..... 24
8.15 Password and Password Protection Mode Lock Bit ..... 24
8.16 Persistent Protection Bit Lock (PPB Lock Bit) ..... 24
8.17 Secure Silicon Region Flash Memory ..... 25
8.18 Write Protect (WP#/ACC) ..... 26
8.19 Hardware Data Protection ..... 26
**9. Common Flash Memory Interface (CFI)** ..... 27
**10. Command Definitions** ..... 30
10.1 Reading Array Data ..... 30
10.2 Reset Command ..... 30
10.3 Autoselect Command Sequence ..... 31
10.4 Status Register ASO ..... 31
10.5 Enter / Exit Secure Silicon Region Command Sequence ..... 31
10.6 ECC Status ASO ..... 31
10.7 Word Program Command Sequence ..... 32
10.8 Unlock Bypass Command Sequence ..... 32
10.9 Write Buffer Programming ..... 33
10.10 Accelerated Program ..... 34
10.11 Program Suspend / Program Resume Command Sequence ..... 36
10.12 Chip Erase Command Sequence ..... 38
10.13 Sector Erase Command Sequence ..... 38
10.14 Erase Suspend / Erase Resume Commands ..... 39
10.15 Evaluate Erase Status ..... 40
10.16 Continuity Check ..... 40
10.17 Command Definitions ..... 42
**11. Data Integrity** ..... 49
11.1 Erase Endurance ..... 49
11.2 Data Retention ..... 49
**12. Status Monitoring** ..... 50
12.1 Status Register ..... 50
12.2 Write Operation Status ..... 51
12.3 DQ7: Data# Polling ..... 52
12.4 DQ6: Toggle Bit I ..... 54
12.5 DQ2: Toggle Bit II ..... 56
12.6 Reading Toggle Bits DQ6/DQ2 ..... 56
12.7 DQ5: Exceeded Timing Limits ..... 56
12.8 DQ3: Sector Erase Timer ..... 56
12.9 DQ1: Write-to-Buffer Abort ..... 57
12.10 RY/BY#: Ready/Busy# ..... 57
12.11 Error Types and Clearing Procedures ..... 57
**13. Command State Transitions** ..... 61
**14. Electrical Specifications** ..... 74
14.1 Absolute Maximum Ratings ..... 74
14.2 Latchup Characteristics ..... 74
14.3 Thermal Resistance ..... 74
14.4 Operating Ranges ..... 74
**15. DC Characteristicst** ..... 77
15.1 Capacitance Characteristics ..... 79
**16. Test Specifications** ..... 81
16.1 Key to Switching Waveforms ..... 81
16.2 AC Test Conditions ..... 81
16.3 Power-On Reset (POR) and Warm Reset ..... 82
**17. AC Characteristics** ..... 84
17.1 Read-Only Operations ..... 84
17.2 Asynchronous Write Operations ..... 88
17.3 Alternative CE# Controlled Write Operations ..... 94
**18. Erase and Programming Performance** ..... 97
**19. Physical Dimensions** ..... 99
19.1 TS048—48-Pin Standard Thin Small Outline Package (TSOP) ..... 99
19.2 TS056—56-Pin Standard Thin Small Outline Package (TSOP) ..... 100
19.3 VBK048—Ball Fine-pitch Ball Grid Array (BGA) 8.15 x 6.15 mm Package ..... 101
19.4 LAA064—64-Ball Fortified Ball Grid Array (BGA) 13 x 11 mm Package ..... 102
19.5 LAE064—64-Ball Fortified Ball Grid Array (BGA) 9 x 9 mm Package ..... 103
**20. Revision History** ..... 104




Downloaded from Arrow.com.

# 1. Product Selector Guide

**Table 1. Product Selector Guide for Industrial Temperature Range (–40°C to +85°C)**

| Part Number | | | S29GL064S | |
|---|---|---|---|---|
| **Speed Option** | $V_{CC}$ = 2.7–3.6V | $V_{IO}$ = 2.7–3.6V | 70 | |
| | | $V_{IO}$ = 1.65–3.6V | | 80 |
| Max. Access Time (ns) | | | 70 | 80 |
| Max. CE# Access Time (ns) | | | 70 | 80 |
| Max. Page Access Time (ns) | | | 15 | 25 |
| Max. OE# Access Time (ns) | | | 15 | 25 |

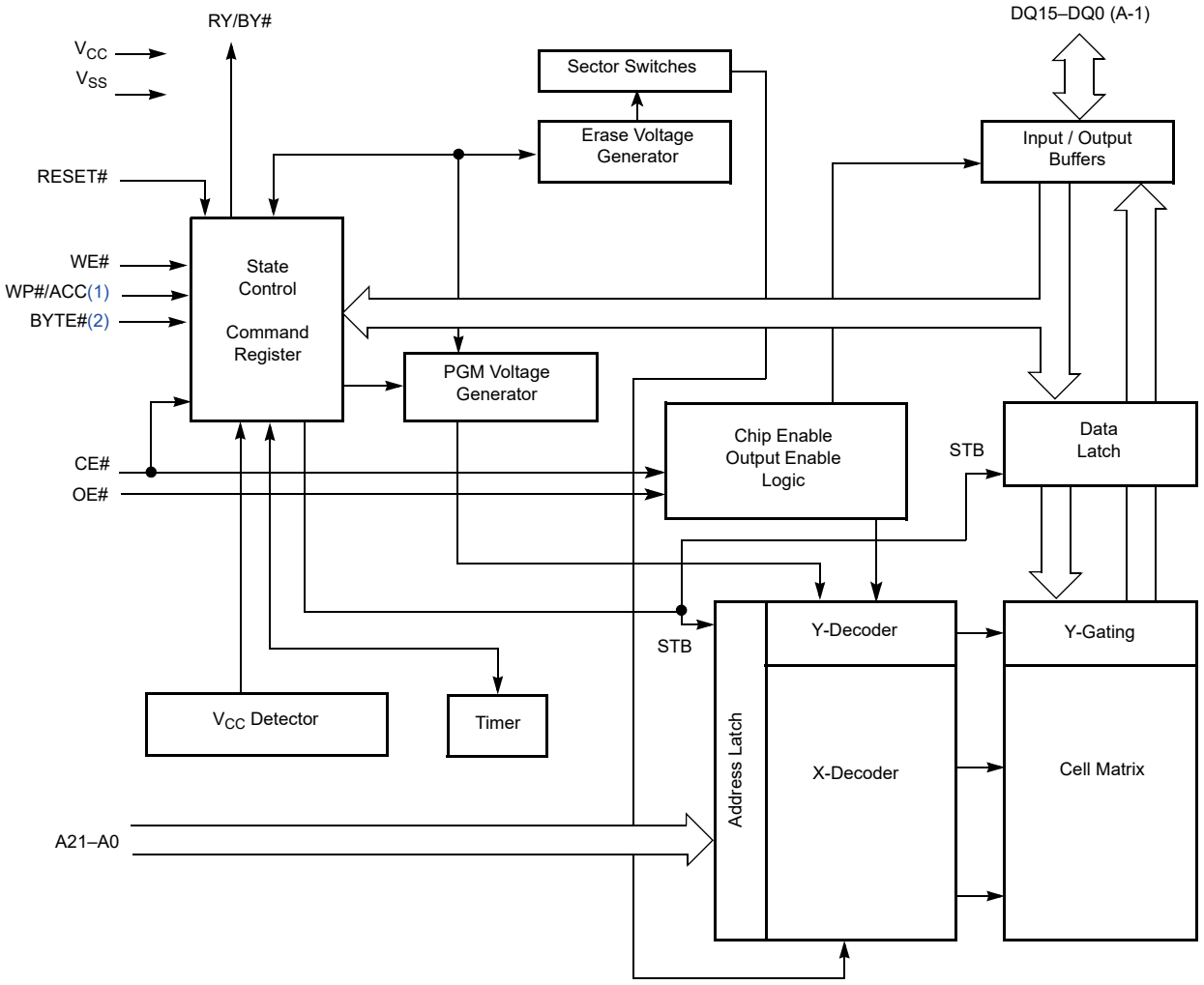
**Table 2. Product Selector Guide for Industrial Plus Temperature Range (–40°C to +105°C)**

| Part Number | | | S29GL064S | |
|---|---|---|---|---|
| **Speed Option** | $V_{CC}$ = 2.7–3.6V | $V_{IO}$ = 2.7–3.6V | 80 | |
| | | $V_{IO}$ = 1.65–3.6V | | 90 |
| Max. Access Time (ns) | | | 80 | 90 |
| Max. CE# Access Time (ns) | | | 80 | 90 |
| Max. Page Access Time (ns) | | | 15 | 25 |
| Max. OE# Access Time (ns) | | | 15 | 25 |



Downloaded from Arrow.com.

## 2. Block Diagram




***Notes:***

1. *Available on separate pins for models 06, 07, V6, V7.*
2. *Available only on X8/x16 devices.*



Downloaded from Arrow.com.

# 3. Connection Diagrams

### Special Package Handling Instructions

Special handling is required for flash memory products in molded packages (TSOP and BGA). The package and/or data integrity may be compromised if the package body is exposed to temperatures above 150°C for prolonged periods of time.

**Figure 1. 48-Pin Standard TSOP**

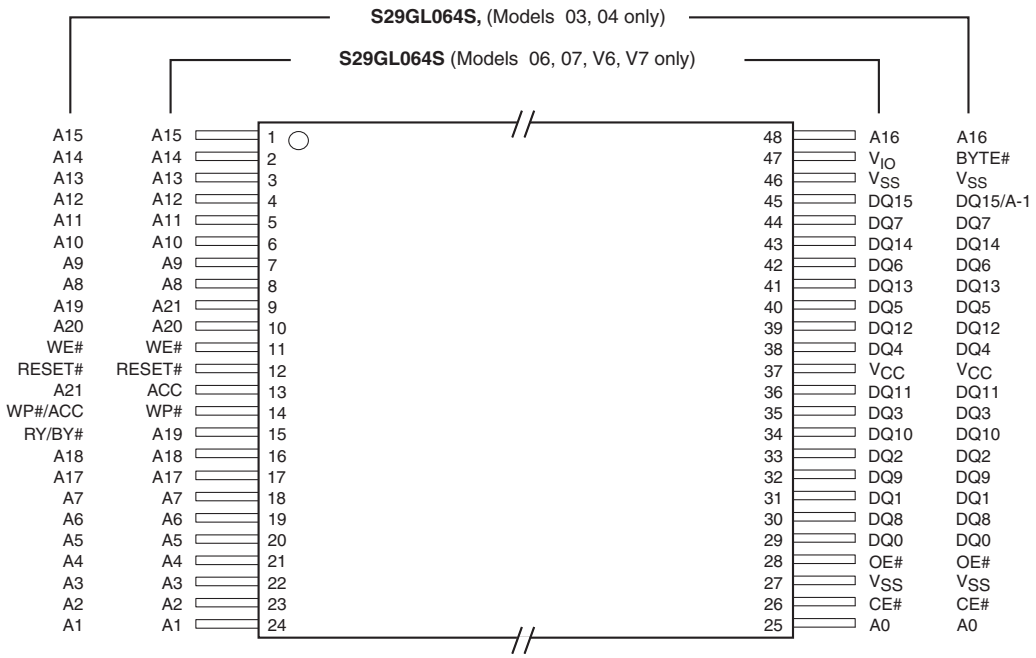
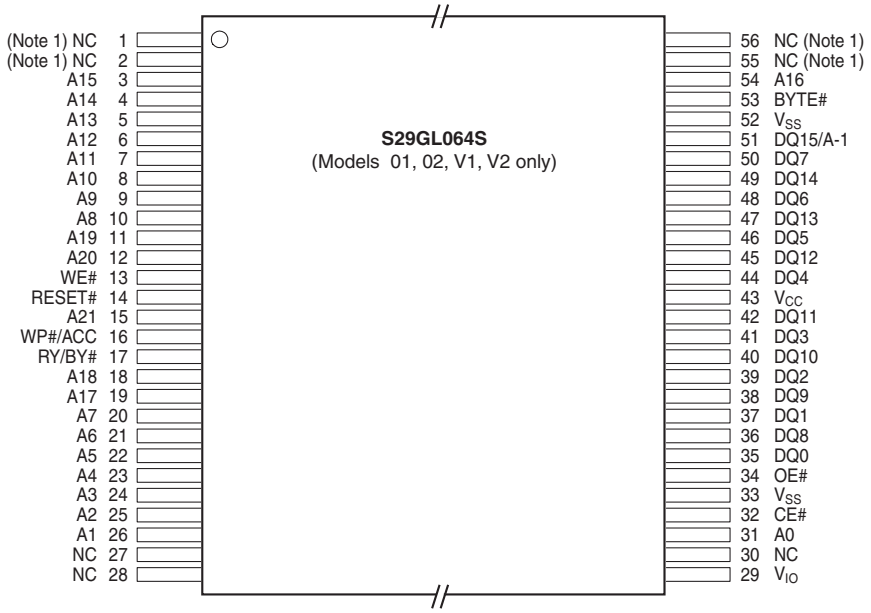



**Figure 2. 56-Pin Standard TSOP**



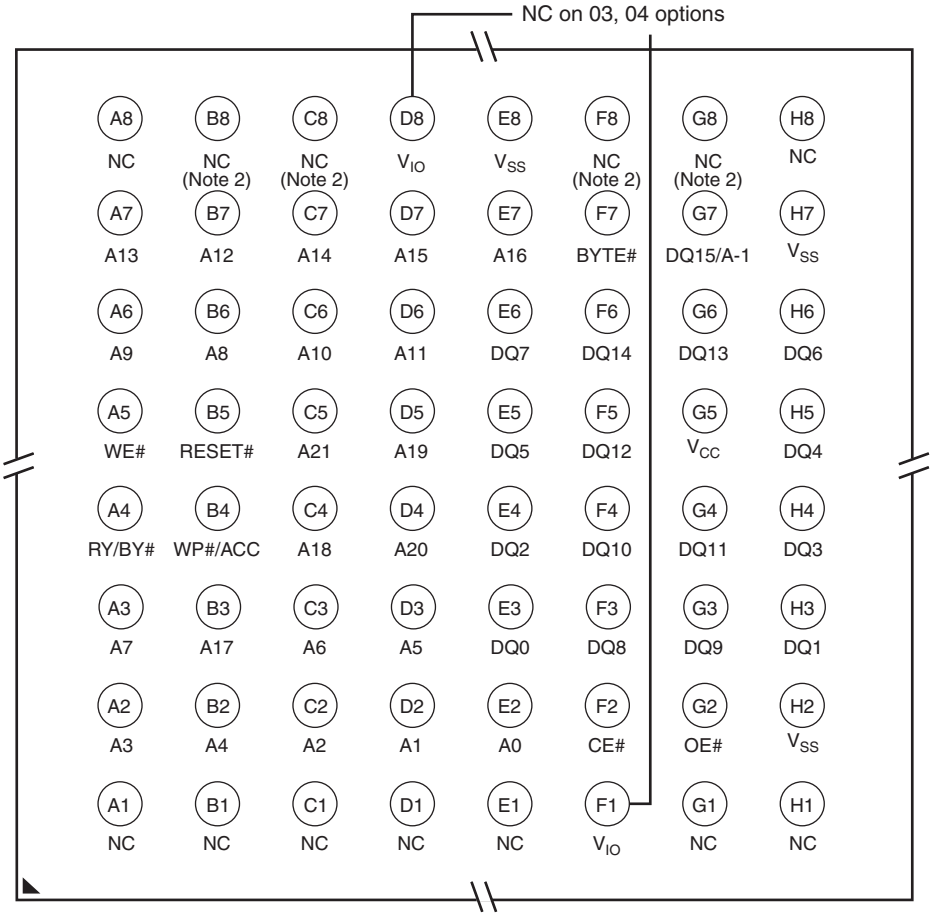

***Note:***

*1. These pins are NC on the S29GL064S, however, are used by 128-Mbit –1-Gbit density GL devices as the high order address inputs.*



Downloaded from Arrow.com.

**Figure 3. 64-Ball Fortified BGA**




***Notes:***

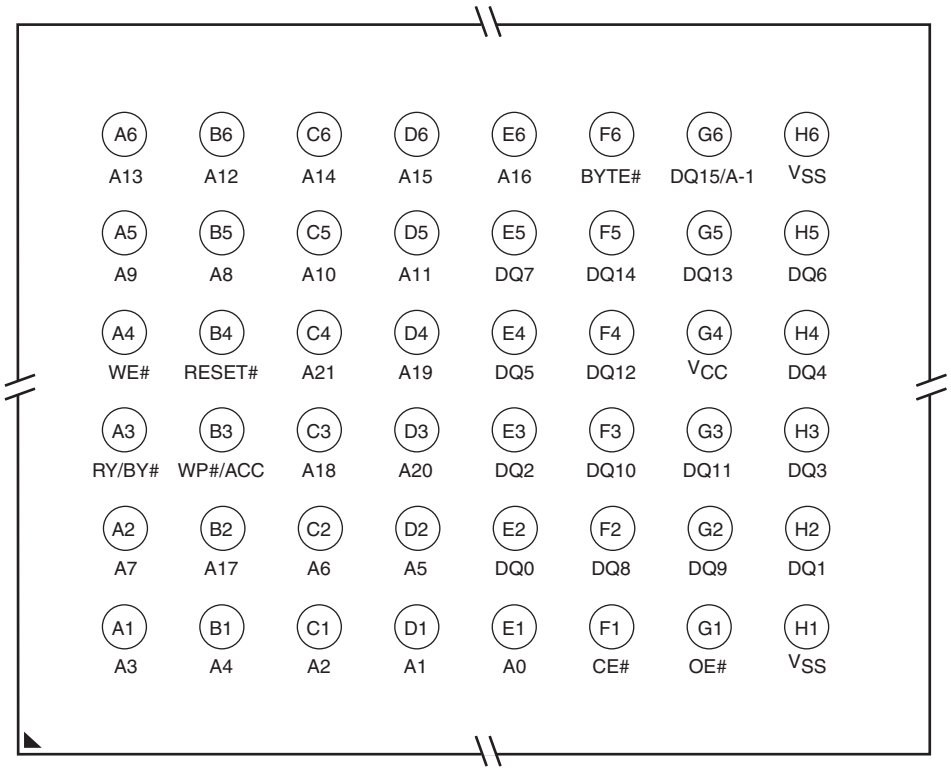
1. *S29GL064S (Models 01, 02, 03, 04, V1, V2).*
2. *These balls are NC on the S29GL064S, however, are used by 128- Mbit – 1-Gbit density GL devices as the high order address inputs.*



Downloaded from Arrow.com.

**Figure 4. 48-Ball Fine-Pitch BGA (VBK 048)**

**S29GL064S** (Models 03, 04 only)
Top View, Balls Facing Down






Downloaded from Arrow.com.

# 4. Pin Description

| Pin | Description |
|---|---|
| A21–A0 | 22 Address inputs (S29GL064S) |
| DQ7–DQ0 | 8 Data inputs / outputs |
| DQ14–DQ0 | 15 Data inputs / outputs |
| DQ15/A-1 | DQ15 (Data input / output, word mode), A-1 (LSB Address input, byte mode) |
| CE# | Chip Enable input |
| OE# | Output Enable input |
| WE# | Write Enable input |
| WP#/ACC | Hardware Write Protect input / Programming Acceleration input |
| ACC | Programming Acceleration input |
| WP# | Hardware Write Protect input |
| RESET# | Hardware Reset Pin input |
| RY/BY# | Ready/Busy output |
| BYTE# | Selects 8-bit or 16-bit mode |
| VCC | 3.0 volt-only single power supply (see Product Selector Guide on page 4 for speed options and voltage supply tolerances) |
| VIO | Output Buffer Power |
| VSS | Device Ground |
| NC | Pin Not Connected Internally |
| RFU | Reserved for Future Use. Not currently connected internally but the pin/ball location should be left unconnected and unused by PCB routing channel for future compatibility. The pin/ball may be used by a signal in the future. |



Downloaded from Arrow.com.

# 5. S29GL064S Logical Symbols

**Figure 5. S29GL064S Logic Symbol (Models 01, 02, V1, V2)**

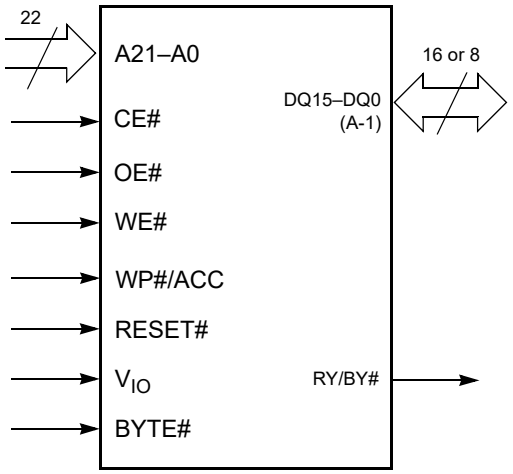



**Figure 6. S29GL064S Logic Symbol (Models 03, 04)**

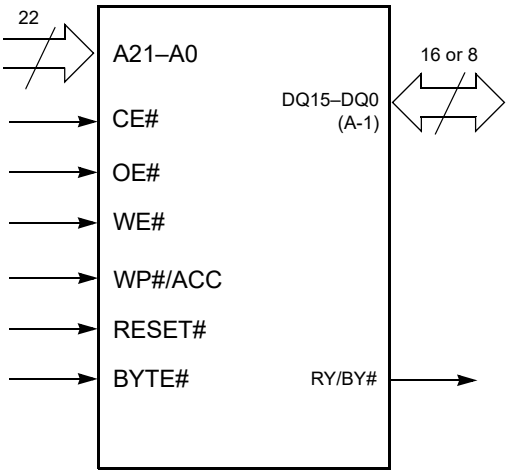
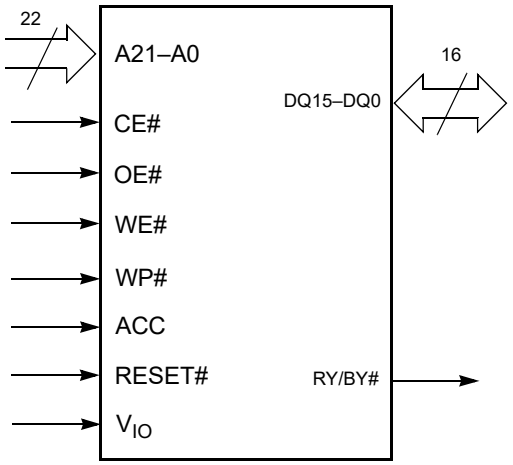



**Figure 7. S29GL064S Logic Symbol (Models 06, 07, V6, V7)**

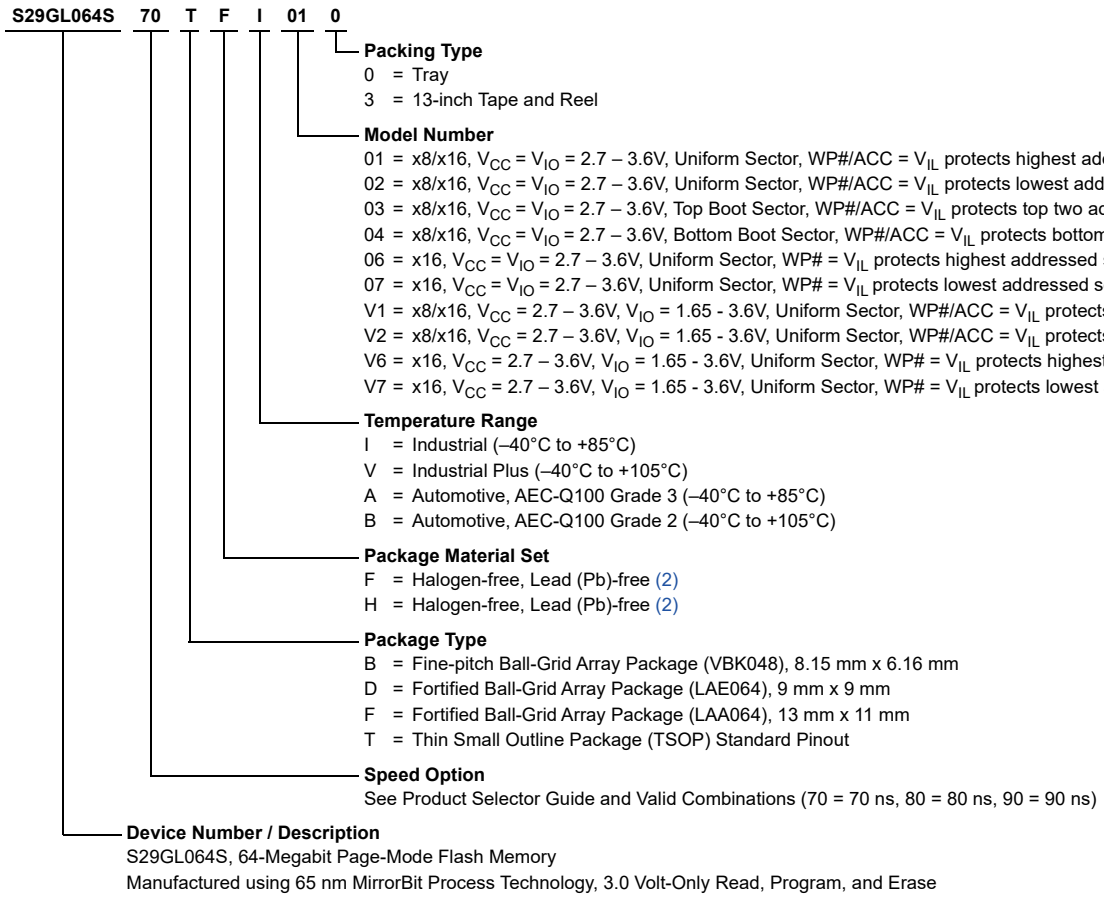





Downloaded from Arrow.com.

# 6. Ordering Information

Standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of the following:




***Notes:***

1. *$V_{IO}$ is tied internally to $V_{CC}$.*
2. *Halogen-free definition is in accordance with IEC 61249-2-21 specification.*



Downloaded from Arrow.com.

## 6.1 Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult your local sales office to confirm availability of specific valid combinations and to check on newly released combinations. Table 3 and Table 4 list configurations that are standard units and Automotive Grade / AEC-Q100 qualified units.

Production Part Approval Process (PPAP) support is only provided for AEC-Q100 grade products.

Products to be used in end-use applications that require ISO/TS-16949 compliance must be AEC-Q100 grade products in combination with PPAP. Non–AEC-Q100 grade products are not manufactured or documented in full compliance with ISO/TS-16949 requirements. AEC-Q100 grade products are also offered without PPAP support for end-use applications that do not require ISO/TS-16949 compliance.

**Table 3. Industrial (-40°C to +85°C)**

| S29GL064S Valid Combinations | | | | | Package Description | |
|---|---|---|---|---|---|---|
| Device Number | Speed Option | Package, Material, and Temperature Range | Model Number | Packing Type | | |
| S29GL064S | 70 | TFI, TFA | 03, 04, 06, 07 | 0,3 (Note 1) | TS048 (Note 2) | TSOP |
| | 80 | | V6, V7 | | | |
| | 70 | | 01, 02 | | TS056 (Note 2) | |
| | 80 | | V1, V2 | | | |
| | 70 | BHI, BHA | 03, 04 | | VBK048 (Note 3) | Fine-Pitch BGA |
| | 70 | FHI, FHA | 01, 02, 03, 04 | | LAA064 (Note 3) | Fortified BGA |
| | 80 | | V1, V2 | | | |
| | 70 | DHI, FHA | 01, 02, 03, 04 | | LAE064 (Note 3) | |
| | 80 | | V1, V2 | | | |

***Notes:***

1. *Type 0 is standard. Specify others as required.*
2. *TSOP package marking omits packing type designator from ordering part number.*
3. *BGA package marking omits leading S29 and packing type designator from ordering part number.*

**Table 4. Industrial Plus (-40°C to +105°C)**

| S29GL064S Valid Combinations | | | | | Package Description | |
|---|---|---|---|---|---|---|
| Device Number | Speed Option | Package, Material, and Temperature Range | Model Number | Packing Type | | |
| S29GL064S | 80 | TFV, TFB | 03, 04, 06, 07 | 0,3 (Note 1) | TS048 (Note 2) | TSOP |
| | 90 | | V6, V7 | | | |
| | 80 | | 01, 02 | | TS056 (Note 2) | |
| | 90 | | V1, V2 | | | |
| | 80 | BHV, BHB | 03, 04 | | VBK048 (Note 3) | Fine-Pitch BGA |
| | 80 | FHV, FHB | 01, 02, 03, 04 | | LAA064 (Note 3) | Fortified BGA |
| | 90 | | V1, V2 | | | |
| | 80 | DHV, DHB | 01, 02, 03, 04 | | LAE064 (Note 3) | |
| | 90 | | V1, V2 | | | |

***Notes:***

1. *Type 0 is standard. Specify others as required.*
2. *TSOP package marking omits packing type designator from ordering part number.*
3. *BGA package marking omits leading S29 and packing type designator from ordering part number.*



Downloaded from Arrow.com.

# 7. Other Resources

## 7.1 Cypress Flash Memory Roadmap

www.cypress.com/Flash-Roadmap

## 7.2 Links to Software

www.cypress.com/software-and-drivers-cypress-flash-memory

## 7.3 Links to Application Notes

www.cypress.com/cypressappnotes



Downloaded from Arrow.com.

# 8. Device Bus Operations

This section describes the requirements and use of the device bus operations, which are initiated through the internal command register. The command register itself does not occupy any addressable memory location. The register is a latch used to store the commands, along with the address and data information needed to execute the command. The contents of the register serve as inputs to the internal state machine. The state machine outputs dictate the function of the device. Table 5 lists the device bus operations, the inputs and control levels they require, and the resulting output. The following subsections describe each of these operations in further detail.

**Table 5. Device Bus Operations**

| Operation | CE# | OE# | WE# | RESET# | BYTE# (Note 4) | WP# | ACC | Addresses | DQ0–DQ7 | DQ8–DQ15 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | BYTE# = $V_{IH}$ | BYTE# = $V_{IL}$ |
| Read | L | L | H | H | L or H | X | X | $A_{IN}$ | $D_{OUT}$ | $D_{OUT}$ | DQ8–DQ14 = High-Z, DQ15 = A-1 |
| Autoselect (HV) | L | L | H | H | L or H | X | H | $A_{IN}$ (Note 3) | $D_{OUT}$ | $D_{OUT}$ | |
| Write (Program / Erase) | L | H | L | H | L or H | (Note 1) | X | $A_{IN}$ | (Note 2) | (Note 2) | |
| Accelerated Program | L | H | L | H | L or H | (Note 1) | $V_{HH}$ | $A_{IN}$ | (Note 2) | (Note 2) | |
| Standby | $V_{IO} \pm 0.3V$ | X | X | $V_{IO} \pm 0.3V$ | L or H | X | H | X | High-Z | High-Z | High-Z |
| Output Disable | L | H | H | H | L or H | X | X | X | High-Z | High-Z | High-Z |
| | H | X | X | | | | | | | | |
| Reset | X | X | X | L | L or H | X | X | X | High-Z | High-Z | High-Z |

***Legend:***

*L = Logic Low = $V_{IL}$*
*H = Logic High = $V_{IH}$*
*$V_{HH}$ = Voltage for ACC Program Acceleration*
*$V_{ID}$ = Voltage for Autoselect*
*X = Don't Care*
*$A_{IN}$ = Address In*
*$D_{IN}$ = Data In*
*$D_{OUT}$ = Data Out*

***Notes:***

1. *If WP# = $V_{IL}$, the first or last sector remains protected (for uniform sector devices), and the two outer boot sectors are protected (for boot sector devices). If WP# = $V_{IH}$, the first or last sector, or the two outer boot sectors are protected or unprotected as determined by the method described in Write Protect (WP#). All sectors are unprotected when shipped from the factory (The Secure Silicon Region may be factory protected depending on version ordered.)*
2. *$D_{IN}$ or $D_{OUT}$ as required by command sequence, data polling, or sector protect algorithm (see Figure 12 on page 53).*
3. *A9 is raised to $V_{ID}$ to enable Autoselect reads.*
4. *$V_{IL} = V_{SS}$ and $V_{IH} = V_{IO}$.*

## 8.1 Word / Byte Configuration

The BYTE# pin controls whether the device data I/O pins operate in the byte or word configuration. If the BYTE# pin is set at logic 1, the device is in word configuration, DQ0–DQ15 are active and controlled by CE#, WE# and OE#.

If the BYTE# pin is set at logic 0, the device is in byte configuration, and only data I/O pins DQ0–DQ7 are active and controlled by CE#, WE# and OE#. The data I/O pins DQ8–DQ14 are tri-stated, and the DQ15 pin is used as an input for the LSB (A-1) address function.

The BYTE# pin must be driven set to a logic 0 or 1 state prior to CE# being driven low. The BYTE# pin should not change logic state while CE# is low.



Downloaded from Arrow.com.

## 8.2 Requirements for Reading Array Data

All memories require access time to output array data. In a read operation, data is read from one memory location at a time. Addresses are presented to the device in random order, and the propagation delay through the device causes the data on its outputs to arrive with the address on its inputs.

The device defaults to reading array data after device power-up or hardware reset. To read data from the memory array, the system must first assert a valid address on Amax–A0, while driving OE# and CE# to $V_{IL}$. WE# must remain at $V_{IH}$. Data will appear on DQ15–DQ0 after address access time ($t_{ACC}$), which is equal to the delay from stable addresses to valid output data. The OE# signal must be driven to $V_{IL}$. Data is output on DQ15-DQ0 pins after the access time ($t_{OE}$) has elapsed from the falling edge of OE#.

See Reading Array Data on page 30 for more information. Refer to Table 67 on page 84 and Table 68 on page 85 for timing specifications and the timing diagram. Refer to Table 59 on page 77 and Table 60 on page 78 for the active current specification on reading array data.

### 8.2.1 Page Mode Read

The device is capable of fast page mode read and is compatible with the page mode Mask ROM read operation. This mode provides faster read access speed for random locations within a page. The page size of the device is 8 words / 16 bytes. The appropriate page is selected by the higher address bits A(max)–A3. Address bits A2–A0 in word mode (A2–A-1 in byte mode) determine the specific word within a page. This is an asynchronous operation; the microprocessor supplies the specific word location.

The random or initial page access is equal to $t_{ACC}$ or $t_{CE}$ and subsequent page read accesses (as long as the locations specified by the microprocessor falls within that page) is equivalent to $t_{PACC}$. When CE# is deasserted and reasserted for a subsequent access, the access time is $t_{ACC}$ or $t_{CE}$. Fast page mode accesses are obtained by keeping the *read-page addresses* constant and changing the intra-read *page* addresses.

## 8.3 Writing Commands / Command Sequences

To write a command or command sequence (which includes programming data to the device and erasing sectors of memory), the system must drive WE# and CE# to $V_{IL}$, and OE# to $V_{IH}$.

The device features an **Unlock Bypass** mode to facilitate faster programming. Once the device enters the Unlock Bypass mode, only two write cycles are required to program a word, instead of four. The Table 18 on page 32 contains details on programming data to the device using both standard and Unlock Bypass command sequences.

An erase operation can erase one sector, multiple sectors, or the entire device. Tables 6–9 indicate the address space that each sector occupies.

Refer to DC Characteristicst on page 77 for the active current specification for the write mode. The *AC Characteristics* section contains timing specification tables and timing diagrams for write operations.

### 8.3.1 Write Buffer

Write Buffer Programming allows the system write to a maximum of 128 words / 256 bytes in one programming operation. This results in faster effective programming time than the standard programming algorithms.

### 8.3.2 Accelerated Program Operation

The device offers program operations through the ACC function. This is one of two functions provided by the WP#/ACC or ACC pin, depending on model number. This function is primarily intended to support manufacturing programming operations at the factory.

If the system asserts $V_{HH}$ on this pin, the device automatically enters the Unlock Bypass mode, protected sectors will remain protected. The system would use a two-cycle program command sequence as required by the Unlock Bypass mode. Removing $V_{HH}$ from the WP#/ACC or ACC pin, depending on model number, returns the device to normal operation. *Note that the WP#/ACC or ACC pin must be raised to $V_{HH}$ prior to any accelerated operation and should return to $V_{IL}/V_{IH}$ after the completion of the accelerated operation. It should not be at $V_{HH}$ for operations other than accelerated programming, or device damage may result. WP# contains an internal pull-up; when unconnected, WP# is at $V_{IH}$.*



Downloaded from Arrow.com.

### 8.3.3 Autoselect Functions

If the system writes the autoselect command sequence, the device enters the autoselect mode. The system can then read autoselect codes from the internal register (which is separate from the memory array) on DQ7-DQ0. Standard read cycle timings ($t_{ACC}$) apply in this mode. Refer to Autoselect Mode on page 19 and Autoselect Command Sequence on page 31 for more information.

## 8.4 Automatic ECC

### 8.4.1 ECC Overview

The Automatic ECC feature works transparently with normal program, erase, and read operations. As the device transfers each page of data from the Write Buffer to the memory array, internal ECC logic programs the ECC code for that page into a portion of the memory array that is not visible to the host system. The device evaluates the page data and the ECC code during each initial page access. If needed, the internal ECC logic will correct a single bit error during the initial access.

Programming more than once to a particular page will disable the ECC function for that page. The ECC function for that page will remain disabled until the next time the host system erases the sector containing that page. The host system may read data stored in that page following multiple programming operations; however, ECC remains disables and the device will not detect or correct an error in that page.

### 8.4.2 Program and Erase Summary

For performance and reliability reasons, the device performs reading and programming operations on full 32-byte pages in parallel. Internal device logic provides ECC on each page by adding an ECC code when the page is first programmed.

### 8.4.3 ECC Implementation

Each 32-byte page in the main flash array, as well as each 32-byte OTP region, features an associated ECC code. Internal ECC logic is able to detect and correct any single bit error found in a page or the associated ECC code during a read access. The first Write Buffer program operation applied to a page programs the ECC code for that page. Subsequent programming operations that occur more than once on a particular page will disable the ECC function for that page. This allows bit or word programming; however, multiple programming operations to the same page will disable the ECC function on the page where incremental programming occurs. An erase of the sector containing the page with ECC disabled will re-enable the ECC function for that Page.

The ECC function is automatic and transparent to the user. The transparency of the Automatic ECC function enhances data integrity for typical programming operations that write data once to each page. The ECC function also facilitates software compatibility to previous generations of GL Family products by allowing single word programming and bit-walking where the user programs the same page or word more than once. When a page has Automatic ECC disabled, the ECC function will not detect or correct any errors upon a data read from that page.

### 8.4.4 Word Programming

A word programming operation programs a single word anywhere in the main memory array. Programming multiple words within the same 32-byte page disables the Automatic ECC function for that page. An erase of the sector containing that page will re-enable Automatic ECC following multiple word programming operation on that page.

### 8.4.5 Write Buffer Programming

Each Write Buffer program operation allows the user to program a single bit up to 256 bytes. A 32-byte page is the smallest program granularity that features Automatic ECC protection. Programming to the same page more than once will disable the Automatic ECC function for that page. Cypress recommends the use of a Write Buffer programming operation to program multiple pages in an operation and to write each page only once. This keeps the Automatic ECC function enabled on each page. For the very best performance, program in full lines of 256 bytes aligned on 256-byte boundaries.



Downloaded from Arrow.com.

## 8.5 Standby Mode

When the system is not reading or writing to the device, it can be placed in to standby mode. In this mode, current consumption is greatly reduced, and the outputs are placed in the high impedance state, independent of the OE# input.

The device enters the CMOS standby mode when the CE# and RESET# pins are both held at $V_{IO}$ ± 0.3V. (Note that this is a more restricted voltage range than $V_{IH}$.) If CE# and RESET# are held at $V_{IH}$, but not within $V_{IO}$ ± 0.3V, the device is in the standby mode, but the standby current is greater. The device requires standard access time ($t_{ACC}$/$t_{CE}$) for read access when the device is in either of these standby modes, before it is ready to read data.

If the device is deselected during erasure or programming, the device draws active current until the operation is completed.

Refer to the DC Characteristicst on page 77, for the standby current specification.

## 8.6 Automatic Sleep Mode

The automatic sleep mode reduces device interface energy consumption to the sleep level ($I_{CC6}$) following the completion of a random read access time. The device automatically enables this mode when addresses remain stable for $t_{ACC}$ + 30 ns. While in sleep mode, output data is latched and always available to the system. Output of the data depends on the level of the OE# signal but, the automatic sleep mode current is independent of the OE# signal level. Standard address access timings ($t_{ACC}$ or $t_{PACC}$) provide new data when addresses are changed. Refer to the DC Characteristicst on page 77 for the automatic sleep mode current specification $I_{CC6}$.

Automatic sleep helps reduce current consumption especially when the host system clock is slowed for power reduction. During slow system clock periods, read and write cycles may extend many times their length versus when the system is operating at high speed. Even though CE# may be Low throughout these extended data transfer cycles, the memory device host interface will go to the Automatic Sleep current at $t_{ACC}$ + 30 ns. The device will remain at the Automatic Sleep current for $t_{ASSB}$. Then the device will transition to the standby current level. This keeps the memory at the Automatic Sleep or standby power level for most of the long duration data transfer cycles, rather than consuming full read power all the time that the memory device is selected by the host system.

However, the EAC operates independent of the automatic sleep mode of the host interface and will continue to draw current during an active Embedded Algorithm. Only when both the host interface and EAC are in their standby states is the standby level current achieved.

## 8.7 RESET#: Hardware Reset Pin

The RESET# pin provides a hardware method of resetting the device to reading array data. When the RESET# pin is driven low for at least a period of $t_{RP}$, the device immediately terminates any operation in progress, output pins go to High-Z, and all read / write commands are ignored for the duration of the RESET# pulse. Program / Erase operations that were interrupted should be reinitiated once the device is ready to accept another command sequence, to ensure data integrity.

Current is reduced for the duration of the RESET# pulse. When RESET# is held at $V_{SS}$ ±0.3V long enough, the device draws CMOS standby current ($I_{CC5}$).

The RESET# pin may be tied to the system reset circuitry. A system reset would thus also reset the flash memory, enabling the system to read the boot-up firmware from the flash memory.

Refer to the AC Characteristics on page 84 for RESET# parameters and to Figure 21 on page 83 for the timing diagram.



Downloaded from Arrow.com.

## 8.8 Output Disable Mode

When the OE# input is at $V_{IH}$, output from the device is disabled. The output pins are placed in a high impedance state.

## 8.9 Memory Map

**Table 6. S29GL064S (Models 01, 02, V1, V2) Sector Addresses**

| Sector | A21–A15 | Sector Size (kB/ kwords) | 8-bit Address Range | 16-bit Address Range |
|---|---|---|---|---|
| SA0 | 0000000 | 64/32 | 000000h–00FFFFh | 000000h–007FFFh |
| SA1 | 0000001 | 64/32 | 010000h–01FFFFh | 008000h–00FFFFh |
| SA2 | 0000010 | 64/32 | 020000h–02FFFFh | 010000h–017FFFh |
| SA3 | 0000011 | 64/32 | 030000h–03FFFFh | 018000h–01FFFFh |
| SA4 | 0000100 | 64/32 | 040000h–04FFFFh | 020000h–027FFFh |
| SA5 | 0000101 | 64/32 | 050000h–05FFFFh | 028000h–02FFFFh |
| SA6 | 0000110 | 64/32 | 060000h–06FFFFh | 030000h–037FFFh |
| SA7 | 0000111 | 64/32 | 070000h–07FFFFh | 038000h–03FFFFh |
| SA8 | 0001000 | 64/32 | 080000h–08FFFFh | 040000h–047FFFh |
| SA9 | 0001001 | 64/32 | 090000h–09FFFFh | 048000h–04FFFFh |
| ... | ... | ... | ... | ... |

| Sector | A21–A15 | Sector Size (kB/ kwords) | 8-bit Address Range | 16-bit Address Range |
|---|---|---|---|---|
| ... | ... | ... | ... | ... |
| SA118 | 1110110 | 64/32 | 760000h–76FFFFh | 3B0000h–3B7FFFh |
| SA119 | 1110111 | 64/32 | 770000h–77FFFFh | 3B8000h–3BFFFFh |
| SA120 | 1111000 | 64/32 | 780000h–78FFFFh | 3C0000h–3C7FFFh |
| SA121 | 1111001 | 64/32 | 790000h–79FFFFh | 3C8000h–3CFFFFh |
| SA122 | 1111010 | 64/32 | 7A0000h–7AFFFFh | 3D0000h–3D7FFFh |
| SA123 | 1111011 | 64/32 | 7B0000h–7BFFFFh | 3D8000h–3DFFFFh |
| SA124 | 1111100 | 64/32 | 7C0000h–7CFFFFh | 3E0000h–3E7FFFh |
| SA125 | 1111101 | 64/32 | 7D0000h–7DFFFFh | 3E8000h–3EFFFFh |
| SA126 | 1111110 | 64/32 | 7E0000h–7EFFFFh | 3F0000h–3F7FFFh |
| SA127 | 1111111 | 64/32 | 7F0000h–7FFFFFh | 3F8000h–3FFFFFh |

**Table 7. S29GL064S (Model 03) Top Boot Sector Addresses**

| Sector | A21–A12 | Sector Size (kB/ kwords) | 8-bit Address Range | 16-bit Address Range |
|---|---|---|---|---|
| SA0 | 0000000xxx | 64/32 | 000000h–00FFFFh | 000000h–007FFFh |
| SA1 | 0000001xxx | 64/32 | 010000h–01FFFFh | 008000h–00FFFFh |
| SA2 | 0000010xxx | 64/32 | 020000h–02FFFFh | 010000h–017FFFh |
| SA3 | 0000011xxx | 64/32 | 030000h–03FFFFh | 018000h–01FFFFh |
| SA4 | 0000100xxx | 64/32 | 040000h–04FFFFh | 020000h–027FFFh |
| SA5 | 0000101xxx | 64/32 | 050000h–05FFFFh | 028000h–02FFFFh |
| SA6 | 0000110xxx | 64/32 | 060000h–06FFFFh | 030000h–037FFFh |
| SA7 | 0000111xxx | 64/32 | 070000h–07FFFFh | 038000h–03FFFFh |
| SA8 | 0001000xxx | 64/32 | 080000h–08FFFFh | 040000h–047FFFh |
| SA9 | 0001001xxx | 64/32 | 090000h–09FFFFh | 048000h–04FFFFh |
| ... | ... | ... | ... | ... |

| Sector | A21–A12 | Sector Size (kB/ kwords) | 8-bit Address Range | 16-bit Address Range |
|---|---|---|---|---|
| ... | ... | ... | ... | ... |
| SA125 | 1111101xxx | 64/32 | 7D0000h–7DFFFFh | 3E8000h–3EFFFFh |
| SA126 | 1111110xxx | 64/32 | 7E0000h–7EFFFFh | 3F0000h–3F7FFFh |
| SA127 | 1111111000 | 8/4 | 7F0000h–7F1FFFh | 3F8000h–3F8FFFh |
| SA128 | 1111111001 | 8/4 | 7F2000h–7F3FFFh | 3F9000h–3F9FFFh |
| SA129 | 1111111010 | 8/4 | 7F4000h–7F5FFFh | 3FA000h–3FAFFFh |
| SA130 | 1111111011 | 8/4 | 7F6000h–7F7FFFh | 3FB000h–3FBFFFh |
| SA131 | 1111111100 | 8/4 | 7F8000h–7F9FFFh | 3FC000h–3FCFFFh |
| SA132 | 1111111101 | 8/4 | 7FA000h–7FBFFFh | 3FD000h–3FDFFFh |
| SA133 | 1111111110 | 8/4 | 7FC000h–7FDFFFh | 3FE000h–3FEFFFh |
| SA134 | 1111111111 | 8/4 | 7FE000h–7FFFFFh | 3FF000h–3FFFFFh |



Downloaded from Arrow.com.

**Table 8. S29GL064S (Model 04) Bottom Boot Sector Addresses**

| Sector | A21–A12 | Sector Size (kB/ kwords) | 8-bit Address Range | 16-bit Address Range |
|---|---|---|---|---|
| SA0 | 0000000000 | 8/4 | 000000h–001FFFh | 000000h–000FFFh |
| SA1 | 0000000001 | 8/4 | 002000h–003FFFh | 001000h–001FFFh |
| SA2 | 0000000010 | 8/4 | 004000h–005FFFh | 002000h–002FFFh |
| SA3 | 0000000011 | 8/4 | 006000h–007FFFh | 003000h–003FFFh |
| SA4 | 0000000100 | 8/4 | 008000h–009FFFh | 004000h–004FFFh |
| SA5 | 0000000101 | 8/4 | 00A000h–00BFFFh | 005000h–005FFFh |
| SA6 | 0000000110 | 8/4 | 00C000h–00DFFFh | 006000h–006FFFh |
| SA7 | 0000000111 | 8/4 | 00E000h–00FFFFh | 007000h–007FFFh |
| SA8 | 0000001xxx | 64/32 | 010000h–01FFFFh | 008000h–00FFFFh |
| SA9 | 0000010xxx | 64/32 | 020000h–02FFFFh | 010000h–017FFFh |
| ... | ... | ... | ... | ... |

| Sector | A21–A12 | Sector Size (kB/ kwords) | 8-bit Address Range | 16-bit Address Range |
|---|---|---|---|---|
| ... | ... | ... | ... | ... |
| SA125 | 1110110xxx | 64/32 | 760000h–76FFFFh | 3B0000h–3B7FFFh |
| SA126 | 1110111xxx | 64/32 | 770000h–77FFFFh | 3B8000h–3BFFFFh |
| SA127 | 1111000xxx | 64/32 | 780000h–78FFFFh | 3C0000h–3C7FFFh |
| SA128 | 1111001xxx | 64/32 | 790000h–79FFFFh | 3C8000h–3CFFFFh |
| SA129 | 1111010xxx | 64/32 | 7A0000h–7AFFFFh | 3D0000h–3D7FFFh |
| SA130 | 1111011xxx | 64/32 | 7B0000h–7BFFFFh | 3D8000h–3DFFFFh |
| SA131 | 1111100xxx | 64/32 | 7C0000h–7CFFFFh | 3E0000h–3E7FFFh |
| SA132 | 1111101xxx | 64/32 | 7D0000h–7DFFFFh | 3E8000h–3EFFFFh |
| SA133 | 1111110xxx | 64/32 | 7E0000h–7EFFFFh | 3F0000h–3F7FFFh |
| SA134 | 1111111xxx | 64/32 | 7F0000h–7FFFFFh | 3F8000h–3FFFFFh |

**Table 9. S29GL064S (Models 06, 07, V6, V7) Sector Addresses**

| Sector | A21–A15 | Sector Size (kB/ kwords) | 16-bit Address Range |
|---|---|---|---|
| SA0 | 0000000 | 64/32 | 000000–007FFF |
| SA1 | 0000001 | 64/32 | 008000–00FFFF |
| SA2 | 0000010 | 64/32 | 010000–017FFF |
| SA3 | 0000011 | 64/32 | 018000–01FFFF |
| SA4 | 0000100 | 64/32 | 020000–027FFF |
| SA5 | 0000101 | 64/32 | 028000–02FFFF |
| SA6 | 0000110 | 64/32 | 030000–037FFF |
| SA7 | 0000111 | 64/32 | 038000–03FFFF |
| SA8 | 0001000 | 64/32 | 040000–047FFF |
| SA9 | 0001001 | 64/32 | 048000–04FFFF |
| ... | ... | ... | ... |

| Sector | A21–A15 | Sector Size (kB/ kwords) | 16-bit Address Range |
|---|---|---|---|
| ... | ... | ... | ... |
| SA118 | 1110110 | 64/32 | 3B0000–3B7FFF |
| SA119 | 1110111 | 64/32 | 3B8000–3BFFFF |
| SA120 | 1111000 | 64/32 | 3C0000–3C7FFF |
| SA121 | 1111001 | 64/32 | 3C8000–3CFFFF |
| SA122 | 1111010 | 64/32 | 3D0000–3D7FFF |
| SA123 | 1111011 | 64/32 | 3D8000–3DFFFF |
| SA124 | 1111100 | 64/32 | 3E0000–3E7FFF |
| SA125 | 1111101 | 64/32 | 3E8000–3EFFFF |
| SA126 | 1111110 | 64/32 | 3F0000–3F7FFF |
| SA127 | 1111111 | 64/32 | 3F8000–3FFFFF |

## 8.10 Autoselect Mode

The autoselect mode provides manufacturer and device identification, and sector protection verification, through identifier codes output on DQ7–DQ0. This mode is primarily intended for programming equipment to automatically match a device to be programmed with its corresponding programming algorithm. However, the autoselect codes can also be accessed in-system through the command register.

When using programming equipment, the autoselect mode requires $V_{ID}$ on address pin A9. Address pins A6, A3, A2, A1, and A0 must be as shown in Table 10 on page 20. In addition, when verifying sector protection, the sector address must appear on the appropriate highest order address bits (see Table 6–9). Table 10 shows the remaining address bits that are don't care. When all necessary bits are set as required, the programming equipment may then read the corresponding identifier code on DQ7–DQ0. *Note that the A9 pin must not be at $V_{ID}$ for operations other than Autoselect, or device damage may result. Autoselect using $V_{ID}$ is supported at room temperature only. It must be raised to $V_{ID}$ prior to any autoselect operations and should return to $V_{IL}/V_{IH}$ after the completion of the autoselect operation. It should not be at $V_{ID}$ for operations other than autoselect, or device damage may result.*

To access the autoselect codes in-system, the host system can issue the autoselect command via the command register, as shown in Table 21 on page 42 and Table 23 on page 45. This method does not require $V_{ID}$. Refer to the Autoselect Command Sequence on page 31 for more information.



Downloaded from Arrow.com.

ID-CFI Location 02h displays sector protection status for the sector selected by the sector address (SA) used in the ID-CFI enter command. To read the protection status of more than one sector it is necessary to exit the ID ASO and enter the ID ASO using the new SA. The access time to read location 02h is always $t_{ACC}$ and a read of this location requires CE# to go High before the read and return Low to initiate the read (asynchronous read access). Page mode read between location 02h and other ID locations is not supported. Page mode read between ID locations other than 02h is supported.

In x8 mode, address A-1 is ignored and the lower 8 bits of data will be returned for both address.

**Table 10. Autoselect Codes, (High Voltage Method)**

| Description | | CE# | OE# | WE# | $A_{max}$ to A15 | A14 to A10 | A9 | A8 to A7 | A6 | A5 to A4 | A3 to A2 | A1 | A0 | DQ8 to DQ15 | | DQ7 to DQ0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | Model Number | | |
| | | | | | | | | | | | | | | BYTE# = $V_{IH}$ | BYTE# = $V_{IL}$ | 01, 02 V1, V2 | 03, 04 | 06, 07, V6, V7 |
| Manufacturer ID: Cypress Products | | L | L | H | X | X | $V_{ID}$ | X | L | X | L | L | L | 00 | X | 01h | 01h | 01h |
| S29GL064S | Cycle 1 | L | L | H | X | X | $V_{ID}$ | X | L | X | L | L | H | 22 | X | 7Eh | 7Eh | 7Eh |
| | Cycle 2 | | | | | | | | | | H | H | L | 22 | X | 0Ch | 10h | 13h |
| | Cycle 3 | | | | | | | | | | H | H | H | 22 | X | 01h | 00h (04, bottom boot) 01h (03, top boot) | 01h |
| Sector Protection Verification | | L | L | H | SA | X | $V_{ID}$ | X | L | X | L | H | L | X | X | 01h (protected), 00h (unprotected) | | |
| Secure Silicon Region Indicator Bit (DQ7), WP# protects highest address sector | | L | L | H | X | X | $V_{ID}$ | X | L | X | L | H | H | X | X | 9A (factory locked), 1A (not factory locked) | | |
| Secure Silicon Region Indicator Bit (DQ7), WP# protects lowest address sector | | L | L | H | X | X | $V_{ID}$ | X | L | X | L | H | H | X | X | 8A (factory locked), 0A (not factory locked) | | |

***Legend:***
*L = Logic Low = $V_{IL}$*
*H = Logic High = $V_{IH}$*
*SA = Sector Address*
*X = Don't care*

## 8.11 Advanced Sector Protection

The device features several levels of sector protection, which can disable both the program and erase operations in certain sectors.

### 8.11.1 Persistent Sector Protection

A command sector protection method that replaces the old 12V controlled protection method.

### 8.11.2 Password Sector Protection

A highly sophisticated protection method that requires a password before changes to certain sectors are permitted.



Downloaded from Arrow.com.

### 8.11.3 WP# Hardware Protection

A write protect pin that can prevent program or erase operations in the outermost sectors.

The WP# Hardware Protection feature is always available, independent of the software managed protection method chosen.

### 8.11.4 Selecting a Sector Protection Mode

All parts default to operate in the Persistent Sector Protection mode. The user must then choose if the Persistent or Password Protection method is most desirable. There are two one-time programmable non-volatile bits that define which sector protection method is used. If the user decides to continue using the Persistent Sector Protection method, they must set the Persistent Sector Protection Mode Locking Bit. This permanently sets the part to operate only using Persistent Sector Protection. If the user decides to use the password method, they must set the Password Mode Locking Bit. This permanently sets the part to operate only using password sector protection.

It is important to remember that setting either the Persistent Sector Protection Mode Locking Bit or the Password Mode Locking Bit permanently selects the protection mode. It is not possible to switch between the two methods once a locking bit is set. **It is important that one mode is explicitly selected when the device is first programmed, rather than relying on the default mode alone.** This is so that it is not possible for a system program or virus to later set the Password Mode Locking Bit, which would cause an unexpected shift from the default Persistent Sector Protection Mode into the Password Protection Mode.

The device is shipped with all sectors unprotected. Cypress offers the option of programming and protecting sectors at the factory prior to shipping the device through the ExpressFlash™ Service. Contact your sales representative for details.

It is possible to determine whether a sector is protected or unprotected. See Autoselect Command Sequence on page 31 for details.

## 8.12 Lock Register

The Lock Register consists of 3 bits (DQ2, DQ1, and DQ0). These DQ2, DQ1, DQ0 bits of the Lock Register are programmable by the user. Users are not allowed to program both DQ2 and DQ1 bits of the Lock Register to the 00 state. If the user tries to program DQ2 and DQ1 bits of the Lock Register to the 00 state, the device aborts the Lock Register back to the default 11 state. Once either DQ2 and DQ1 bits of the Lock Register are programmed than no further changes are allow on DQ2 and DQ1. The programming time of the Lock Register is same as the typical word programming time ($t_{WHWH1}$) without utilizing the Write Buffer of the device. During a Lock Register programming sequence execution, the DQ6 Toggle Bit I toggles until the programming of the Lock Register has completed to indicate programming status. All Lock Register bits are readable to allow users to verify Lock Register statuses.

The Customer Secure Silicon Region Protection Bit is DQ0, Persistent Protection Mode Lock Bit is DQ1, and Password Protection Mode Lock Bit is DQ2 are accessible by all users. Each of these bits are non-volatile. DQ15–DQ3 are reserved and must be 1's when the user tries to program the DQ2, DQ1, and DQ0 bits of the Lock Register. The user is not required to program DQ2, DQ1 and DQ0 bits of the Lock Register at the same time. This allows users to lock the Secure Silicon Region and then set the device either permanently into Password Protection Mode or Persistent Protection Mode and then lock the Secure Silicon Region at separate instances and time frames.

- Secure Silicon Region Protection allows the user to lock the Secure Silicon Region area.
- Persistent Protection Mode Lock Bit allows the user to set the device permanently to operate in the Persistent Protection Mode.
- Password Protection Mode Lock Bit allows the user to set the device permanently to operate in the Password Protection Mode.

**Table 11. Lock Register**

| Bit | DQ15–6 | DQ5 | DQ4 | DQ3 | DQ2 | DQ1 | DQ0 |
|---|---|---|---|---|---|---|---|
| **Name** | Don't Care | Reserved | Reserved | Reserved | Password Protection Mode Lock Bit | Persistent Protection Mode Lock Bit | Secure Silicon Region Protection Bit |
| **Default Value** | 1 | 1 | 1 | 1 | 1 | 1 | 0 |



Downloaded from Arrow.com.

## 8.13 Persistent Sector Protection

The Persistent Sector Protection method replaces the old 12V controlled protection method while at the same time enhancing flexibility by providing three different sector protection states.

| | |
|---|---|
| Dynamically Locked | The sector is protected and can be changed by a simple command. |
| Persistently Locked | A sector is protected and cannot be changed. |
| Unlocked | The sector is unprotected and can be changed by a simple command. |

To achieve these states, three types of “bits” are used:

### 8.13.1 Dynamic Protection Bit (DYB)

A volatile protection bit is assigned for each sector. After power-up or hardware reset, the contents of all DYB bits are in the “unprotected state”. Each DYB is individually modifiable through the DYB Set Command and DYB Clear Command. The DYB bits and Persistent Protect Bits (PPB) Lock bit are defaulted to power up in the cleared state or unprotected state - meaning the all PPB bits are changeable.

The Protection State for each sector is determined by the logical OR of the PPB and the DYB related to that sector. For the sectors that have the PPB bits cleared, the DYB bits control whether or not the sector is protected or unprotected. By issuing the DYB Set and DYB Clear command sequences, the DYB bits is protected or unprotected, thus placing each sector in the protected or unprotected state. These are the so-called Dynamic Locked or Unlocked states. They are called dynamic states because it is very easy to switch back and forth between the protected and un-protected conditions. This allows software to easily protect sectors against inadvertent changes yet does not prevent the easy removal of protection when changes are needed.

The DYB bits maybe set or cleared as often as needed. The PPB bits allow for a more static, and difficult to change, level of protection. The PPB bits retain their state across power cycles because they are Non-Volatile. Individual PPB bits are set with a program command but must all be cleared as a group through an erase command.

The PPB Lock Bit adds an additional level of protection. Once all PPB bits are programmed to the desired settings, the PPB Lock Bit may be set to the ‘freeze state’. Setting the PPB Lock Bit to the freeze state disables all program and erase commands to the Non-Volatile PPB bits. In effect, the PPB Lock Bit locks the PPB bits into their current state. The only way to clear the PPB Lock Bit to the ‘unfreeze state’ is to go through a power cycle, or hardware reset. The Software Reset command does not clear the PPB Lock Bit to the unfreeze state. System boot code can determine if any changes to the PPB bits are needed e.g., to allow new system code to be downloaded. If no changes are needed then the boot code can set the PPB Lock Bit to disable any further changes to the PPB bits during system operation.

The WP# write protect pin adds a final level of hardware protection. When this pin is low it is not possible to change the contents of the WP# protected sectors. These sectors generally hold system boot code. So, the WP# pin can prevent any changes to the boot code that could override the choices made while setting up sector protection during system initialization.

It is possible to have sectors that have been persistently locked, and sectors that are left in the dynamic state. The sectors in the dynamic state are all unprotected. If there is a need to protect some of them, a simple DYB Set command sequence is all that is necessary. The DYB Set and DYB Clear commands for the dynamic sectors switch the DYB bits to signify protected and unprotected, respectively. If there is a need to change the status of the persistently locked sectors, a few more steps are required. First, the PPB Lock Bit must be disabled to the unfreeze state by either putting the device through a power-cycle, or hardware reset. The PPB bits can then be changed to reflect the desired settings. Setting the PPB Lock Bit once again to the freeze state locks the PPB bits, and the device operates normally again.

To achieve the best protection, execute the PPB Lock Bit Set command early in the boot code, and protect the boot code by holding WP# = $V_{IL}$.



Downloaded from Arrow.com.

### 8.13.2 Persistent Protection Bit (PPB)

A single Persistent (non-volatile) Protection Bit is assigned to each sector. If a PPB is programmed to the protected state through the PPB Program command, that sector is protected from program or erase operations and is therefor read-only. If a PPB requires erasure, all of the sector PPB bits must first be erased in parallel through the All PPB Erase command. The All PPB Erase command preprograms all PPB bits prior to PPB erasing. All PPB bits erase in parallel, unlike programming where individual PPB bits are programmable. The PPB bits are limited to the same number of cycles as a flash memory sector.

Programming the PPB bit requires the typical word programming time without utilizing the Write Buffer. During a PPB bit programming and all PPB bit erasing sequence executions, the DQ6 Toggle Bit I toggles until the programming of the PPB bit or erasing of all PPB bits has completed to indicate programming and erasing status. Erasing all of the PPB bits at once requires typical sector erase time. During the erasing of all PPB bits, the DQ3 Sector Erase Timer bit outputs a 1 to indicate the erasure of all PPB bits are in progress. Reading the PPB Status bit requires the initial access time of the device.

### 8.13.3 Persistent Protection Bit Lock (PPB Lock Bit)

A global volatile bit. When set to the freeze state, the PPB bits cannot be changed. When cleared to the unfreeze state, the PPB bits are changeable. There is only one PPB Lock Bit per device. The PPB Lock Bit is cleared to the unfreeze state at power-up or hardware reset.

Configuring the PPB Lock Bit to the freeze state requires approximately $t_{WC}$. Reading the PPB Lock Status bit requires the initial access time ($t_{ACC}$) of the device.

**Table 12. Sector Protection Schemes**

| Protection States | | | Sector State |
|---|---|---|---|
| DYB Bit | PPB Bit | PPB Lock Bit | |
| Unprotect | Unprotect | Unfreeze | Unprotected – PPB and DYB are changeable |
| Unprotect | Unprotect | Freeze | Unprotected – PPB not changeable, DYB is changeable |
| Unprotect | Protect | Unfreeze | Protected – PPB and DYB are changeable |
| Unprotect | Protect | Freeze | Protected – PPB not changeable, DYB is changeable |
| Protect | Unprotect | Unfreeze | Protected – PPB and DYB are changeable |
| Protect | Unprotect | Freeze | Protected – PPB not changeable, DYB is changeable |
| Protect | Protect | Unfreeze | Protected – PPB and DYB are changeable |
| Protect | Protect | Freeze | Protected – PPB not changeable, DYB is changeable |

Table 12 contains all possible combinations of the DYB bit, PPB bit, and PPB Lock Bit relating to the status of the sector. In summary, if the PPB bit is set, and the PPB Lock Bit is set, the sector is protected and the protection cannot be removed until the next power cycle or hardware reset clears the PPB Lock Bit to unfreeze state. If the PPB bit is cleared, the sector can be dynamically locked or unlocked. The DYB bit then controls whether or not the sector is protected or unprotected. If the user attempts to program or erase a protected sector, the device ignores the command and returns to read mode. A program or erase command to a protected sector enables status polling for $t_{DP}$ before the device returns to read mode without having modified the contents of the protected sector. The programming of the DYB bit, PPB bit, and PPB Lock Bit for a given sector can be verified by writing a DYB Status Read, PPB Status Read, and PPB Lock Status Read commands to the device.

The Autoselect Sector Protection Verification outputs the OR function of the DYB bit and PPB bit per sector basis. When the OR function of the DYB bit and PPB bit is a 1, the sector is either protected by DYB or PPB or both. When the OR function of the DYB bit and PPB bit is a 0, the sector is unprotected through both the DYB and PPB.



Downloaded from Arrow.com.

## 8.14 Password Sector Protection

The Password Sector Protection method allows an even higher level of security than the Persistent Sector Protection method. There are two main differences between the Persistent Sector Protection and the Password Sector Protection methods:

- When the device is first powered on, or comes out of a reset cycle, the PPB Lock Bit is set to the locked state, or the freeze state, rather than cleared to the unlocked state, or the unfreeze state.
- The only means to clear and unfreeze the PPB Lock Bit is by writing a unique 64-bit Password to the device.

The Password Sector Protection method is otherwise identical to the Persistent Sector Protection method.

A 64-bit password is the only additional tool utilized in this method.

The password is stored in a one-time programmable (OTP) region outside of the flash memory. Once the Password Protection Mode Lock Bit is set, the password is permanently set with no means to read, program, or erase it. The password is used to clear and unfreeze the PPB Lock Bit. The Password Unlock command must be written to the flash, along with a password. The flash device internally compares the given password with the pre-programmed password. If they match, the PPB Lock Bit is cleared to the unfreezed *state*, and the PPB bits can be altered. If they do not match, the flash device does nothing. There is a built-in $t_{PPB}$ delay for each password check after the valid 64-bit password is entered for the PPB Lock Bit to be cleared to the unfreezed state. This delay is intended to thwart any efforts to run a program that tries all possible combinations in order to crack the password.

## 8.15 Password and Password Protection Mode Lock Bit

In order to select the Password Sector Protection method, the user must first program the password. Cypress recommends that the password be somehow correlated to the unique Electronic Serial Number (ESN) of the particular flash device. Each ESN is different for every flash device; therefore each password should be different for every flash device. While programming in the password region, the customer may perform Password Read operations. Once the desired password is programmed in, the customer must then set the Password Protection Mode Lock Bit. This operation achieves two objectives:

1. It permanently sets the device to operate using the Password Protection Mode. It is not possible to reverse this function.
2. It also disables all further commands to the password region. All program, and read operations are ignored.

Both of these objectives are important, and if not carefully considered, may lead to unrecoverable errors. The user must be sure that the Password Sector Protection method is desired when programming the Password Protection Mode Lock Bit. More importantly, the user must be sure that the password is correct when the Password Protection Mode Lock Bit is programmed. Due to the fact that read operations are disabled, there is no means to read what the password is afterwards. If the password is lost after programming the Password Protection Mode Lock Bit, there is no way to clear and unfreeze the PPB Lock Bit. The Password Protection Mode Lock Bit, once programmed, prevents reading the 64-bit password on the DQ bus and further password programming. The Password Protection Mode Lock Bit is not erasable. Once Password Protection Mode Lock Bit is programmed, the Persistent Protection Mode Lock Bit is disabled from programming, guaranteeing that no changes to the protection scheme are allowed.

### 8.15.1 64-Bit Password

The 64-bit password is located in its own memory space and is accessible through the use of the Password Program and Password Read commands. The password function works in conjunction with the Password Protection Mode Lock Bit, which when programmed, prevents the Password Read command from reading the contents of the password on the pins of the device.

## 8.16 Persistent Protection Bit Lock (PPB Lock Bit)

A global volatile bit. The PPB Lock Bit is a volatile bit that reflects the state of the Password Protection Mode Lock Bit after power-up reset. If the Password Protection Mode Lock Bit is also programmed after programming the Password, the Password Unlock command must be issued to clear and unfreeze the PPB Lock Bit after a hardware reset (RESET# asserted) or a power-up reset. Successful execution of the Password Unlock command clears and unfreezes the PPB Lock Bit, allowing for sector PPB bits to be modified. Without issuing the Password Unlock command, while asserting RESET#, taking the device through a power-on reset, or issuing the PPB Lock Bit Set command sets the PPB Lock Bit to a the freeze state.

If the Password Protection Mode Lock Bit is not programmed, the device defaults to Persistent Protection Mode. In the Persistent Protection Mode, the PPB Lock Bit is cleared to the unfreeze state after power-up or hardware reset. The PPB Lock Bit is set to the freeze state by issuing the PPB Lock Bit Set command. Once set to the freeze state the only means for clearing the PPB Lock Bit to the unfreeze state is by issuing a hardware or power-up reset. The Password Unlock command is ignored in Persistent Protection Mode.

Reading the PPB Lock Bit requires the initial access time ($t_{ACC}$) of the device.



Downloaded from Arrow.com.

## 8.17 Secure Silicon Region Flash Memory

The Secure Silicon Region feature provides a flash memory region that enables permanent part identification through an Electronic Serial Number (ESN). The Secure Silicon Region is 256 bytes in length, and uses a Secure Silicon Region Indicator Bit (DQ7) in Autoselect Mode to indicate whether or not the Secure Silicon Region is locked when shipped from the factory. This bit is permanently set at the factory and cannot be changed, which prevents cloning of a factory locked part. This ensures the security of the ESN once the product is shipped to the field.

The factory offers the device with the Secure Silicon Region either customer lockable (standard shipping option) or factory locked (contact a sales representative for ordering information). The customer-lockable version is shipped with the Secure Silicon Region unprotected, allowing customers to program the sector after receiving the device. The customer-lockable version also has the Secure Silicon Region Indicator Bit permanently set to a 0. The factory-locked version is always protected when shipped from the factory, and has the Secure Silicon Region Indicator Bit permanently set to a 1. Thus, the Secure Silicon Region Indicator Bit prevents customer-lockable devices from being used to replace devices that are factory locked.

The Secure Silicon Region address space in this device is allocated as follows:

| Secure Silicon Region Address Range | Customer Lockable | ESN Factory Locked | ExpressFlash Factory Locked |
|---|---|---|---|
| 000000h–000007h | Determined by customer | ESN | ESN or determined by customer |
| 000008h–00007Fh | | Unavailable | Determined by customer |

The system accesses the Secure Silicon Region through a command sequence (see Table 21 and Table 23). After the system has written the Enter Secure Silicon Region command sequence, it may read the Secure Silicon Region by using the addresses normally occupied by the first sector (SA0). This mode of operation continues until the system issues the Exit Secure Silicon Region command sequence, Reset / ASO Exit command, or until power is removed from the device. On power-up, or following a hardware reset, the device reverts to sending commands to sector SA0.

### 8.17.1 Customer Lockable: Secure Silicon Region NOT Programmed or Protected At the Factory

Unless otherwise specified, the device is shipped such that the customer may program and protect the 256-byte Secure Silicon Region.

The system may program the Secure Silicon Region using the write-buffer method, in addition to the standard programming command sequence. See Command Definitions on page 30. *Note that the ACC function and unlock bypass modes are not available when the Secure Silicon Region is enabled.*

Programming and protecting the Secure Silicon Region must be used with caution since, once protected, there is no procedure available for unprotecting the Secure Silicon Region area and none of the bits in the Secure Silicon Region memory space can be modified in any way.

The Secure Silicon Region area can be protected using one of the following procedures:

- Write the three-cycle Enter Secure Silicon Region command.
- To verify the protect / unprotect status of the Secure Silicon Region, follow the algorithm.

Once the Secure Silicon Region is programmed, locked and verified, the system must write the Exit Secure Silicon Region command sequence or Reset / ASO Exit command to return to reading and writing within the remainder of the array.



Downloaded from Arrow.com.

### 8.17.2 Factory Locked: Secure Silicon Region Programmed and Protected At the Factory

In devices with an ESN, the Secure Silicon Region is protected when the device is shipped from the factory. The Secure Silicon Region cannot be modified in any way. An ESN Factory Locked device has an 16-byte random ESN at addresses 000000h–000007h. Please contact your sales representative for details on ordering ESN Factory Locked devices.

Customers may opt to have their code programmed by the factory through the ExpressFlash service (Express Flash Factory Locked). The devices are then shipped from the factory with the Secure Silicon Region permanently locked. Contact your sales representative for details on using the ExpressFlash service.

## 8.18 Write Protect (WP#/ACC)

The Write Protect function provides a hardware method of protecting the first or last sector for Uniform Sector Model or it protects the first or last two sectors for the Boot Sector Model without using $V_{ID}$. Write Protect is one of two functions provided by the WP#/ACC input.

If the system asserts $V_{IL}$ on the WP#/ACC pin, the device disables program and erase functions in the first or last sector independently of whether those sectors were protected or unprotected. *Note that if WP#/ACC is at $V_{IL}$ when the device is in the standby mode, the maximum input load current is increased.* See the table in DC Characteristicst on page 77.

If the system asserts $V_{IH}$ on the WP#/ACC pin, the device reverts to whether the protected sectors previously set to be protected or unprotected using the method described in Section 8.11–8.16. *Note that WP#/ACC contains an internal pull-up; when unconnected, WP#/ACC is at $V_{IH}$.*

## 8.19 Hardware Data Protection

The command sequence requirement of unlock cycles for programming or erasing provides data protection against inadvertent writes (refer to Table 21 on page 42 and Table 23 on page 45 for command definitions). In addition, the following hardware data protection measures prevent accidental erasure or programming, which might otherwise be caused by spurious system level signals during $V_{CC}$ power-up and power-down transitions, or from system noise.

### 8.19.1 Low $V_{CC}$ Write Inhibit

When $V_{CC}$ is less than $V_{LKO}$, the device does not accept any write cycles. This protects data during $V_{CC}$ power-up and power-down. The command register and all internal program / erase circuits are disabled, and the device resets to the read mode. Subsequent writes are ignored until $V_{CC}$ is greater than $V_{LKO}$. The system must provide the proper signals to the control pins to prevent unintentional writes when $V_{CC}$ is greater than $V_{LKO}$.

### 8.19.2 Write Pulse *Glitch* Protection

Noise pulses of less than 5 ns (typical) on OE#, CE# or WE# do not initiate a write cycle.

### 8.19.3 Logical Inhibit

Write cycles are inhibited by holding any one of OE# = $V_{IL}$, CE# = $V_{IH}$ or WE# = $V_{IH}$. To initiate a write cycle, CE# and WE# must be a logical zero while OE# is a logical one.

### 8.19.4 Power-Up Write Inhibit

If WE# = CE# = $V_{IL}$ and OE# = $V_{IH}$ during power up, the device does not accept commands on the rising edge of WE#. The internal state machine is automatically reset to the read mode on power-up.



Downloaded from Arrow.com.

# 9. Common Flash Memory Interface (CFI)

The Common Flash Interface (CFI) specification outlines device and host system software interrogation handshake, which allows specific vendor-specified software algorithms to be used for entire families of devices. Software support can then be device-independent, JEDEC ID-independent, and forward- and backward-compatible for the specified flash device families. Flash vendors can standardize their existing interfaces for long-term compatibility.

This device enters the CFI Query mode when the system writes the CFI Query command, 98h, to address 55h, any time the device is ready to read array data. The system can read CFI information at the addresses given in Table 13 on page 27 to Table 16 on page 29. To terminate reading CFI data, the system must write the reset command (0xF0) or 0xFF.

The system can also write the CFI query command when the device is in the autoselect mode. The device enters the CFI query mode, and the system can read CFI data at the addresses given in Table 13 on page 27 to Table 16 on page 29. The system must write the reset command to return the device to reading array data.

For further information, please refer to the CFI Specification and CFI Publication 100. Alternatively, contact your sales representative for copies of these documents.

**Table 13. CFI Query Identification String**

| Addresses (x16) | Addresses (x8) | Data | Description |
|---|---|---|---|
| 10h<br>11h<br>12h | 20h<br>22h<br>24h | 0051h<br>0052h<br>0059h | Query Unique ASCII string “QRY” |
| 13h<br>14h | 26h<br>28h | 0002h<br>0000h | Primary OEM Command Set |
| 15h<br>16h | 2Ah<br>2Ch | 0040h<br>0000h | Address for Primary Extended Table |
| 17h<br>18h | 2Eh<br>30h | 0000h<br>0000h | Alternate OEM Command Set (00h = none exists) |
| 19h<br>1Ah | 32h<br>34h | 0000h<br>0000h | Address for Alternate OEM Extended Table (00h = none exists) |

**Table 14. System Interface String**

| Addresses (x16) | Addresses (x8) | Data | Description |
|---|---|---|---|
| 1Bh | 36h | 0027h | $V_{CC}$ Min. (write / erase)<br>D7–D4: volt, D3–D0: 100 millivolt |
| 1Ch | 38h | 0036h | $V_{CC}$ Max. (write / erase)<br>D7–D4: volt, D3–D0: 100 millivolt |
| 1Dh | 3Ah | 0000h | $V_{PP}$ Min. voltage (00h = no $V_{PP}$ pin present) |
| 1Eh | 3Ch | 0000h | $V_{PP}$ Max. voltage (00h = no $V_{PP}$ pin present) |
| 1Fh | 3Eh | 0008h | Typical timeout per single write $2^N$ µs |
| 20h | 40h | 0008h | Typical timeout for Min. size buffer write $2^N$ µs<br>(00h = not supported) |
| 21h | 42h | 0009h | Typical timeout per individual block erase $2^N$ ms |
| 22h | 44h | 0010h | Typical timeout for full chip erase $2^N$ ms (00h = not supported) |
| 23h | 46h | 0003h | Max. timeout for byte / word program $2^N$ times typical. |
| 24h | 48h | 0003h | Max. timeout for buffer write $2^N$ times typical |
| 25h | 4Ah | 0001h | Max. timeout per individual block erase $2^N$ times typical |
| 26h | 4Ch | 0000h | Max. timeout for full chip erase $2^N$ times typical<br>(00h = not supported) |

***Note:***
*CFI data related to $V_{CC}$ and time-outs may differ from actual $V_{CC}$ and time-outs of the product. Please consult the Ordering Information tables to obtain the $V_{CC}$ range for particular part numbers. Please consult the Erase and Programming Performance table for typical timeout specifications.*



Downloaded from Arrow.com.

**Table 15. Device Geometry Definition**

| Addresses (x16) | Addresses (x8) | Data | Description |
|---|---|---|---|
| 27h | 4Eh | 0017h | Device Size = $2^N$ byte |
| 28h<br>29h | 50h<br>52h | 000xh<br>0000h | Flash Device Interface description (refer to CFI publication 100)<br>0001h = x16-only bus devices<br>0002h = x8/x16 bus devices |
| 2Ah<br>2Bh | 54h<br>56h | 0008h<br>0000h | Max. number of byte in multi-byte write = $2^N$<br>(00h = not supported) |
| 2Ch | 58h | 00xxh | Number of Erase Block Regions within device<br>01h = uniform device<br>02h = boot device |
| 2Dh<br>2Eh<br>2Fh<br>30h | 5Ah<br>5Ch<br>5Eh<br>60h | 00xxh<br>0000h<br>00x0h<br>000xh | Erase Block Region 1 Information<br>(refer to the CFI specification or CFI publication 100)<br>007Fh, 0000h, 0000h, 0001h = 64 Mb (01, 02, 06, 07, V1, V2, V6, V7)<br>0007h, 0000h, 0020h, 0000h = 64 Mb (03, 04) |
| 31h<br>32h<br>33h<br>34h | 60h<br>64h<br>66h<br>68h | 00xxh<br>0000h<br>0000h<br>000xh | Erase Block Region 2 Information (refer to CFI publication 100)<br>0000h, 0000h, 0000h, 0000h = 64 Mb (01, 02, 06, 07, V1, V2, V6, V7)<br>007Eh, 0000h, 0000h, 0001h = 64 Mb (03, 04) |
| 35h<br>36h<br>37h<br>38h | 6Ah<br>6Ch<br>6Eh<br>70h | 0000h<br>0000h<br>0000h<br>0000h | Erase Block Region 3 Information (refer to CFI publication 100) |
| 39h<br>3Ah<br>3Bh<br>3Ch | 72h<br>74h<br>76h<br>78h | 0000h<br>0000h<br>0000h<br>0000h | Erase Block Region 4 Information (refer to CFI publication 100) |
| 3Dh<br>3Eh<br>3Fh | 7Ah<br>7Ch<br>7Eh | FFFFh<br>FFFFh<br>FFFFh | Reserved |



Downloaded from Arrow.com.

**Table 16. Primary Vendor-Specific Extended Query**

| Addresses (x16) | Addresses (x8) | Data | Description |
|---|---|---|---|
| 40h<br>41h<br>42h | 80h<br>82h<br>84h | 0050h<br>0052h<br>0049h | Query-unique ASCII string “PRI” |
| 43h | 86h | 0031h | Major version number, ASCII |
| 44h | 88h | 0033h | Minor version number, ASCII |
| 45h | 8Ah | 0020h | Address Sensitive Unlock (Bits 1–0)<br>0 = Required<br>1 = Not Required<br>Process Technology (Bits 5–2) 1000b = 65 nm MirrorBit<br>Reserved (Bits 7-6) |
| 46h | 8Ch | 0002h | Erase Suspend<br>0 = Not Supported<br>1 = To Read Only<br>2 = To Read and Write |
| 47h | 8Eh | 0001h | Sector Protect<br>0 = Not Supported<br>X = Number of sectors in smallest sector |
| 48h | 90h | 0000h | Sector Temporary Unprotect<br>00 = Not Supported<br>01 = Supported |
| 49h | 92h | 0008h | Sector Protect / Unprotect scheme<br>0008h = Advanced sector Protection |
| 4Ah | 94h | 0000h | Simultaneous Operation<br>00 = Not Supported<br>X = Number of Sectors in Bank |
| 4Bh | 96h | 0000h | Burst Mode Type<br>00 = Not Supported<br>01 = Supported |
| 4Ch | 98h | 0002h | Page Mode Type<br>02 = 8 Word Page |
| 4Dh | 9Ah | 00B5h | ACC (Acceleration) Supply Minimum<br>00h = Not Supported<br>D7-D4: Volt<br>D3-D0: 100 mV |
| 4Eh | 9Ch | 00C5h | ACC (Acceleration) Supply Maximum<br>00h = Not Supported<br>D7-D4: Volt<br>D3-D0: 100 mV |
| 4Fh | 9Eh | 00xxh | Top / Bottom Boot Sector Flag<br>02h = Bottom Boot Device<br>03h = Top Boot Device<br>04h = Uniform sectors bottom WP# protect<br>05h = Uniform sectors top WP# protect |
| 50h | A0h | 0001h | Program Suspend<br>00h = Not Supported<br>01h = Supported |



Downloaded from Arrow.com.

# 10. Command Definitions

Writing specific address and data commands or sequences into the command register initiates device operations. Table 21 on page 42 and Table 23 on page 45 define the valid register command sequences. **Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state**. A reset command is then required to return the device to reading array data.

All addresses are latched on the falling edge of WE# or CE#, whichever happens later. All data is latched on the rising edge of WE# or CE#, whichever happens first. Refer to AC Characteristics on page 84 for timing diagrams.

## 10.1 Reading Array Data

The device is automatically set to reading array data after device power-up. No commands are required to retrieve data. The device is ready to read array data after completing an Embedded Program or Embedded Erase algorithm.

After the device accepts an Erase Suspend command, the device enters the erase-suspend-read mode, after which the system can read data from any non-erase-suspended sector. After completing a programming operation in the Erase Suspend mode, the system may once again read array data with the same exception. See Erase Suspend / Erase Resume Commands on page 39 for more information.

The system *must* issue the reset command to return the device to the read (or erase-suspend-read) mode if DQ5 goes high during an active program or erase operation, or if the device is in the autoselect mode. See *Reset Command* below for more information.

See also Requirements for Reading Array Data on page 15 for more information. The Read-Only Operations – AC Characteristics on page 84 provide the read parameters, and Figure 22 on page 87 shows the timing diagram.

## 10.2 Reset Command

Writing the reset command resets the device to the read or erase-suspend-read mode. Address bits are don't cares for this command.

The reset command may be written between the sequence cycles in an erase command sequence before erasing begins. This resets the device to the read mode. Once erasure begins, however, the device ignores reset commands until the operation is complete.

The reset command may be written between the sequence cycles in a program command sequence before programming begins. This resets the device to the read mode. If the program command sequence is written while the device is in the Erase Suspend mode, writing the reset command returns the device to the erase-suspend-read mode. Once programming begins, however, the device ignores reset commands until the operation is complete.

The reset command may be written between the sequence cycles in an autoselect command sequence. Once in the autoselect mode, the reset command must be written to return to the read mode. If the device entered the autoselect mode while in the Erase Suspend mode, writing the reset command returns the device to the erase-suspend-read mode.

If DQ5 goes high during a program or erase operation, writing the reset command returns the device to the read mode (or erase-suspend-read mode if the device was in Erase Suspend).

*Note that if DQ1 goes high during a Write Buffer Programming operation, the system must write the Write-to-Buffer-Abort Reset command sequence to reset the device for the next operation.*



Downloaded from Arrow.com.

## 10.3 Autoselect Command Sequence

The autoselect command sequence allows the host system to read several identifier codes at specific addresses:

| Identifier Code | A7:A0 (x16) | A6:A-1 (x8) |
|---|---|---|
| Manufacturer ID | 00h | 00h |
| Device ID, Cycle 1 | 01h | 02h |
| Device ID, Cycle 2 | 0Eh | 1Ch |
| Device ID, Cycle 3 | 0Fh | 1Eh |
| Secure Silicon Region Factory Protect | 03h | 06h |
| Sector Protect Verify | (SA)02h | (SA)04h |

***Note:***
*The device ID is read over three cycles. SA = Sector Address.*

The autoselect command sequence is initiated by first writing on unlock cycle (two cycles). This is followed by a third write cycle that contains the autoselect command. The device then enters the autoselect mode. The system may read at any address any number of times without initiating another autoselect command sequence:

The system must write the reset command to return to the read mode (or erase-suspend-read mode if the device was previously in Erase Suspend).

## 10.4 Status Register ASO

The Status Register ASO contains a single word of registered volatile status for Embedded Algorithms. When the Status Register Read command is issued, the current status is captured by the register and the ASO is entered. The Status Register content appears at all word locations in the device address space. However, it is recommended to read the status only at word location 0 for future compatibility. The first read access in the Status Register ASO or a Software Reset / ASO Exit write command exits the ASO and returns to the address space map in use when the Status Register read command was issued. It is not recommended to perform any other command after the Status Register Read command is given and before the Status Register ASO is exited.

## 10.5 Enter / Exit Secure Silicon Region Command Sequence

The Secure Silicon Region provides a secured data area containing an 8-word / 16-byte random Electronic Serial Number (ESN). The system can access the Secure Silicon Region by issuing the three-cycle Enter Secure Silicon Region command sequence. The device continues to access the Secure Silicon Region until the system issues the four-cycle Exit Secure Silicon Region command sequence or Reset / ASO Exit command which returns the device to normal operation. Table 21 on page 42 and Table 23 on page 45 show the address and data requirements for both command sequences. See also Secure Silicon Region Flash Memory on page 25 for further information. *Note that the ACC function and unlock bypass modes are not available when the Secure Silicon Region is enabled.*

## 10.6 ECC Status ASO

The system can access the ECC status ASO by issuing the ECC status entry command sequence during Read Mode. The ECC Status ASO provides the status of the ECC function, enabled or disabled, or if the ECC function corrected a single-bit error when reading the selected page. Section 8.4, *Automatic ECC on page 16* describes the ECC function in greater detail.

The ECC Status ASO allows the following activities:

- Read ECC Status for the selected page.
- ASO exit.



Downloaded from Arrow.com.

### 10.6.1 ECC Status

The contents of the ECC Status ASO indicate, for the selected page, whether the ECC logic has corrected an error in the eight bit ECC code, in the 32-byte page of data, or that ECC is disabled for that page. The address specified in the ECC Status Read Command, provided in Table 21, *Command Definitions (x16 Mode, BYTE# = $V_{IH}$) on page 42* and Table 23, *Command Definitions (x8 Mode, BYTE# = $V_{IL}$) on page 45*, selects the desired ECC page.

**Table 17. ECC Status Word - Upper Byte**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Name | RFU | RFU | RFU | RFU | RFU | RFU | RFU | RFU |
| Value | X | X | X | X | X | X | X | X |

**Table 18. ECC Status Word - Lower Byte**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | RFU | RFU | RFU | RFU | ECC Enabled on 16-Word Page | Single Bit Error Corrected in ECC Bits | Singe Bit Error Corrected in Data Bits | RFU |
| Value | X | X | X | X | 0 = ECC Enabled<br>1 = ECC Disabled | 0 = No Error Corrected<br>1 = Single Bit Error Corrected | 0 = No Error Corrected<br>1 = Single Bit Error Corrected | X |

## 10.7 Word Program Command Sequence

Programming is a four-bus-cycle operation. The program command sequence is initiated by writing two unlock write cycles, followed by the program set-up command. The program address and data are written next, which in turn initiate the Embedded Program algorithm. The system is not required to provide further controls or timings. The device automatically provides internally generated program pulses and verifies the programmed cell margin. Table 21 on page 42 and Table 23 on page 45 show the address and data requirements for the word program command sequence, respectively.

When the Embedded Program algorithm is complete, the device then returns to the read mode and addresses are no longer latched. The system can determine the status of the program operation by using DQ7 or DQ6. Refer to Write Operation Status on page 51 for information on these status bits. Any commands written to the device during the Embedded Program Algorithm are ignored. *Note that the Secure Silicon Region, autoselect, and CFI functions are unavailable when a program operation is in progress. Note that a hardware reset immediately terminates the program operation.* The program command sequence should be reinitiated once the device returns to the read mode, to ensure data integrity.

Programming is allowed in any sequence of address locations and across sector boundaries. Programming to the same word address multiple times without intervening erases (incremental bit programming) requires a modified programming method. For such application requirements, please contact your local Cypress representative. Word programming is supported for backward compatibility with existing flash driver software and for occasional writing of individual words. Use of write buffer programming (see below) is strongly recommended for general programming use when more than a few words are to be programmed.

**Any bit in a word cannot be programmed from 0 back to a 1.** Attempting to do so may cause DQ7 and DQ6 status bits to indicate the operation was successful. However, a succeeding read shows that the data is still 0. Only erase operations can convert a 0 to a 1.

## 10.8 Unlock Bypass Command Sequence

This device features an Unlock Bypass mode to facilitate shorter programming and erase commands. Once the device enters the Unlock Bypass mode, only two write cycles are required to program or erase data, instead of the normal four or six cycles, respectively.

The unlock bypass command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the unlock bypass command, 20h. The device then enters the unlock bypass mode.

This mode dispenses with the initial two unlock cycles required in the standard program sequence and four unlock cycles in the standard erase command sequence, resulting in faster total programming and erase times.Table 21 on page 42 and Table 23 on page 45 show the requirements for the unlock bypass command sequences.



Downloaded from Arrow.com.

During the unlock bypass mode, only the Read, Program, Write Buffer Programming, Write-to-Buffer-Abort Reset, Unlock Bypass Sector Erase, Unlock Bypass Chip Erase and Unlock Bypass Reset commands are valid. To exit the unlock bypass mode, the system must issue the two-cycle unlock bypass reset command sequence. The first cycle address is 'don't care' and the data 90h. The second cycle need only contain the data 00h. The sector then returns to the read mode.

## 10.9 Write Buffer Programming

Write Buffer Programming allows the system write to a maximum of 128 words / 256 bytes in one programming operation. This results in faster effective programming time than the standard programming algorithms. The Write Buffer Programming command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the Write Buffer Load command written at the Sector Address in which programming occurs. The fourth cycle writes the sector address and the number of word locations, minus one, to be programmed. For example, if the system programs six unique address locations, then 05h should be written to the device. This tells the device how many write buffer addresses are loaded with data and therefore when to expect the Program Buffer to Flash command. The number of locations to program cannot exceed the size of the write buffer or the operation aborts.

The fifth cycle writes the first address location and data to be programmed. The write-buffer-page is selected by address bits $A_{MAX}$–$A_7$. All subsequent address / data pairs must fall within the selected-write-buffer-page. The system then writes the remaining address / data pairs into the write buffer. Write buffer locations may be loaded in any order.

The write-buffer-page address must be the same for all address / data pairs loaded into the write buffer. (This means Write Buffer Programming cannot be performed across multiple write-buffer pages.) This also means that Write Buffer Programming cannot be performed across multiple sectors. If the system attempts to load programming data outside of the selected write-buffer page, the operation aborts.

*Note that if a Write Buffer address location is loaded multiple times, the address / data pair counter is decremented for every data load operation. The host system must therefore account for loading a write-buffer location more than once. The counter decrements for each data load operation, not for each unique write-buffer-address location. Note also that if an address location is loaded more than once into the buffer, the final data loaded for that address is programmed.*

Once the specified number of write buffer locations are loaded, the system must then write the Program Buffer to Flash command at the sector address. Any other address and data combination aborts the Write Buffer Programming operation. The device then begins programming. Data polling should be used while monitoring the last address location loaded into the write buffer. DQ7, DQ6, DQ5, and DQ1 should be monitored to determine the device status during Write Buffer Programming.

The write-buffer programming operation can be suspended using the standard program suspend / resume commands. Upon successful completion of the Write Buffer Programming operation, the device is ready to execute the next command.

The Write Buffer Programming Sequence can be aborted in the following ways:

- Load a value that is greater than the page buffer size during the Number of Locations to Program step.
- Write to an address in a sector different than the one specified during the Write-Buffer-Load command.
- Write an Address / Data pair to a different write-buffer-page than the one selected by the Starting Address during the write buffer data loading stage of the operation.
- Write data other than the Confirm Command after the specified number of data load cycles.

The abort condition is indicated by DQ1 = 1, DQ7 = DATA# (for the last address location loaded), DQ6 = toggle, and DQ5 = 0. A Write-to-Buffer-Abort Reset command sequence must be written to reset the device for the next operation.

*Note that the Secure Silicon Region, autoselect, and CFI functions are unavailable when a program operation is in progress.* This flash device is capable of handling multiple write buffer programming operations on the same write buffer address range without intervening erases. For applications requiring incremental bit programming, a modified programming method is required; please contact your local Cypress representative. **Any bit in a write buffer address range cannot be programmed from 0 back to a 1.** Attempting to do so may cause the device to set DQ5 = 1, of cause the DQ7 and DQ6 status bits to indicate the operation was successful. However, a succeeding read shows that the data is still 0. Only erase operations can convert a 0 to a 1.



Downloaded from Arrow.com.

## 10.10 Accelerated Program

The device supports program operations when the system asserts $V_{HH}$ on the WP#/ACC or ACC pin. When WP#/ACC or ACC pin is lowered back to $V_{IH}$ or $V_{IL}$ the device exits the Accelerated Programming mode and returns to normal operation. The WP#/ACC is $V_{HH}$ tolerant but is not designed to accelerate the program functions. If the system asserts $V_{HH}$ on this input, the device automatically enters the Unlock Bypass mode. The system can then use the Write Buffer Load command sequence provided by the Unlock Bypass mode. *Note that if a Write-to-Buffer-Abort Reset is required while in Unlock Bypass mode, the full 3-cycle RESET command sequence must be used to reset the device. Note that the WP#/ACC pin must not be at $V_{HH}$ for operations other than accelerated programming, or device damage may result. WP# contains an internal pull-up; when unconnected, WP# is at $V_{IH}$. Accelerated programming is supported at room temperature only.*

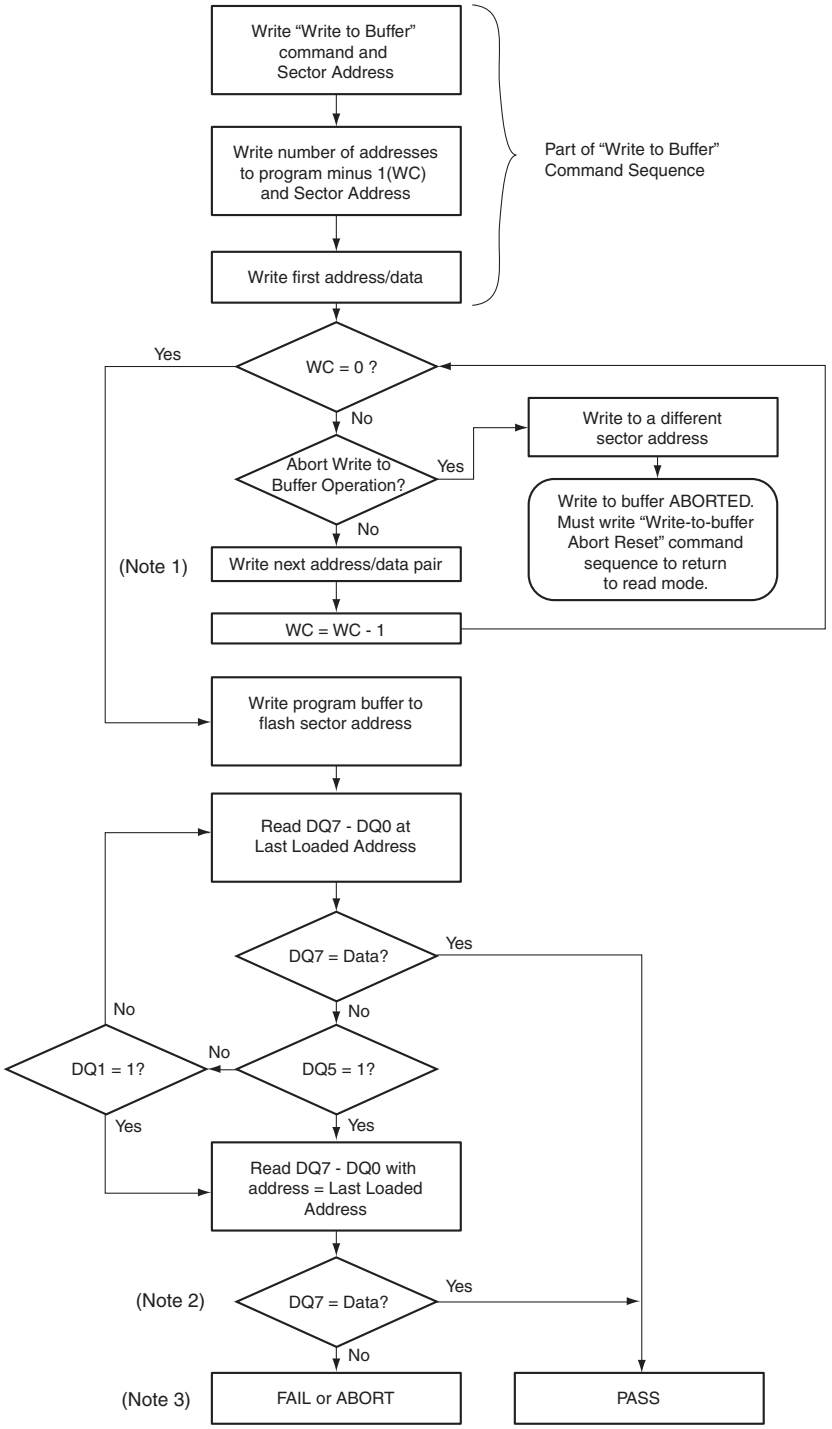
Figure 8 on page 35 illustrates the algorithm for the program operation. Refer to Table 71, *Erase / Program Operations on page 92* for parameters, and Figure 33 on page 93 for timing diagrams.

- Sectors must be unlocked prior to raising WP#/ACC to $V_{HH}$.
- It is recommended that WP#/ACC apply $V_{HH}$ after power-up sequence is completed. In addition, it is recommended that WP#/ACC apply from $V_{HH}$ to $V_{IH}/V_{IL}$ before powering down $V_{CC}/V_{IO}$.



Downloaded from Arrow.com.

**Figure 8. Write Buffer Programming Operation**



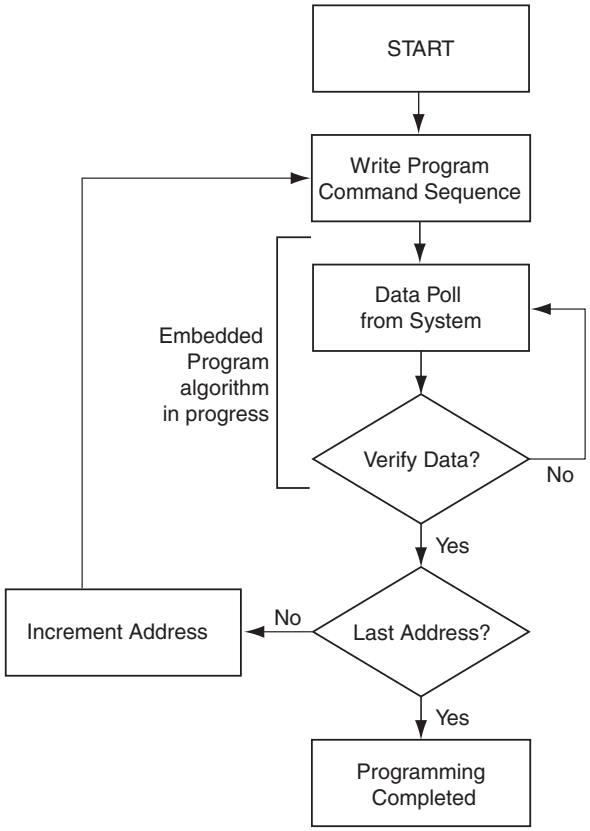

***Notes:***

1. *When Sector Address is specified, any address in the selected sector is acceptable. However, when loading Write-Buffer address locations with data, all addresses must fall within the selected Write-Buffer Page.*
2. *DQ7 may change simultaneously with DQ5. Therefore, DQ7 should be verified.*
3. *If this flowchart location was reached because DQ5= 1, then the device Failed. If this flowchart location was reached because DQ1= 1, then the Write to Buffer operation was Aborted. In either case, the proper reset command must be written before the device can begin another operation. If DQ1= 1, write the Write-Buffer-Programming-Abort-Reset command. if DQ5= 1, write the Reset command.*
4. *See Table 21 on page 42 and Table 23 on page 45 for command sequences required for write buffer programming.*



Downloaded from Arrow.com.

**Figure 9. Program Operation**




***Note:***
*See Table 21 on page 42 and Table 23 on page 45 for program command sequence.*

## 10.11 Program Suspend / Program Resume Command Sequence

The Program Suspend command allows the system to interrupt a programming operation or a Write to Buffer programming operation so that data can be read from any non-suspended sector. When the Program Suspend command is written during a programming process, the device halts the program operation within $t_{PSL}$ (program suspend latency) and updates the status bits. Addresses are not required when writing the Program Suspend command.

There are two commands available for program suspend. The legacy combined Erase / Program suspend command (B0h command code) and the separate Program Suspend command (51h command code). There are also two commands for Program resume. The legacy combined Erase / Program resume command (30h command code) and the separate Program Resume command (50h command code). It is recommended to use the separate program suspend and resume commands for programming and use the legacy combined command only for erase suspend and resume.

After the programming operation is suspended, the system can read array data from any non-suspended sector. The Program Suspend command may also be issued during a programming operation while an erase is suspended. In this case, data may be read from any addresses not in Erase Suspend or Program Suspend. If a read is needed from the Secure Silicon Region area (One-time Program area), then user must use the proper command sequences to enter and exit this region. *Note that the Secure Silicon Region, autoselect, and CFI functions are unavailable when a program operation is in progress.*

The system may also write the autoselect command sequence when the device is in the Program Suspend mode. The system can read as many autoselect codes as required. When the device exits the autoselect mode, the device reverts to the Program Suspend mode, and is ready for another valid operation. See Autoselect Command Sequence on page 31 for more information.

After the Program Resume command is written, the device reverts to programming. The system can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard program operation. See Write Operation Status on page 51 for more information.

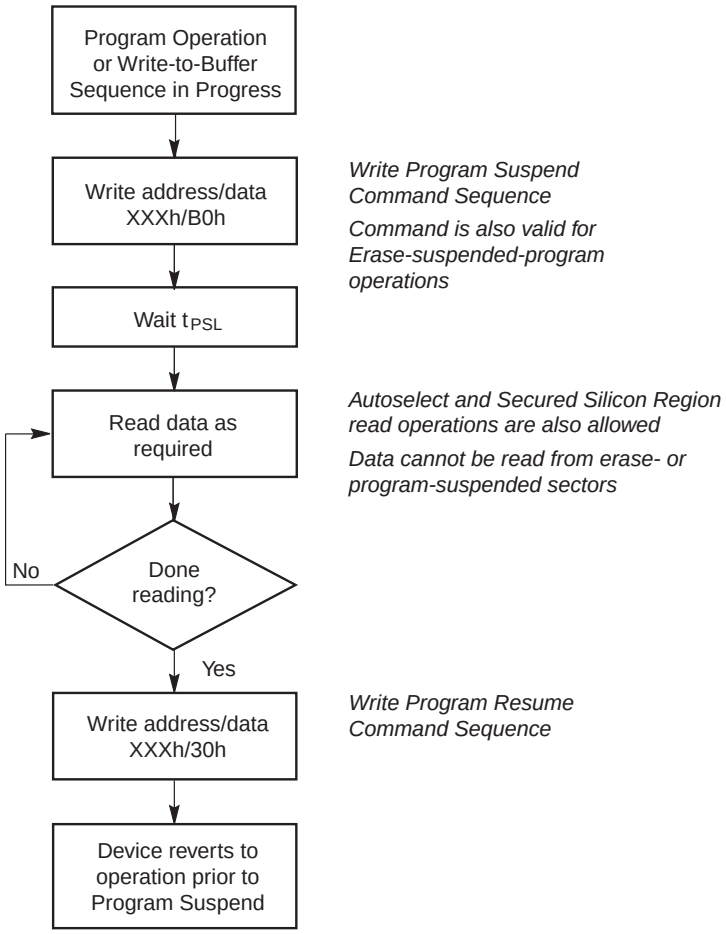


Downloaded from Arrow.com.

The system must write the Program Resume command (address bits are don't care) to exit the Program Suspend mode and continue the programming operation. Further writes of the Resume command are ignored. Another Program Suspend command can be written after the device resumes programming.

Program operations can be interrupted as often as necessary but in order for a program operation to progress to completion there must be some periods of time between resume and the next suspend command greater than or equal to $t_{PRS}$ as listed in Table 73 and Table 74.

**Figure 10. Program Suspend / Program Resume**






Downloaded from Arrow.com.

## 10.12 Chip Erase Command Sequence

Chip erase is a six bus cycle operation. The chip erase command sequence is initiated by writing two unlock cycles, followed by a set-up command. Two additional unlock write cycles are then followed by the chip erase command, which in turn invokes the Embedded Erase algorithm. The device does **not** require the system to pre-program prior to erase. The Embedded Erase algorithm automatically preprograms and verifies the entire memory for an all zero data pattern prior to electrical erase. The system is not required to provide any controls or timings during these operations. Table 21 on page 42 and Table 23 on page 45 show the address and data requirements for the chip erase command sequence.

When the Embedded Erase algorithm is complete, the device returns to the read mode and addresses are no longer latched. The system can determine the status of the erase operation by using DQ7, DQ6, or DQ2. Refer to Write Operation Status on page 51 for information on these status bits.

The Unlock Bypass feature allows the host system to send program commands to the flash device without first writing unlock cycles within the command sequence. See Section 10.8 for details on the Unlock Bypass function.

Any commands written during the chip erase operation are ignored. However, note that a hardware reset immediately terminates the erase operation. If this occurs, the chip erase command sequence should be reinitiated once the device returns to reading array data, to ensure data integrity.

Figure 11 on page 39 illustrates the algorithm for the erase operation. Refer to Table 17.2 on page 88 for parameters, and Figure 34 on page 93 for timing diagrams.

## 10.13 Sector Erase Command Sequence

Sector erase is a six bus cycle operation. The sector erase command sequence is initiated by writing two unlock cycles, followed by a set-up command. Two additional unlock cycles are written, and are then followed by the address of the sector to be erased, and the sector erase command. Table 21 on page 42 and Table 23 on page 45 shows the address and data requirements for the sector erase command sequence.

The device does **not** require the system to pre-program prior to erase. The Embedded Erase algorithm automatically programs and verifies the entire memory for an all zero data pattern prior to electrical erase. The system is not required to provide any controls or timings during these operations.

After the command sequence is written, a sector erase time-out of $t_{SEA}$ occurs. During the time-out period, additional sector addresses and sector erase commands may be written. Invalid commands will be ignored during the time-out period. Loading the sector erase buffer may be done in any sequence, and the number of sectors may be from one sector to all sectors. The time between these additional cycles must be less than 50 µs, otherwise erasure may begin. Any sector erase address and command following the exceeded time-out may or may not be accepted. It is recommended that processor interrupts be disabled during this time to ensure all commands are accepted. The interrupts can be re-enabled after the last Sector Erase command is written. ***Note that the Secure Silicon Region, autoselect, and CFI functions are unavailable when an erase operation is in progress.*** The system must rewrite the command sequence and any additional addresses and commands.

The system can monitor DQ3 to determine if the sector erase timer has timed out (See DQ3: Sector Erase Timer on page 56.). The time-out begins from the rising edge of the final WE# pulse in the command sequence.

If the sector is found to have not completed its last erase successfully, the sector is unconditionally erased. If the last erase was successful, the sector is read to determine if the sector is still erased (blank). The erase operation is started immediately after finding any programmed zero. If the sector is already blank (no programmed zero bit found) the remainder of the erase operation is skipped. This can dramatically reduce erase time when sectors being erased do not need the erase operation. When enabled the blank check feature is used within the parameter erase, sector erase, and bulk erase commands. When blank check is disabled an erase command unconditionally starts the erase operation.

When the Embedded Erase algorithm is complete, the device returns to reading array data and addresses are no longer latched. The system can determine the status of the erase operation by reading DQ7, DQ6, or DQ2 in the erasing sector. Refer to Write Operation Status on page 51 for information on these status bits.

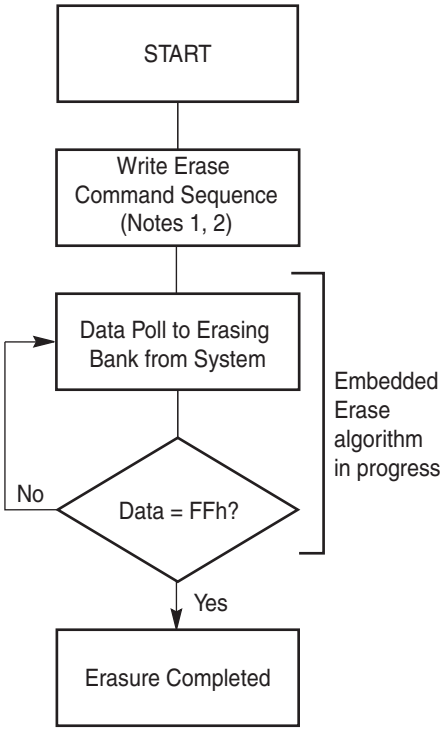
Once the sector erase operation begins, only the Erase Suspend command is valid. All other commands are ignored. However, note that a hardware reset immediately terminates the erase operation. If that occurs, the sector erase command sequence should be reinitiated once the device returns to reading array data, to ensure data integrity.

Figure 11 on page 39 illustrates the algorithm for the erase operation. Refer to Table 17.2 on page 88 for parameters, and Figure 34 on page 93 for timing diagrams.



Downloaded from Arrow.com.

**Figure 11. Erase Operation**




***Notes:***
1. *See Table 21 and Table 23 for program command sequence.*
2. *See DQ3: Sector Erase Timer on page 56 for information on the sector erase timer.*

## 10.14 Erase Suspend / Erase Resume Commands

The Erase Suspend command, B0h, allows the system to interrupt a sector erase operation and then read data from, or program data to, any sector not selected for erasure. This command is valid only during the sector erase operation, including the $t_{ESL}$ time-out period during the sector erase command sequence. The Erase Suspend command is ignored if written during the chip erase operation or Embedded Program algorithm.

When the Erase Suspend command is written during the sector erase operation, the device requires $t_{ESL}$ (erase suspend latency) to suspend the erase operation. However, when the Erase Suspend command is written during the sector erase time-out, the device immediately terminates the time-out period and suspends the erase operation.

After the erase operation is suspended, the device enters the erase-suspend-read mode. The system can read data from or program data to any sector not selected for erasure. (The device *erase suspends* all sectors selected for erasure.) Reading at any address within erase-suspended sectors produces status information on DQ7–DQ0. The system can use DQ7, or DQ6 and DQ2 together, to determine if a sector is actively erasing or is erase-suspended. Refer to Write Operation Status on page 51 for information on these status bits.

After an erase-suspended program operation is complete, the device returns to the erase-suspend-read mode. The system can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard word program operation. Refer to Write Operation Status on page 51 for more information.

In the erase-suspend-read mode, the system can also issue the autoselect command sequence. Refer to the Autoselect Mode on page 19 and Autoselect Command Sequence on page 31 sections for details.

To resume the sector erase operation, the system must write the Erase Resume command. Further writes of the Resume command are ignored. Another Erase Suspend command can be written after the chip resumes erasing.



Downloaded from Arrow.com.

During an erase operation, this flash device performs multiple internal operations which are invisible to the system. When an erase operation is suspended, any of the internal operations that were not fully completed must be restarted. As such, if this flash device is continually issued suspend / resume commands in rapid succession, erase progress is impeded as a function of the number of suspends. The result is a longer cumulative erase time than without suspends. *Note that the additional suspends do not affect device reliability or future performance. In most systems rapid erase / suspend activity occurs only briefly. In such cases, erase performance is not significantly impacted.*

Erase operations can be interrupted as often as necessary but in order for an erase operation to progress to completion there must be some periods of time between resume and the next suspend command greater than or equal to $t_{ERS}$ as listed in Table 73 and Table 74.

## 10.15 Evaluate Erase Status

The Evaluate Erase Status (EES) command verifies that the last erase operation on the addressed sector was completed successfully. The EES command can be used to detect erase operations failed due to loss of power, reset, or failure during the erase operation.

To initiate a EES on a Sector, write 35h to the sector address (SA), while the EAC is in the standby state

The ESS command may not be written while the device is actively programming or erasing or suspended.

The EES command does not allow for reads to the array during the operation. Reads to the array while this command is executing will return unknown data.

Use the Status Register read to confirm if the device is still busy and when complete if the sector is erased or not. Bit 7 of the Status Register will show if the device is performing a ESS (similar to an erase operation). Bit 5 of the Status Register will be cleared to 0 if the sector is erased and set to 1 if not erased.

As soon as any bit is found to not be erased, the device will halt the operation and report the results.

Once the ESS is completed, the EAC will return to the Standby State.

The EES command requires $t_{EES}$ (refer to Table 73 on page 97) to complete and update the erase status in SR. The DRB bit (SR[7]) may be read to determine when the EES command is finished. If a sector is found not erased with SR[5]=1, the sector must be erased again to ensure reliable storage of data in the sector.

## 10.16 Continuity Check

The Continuity Check provides a basic test of connectivity from package connectors to each die pad. This feature is an extension of the legacy unlock cycle sequence used at the beginning of several commands. The unlock sequence is two writes with alternating ones and zeros pattern on the lower portion of the address and data lines with the pattern inverted between the first and second write.

To perform a continuity check these patterns are extended to cover all address and data lines:

| | Address Bus | Data Bus |
|---|---|---|
| x16 Mode | AMAX–A0 | D15–D0 |
| x8 Mode | AMAX–A-1 | D7–D0 |

A logic comparison circuit looks for the alternating one and zero pattern that is inverted between the two write cycles.

When the correct patterns are detected, the status register bit zero is set to 1. The status register clear command will clear the status register bit zero to a 0.



Downloaded from Arrow.com.

**Table 19. x16 Data Bus**

| Phase | Access Type | S29GL064S Address A21 to A0 | Data | Comment |
|---|---|---|---|---|
| Set-Up | Write | 555 | XX71 | Clear die status |
| | Write | 555 | XX70 | Write Status Register Read command to die |
| | Read | XXX | RD | Read status from die to confirm status bit zero = 0 |
| Continuity Pattern | Write | 2AAA55 | FF00 | First continuity cycle |
| | Write | 1555AA | 00FF | Second continuity cycle |
| Verify continuity pattern detected | Write | 555 | XX70 | Write Status Register Read command to die |
| | Read | XXX | RD | Read status from die to confirm status bit zero = 1 for continuity pattern detected |

**Table 20. X8 Data Bus**

| Phase | Access Type | S29GL064S Address A21 to A-1 | Data | Comment |
|---|---|---|---|---|
| Set-Up | Write | AAA | 71 | Clear die status |
| | Write | AAA | 70 | Write Status Register Read command to die |
| | Read | XXX | RD | Read status from die to confirm status bit zero = 0 |
| Continuity Pattern | Write | 5554AB | FF | First continuity cycle |
| | Write | 2AAB54 | 00 | Second continuity cycle |
| Verify continuity pattern detected | Write | AAA | 70 | Write Status Register Read command to die |
| | Read | XXX | RD | Read status from die to confirm status bit zero = 1 for continuity pattern detected |

The alternating one and zero pattern checks for adjacent wire shorts. The inversion of the pattern between cycles checks for stuck-at faults. The status output being cleared and set checks for stuck-at faults on the status output. Checking for different status results from each die checks for working die selection logic.



Downloaded from Arrow.com.

## 10.17 Command Definitions

**Table 21. Command Definitions (x16 Mode, BYTE# = $V_{IH}$)**

| Command Sequence (Note 1) | | Cycles | Bus Cycles (Notes 2–5) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Read (Note 5) | | 1 | RA | RD | | | | | | | | | | | | |
| Reset (Note 6) | | 1 | XXX | F0 | | | | | | | | | | | | |
| Status Register Read | | 2 | 555 | 70 | XXX | RD | | | | | | | | | | |
| Status Register Clear | | 1 | 555 | 71 | | | | | | | | | | | | |
| Autoselect (Note 7) | Manufacturer ID | 4 | 555 | AA | 2AA | 55 | 555 | 90 | X00 | 0001 | | | | | | |
| | Device ID (Note 8) | 6 | 555 | AA | 2AA | 55 | 555 | 90 | X01 | 227E | X0E | (18) | X0F | (18) | | |
| | Device ID | 4 | 555 | AA | 2AA | 55 | 555 | 90 | X01 | (17) | | | | | | |
| | Secure Silicon Region Factory Protect | 4 | 555 | AA | 2AA | 55 | 555 | 90 | X03 | (9) | | | | | | |
| | Sector Protect Verify (Note 10) | 4 | 555 | AA | 2AA | 55 | 555 | 90 | (SA) X02 | 00/01 | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |
| Program | | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | | | |
| Write to Buffer (Note 11) | | 3 | 555 | AA | 2AA | 55 | SA | 25 | SA | WC | PA | PD | WBL | PD | | |
| Program Buffer to Flash | | 1 | SA | 29 | | | | | | | | | | | | |
| Write to Buffer Abort Reset (Note 12) | | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | | | |
| Unlock Bypass | Enter | 3 | 555 | AA | 2AA | 55 | 555 | 20 | | | | | | | | |
| | Program (Note 13) | 2 | XXX | A0 | PA | PD | | | | | | | | | | |
| | Write to Buffer (Note 13) | 4 | SA | 25 | SA | WC | PA | PD | WBL | PD | | | | | | |
| | Sector Erase | 2 | XXX | 80 | SA | 30 | | | | | | | | | | |
| | Chip Erase | 2 | XXX | 80 | XXX | 10 | | | | | | | | | | |
| | Reset (Note 14) | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| Chip Erase | | 6 | 555 | AA | 2AA | 55 | 555 | 80 | 555 | AA | 2AA | 55 | 555 | 10 | | |
| Sector Erase | | 6 | 555 | AA | 2AA | 55 | 555 | 80 | 555 | AA | 2AA | 55 | SA | 30 | | |
| Erase Suspend / Program Suspend Legacy Method (Note 15) | | 1 | XXX | B0 | | | | | | | | | | | | |
| Erase Suspend Enhanced Method | | | | | | | | | | | | | | | | |
| Erase Resume / Program Resume Legacy Method (Note 16) | | 1 | XXX | 30 | | | | | | | | | | | | |
| Erase Resume Enhanced Method | | | | | | | | | | | | | | | | |
| Program Suspend Enhanced Method | | 1 | XXX | 51 | | | | | | | | | | | | |
| Program Resume Enhanced Method | | 1 | XXX | 50 | | | | | | | | | | | | |
| Evaluate Erase Status | | 1 | (SA) 555 | 35 | | | | | | | | | | | | |
| Secure Silicon Region (SSR) ASO | SSR Entry | 3 | 555 | AA | 2AA | 55 | 555 | 88 | | | | | | | | |
| | Read (Note 5) | 1 | RA | RD | | | | | | | | | | | | |
| | Word Program | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | | | |
| | Write to Buffer (Note 11) | 6 | 555 | AA | 2AA | 55 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | |
| | Program Buffer to Flash (confirm) | 1 | SA | 29 | | | | | | | | | | | | |
| | Write-to-Buffer-Abort Reset (Note 12) | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | | | |
| | SSR Exit | 4 | 555 | AA | 2AA | 55 | 555 | 90 | XX | 0 | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |
| CFI Query (Note 17) | | 1 | 55 | 98 | | | | | | | | | | | | |
| CFI Exit | | 1 | XXX | F0 | | | | | | | | | | | | |
| CFI Exit (Alternate) | | 1 | XXX | FF | | | | | | | | | | | | |



Downloaded from Arrow.com.

**Table 21. Command Definitions (x16 Mode, BYTE# = $V_{IH}$)** (Continued)

| Command Sequence (Note 1) | | Cycles | Bus Cycles (Notes 2–5) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Continuity Check | | 7 | 555 | XX71 | 555 | XX70 | XXX | RD | 2AAA 55 (19) | FF00 | 1555A A (20) | 00FF | 555 | XX70 | XXX | RD |
| ECC ASO | ECC ASO Entry | 3 | 555 | AA | 2AA | 55 | 555 | 75 | | | | | | | | |
| | ECC Status Read | 1 | RA | RD | | | | | | | | | | | | |
| | ECC ASO Exit | 2 | XXX | F0 | | | | | | | | | | | | |

***Legend:***

*X = Don't care.*

*RA = Read Address of memory location to be read.*

*RD = Read Data read from location RA during read operation.*

*PA = Program Address. Addresses latch on falling edge of WE# or CE# pulse, whichever happens later.*

*PD = Program Data for location PA. Data latches on rising edge of WE# or CE# pulse, whichever happens first.*

*SA = Sector Address of sector to be verified (in autoselect mode) or erased. Address bits $A_{MAX}$–A15 uniquely select any sector for uniform mode device and $A_{MAX}$–A12 for boot mode device.*

*WBL = Write Buffer Location. Address must be within same write buffer page as PA.*

*WC = Word Count. Number of write buffer locations to load minus 1.*

***Notes:***

1. *See Table 5 on page 14 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Shaded cells indicate read cycles. All others are write cycles.*
4. *During unlock and command cycles, when lower address bits are 555 or 2AA as shown in table, address bits above A11 and data bits above DQ7 are don't care.*
5. *No unlock or command cycles required when device is in read mode.*
6. *Reset command is required to return to read mode (or to erase-suspend-read mode if previously in Erase Suspend) when device is in autoselect mode, or if DQ5 goes high while device is providing status information.*
7. *Fourth cycle of the autoselect command sequence is a read cycle. Data bits DQ15–DQ8 are don't care. Except for RD, PD and WC. See Autoselect Command Sequence on page 31 for more information.*
8. *Device ID must be read in three cycles.*
9. *Refer to Table 10 on page 20 for data indicating Secure Silicon Region factory protect status.*
10. *Data is 00h for an unprotected sector and 01h for a protected sector.*
11. *Total number of cycles in command sequence is determined by number of words written to write buffer. Maximum number of cycles in command sequence is 37, including* Program Buffer to Flash *command.*
12. *Command sequence resets device for next command after aborted write-to-buffer operation.*
13. *Unlock Bypass command is required prior to Unlock Bypass Program command.*
14. *Unlock Bypass Reset command is required to return to read mode when device is in unlock bypass mode.*
15. *System may read and program in non-erasing sectors, or enter autoselect mode, when in Erase Suspend mode. Erase Suspend command is valid only during a sector erase operation.*
16. *Erase Resume command is valid only during Erase Suspend mode.*
17. *Command is valid when device is ready to read array data or when device is in autoselect mode.*
18. *Refer to Table 10 on page 20, for individual Device IDs per device density and model number.*
19. *The Address for the fourth cycle depends on the number of address lines supported by the device. See Table 19 on page 41.*
20. *The Address for the fifth cycle depends on the number of address lines supported by the device. See Table 19 on page 41.*



Downloaded from Arrow.com.

### Table 22. Sector Protection Commands (x16)

| Command Sequence (Notes) | | Cycles | Bus Cycles (Notes 2–4) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Lock Register Bits | Command Set Entry (Note 5) | 3 | 555 | AA | 2AA | 55 | 555 | 40 | | | | | | | | |
| | Program (Note 6) | 2 | XX | A0 | XXX | Data | | | | | | | | | | |
| | Read (Note 6) | 1 | 00 | Data | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |
| Password Protection | Command Set Entry (Note 5) | 3 | 555 | AA | 2AA | 55 | 555 | 60 | | | | | | | | |
| | Program (Note 8) | 2 | XX | A0 | PWAx | PWDx | | | | | | | | | | |
| | Read (Note 9) | 4 | 00 | PWD0 | 01 | PWD1 | 02 | PWD2 | 03 | PWD3 | | | | | | |
| | Unlock (Note 10) | 7 | 00 | 25 | 00 | 03 | 00 | PWD0 | 01 | PWD1 | 02 | PWD2 | 03 | PWD3 | 00 | 29 |
| | Command Set Exit (Note 7) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |
| Non-Volatile Sector Protection (PPB) | Command Set Entry (Note 5) | 3 | 555 | AA | 2AA | 55 | 555 | C0 | | | | | | | | |
| | PPB Program (Note 11) | 2 | XX | A0 | SA | 00 | | | | | | | | | | |
| | All PPB Erase (Notes 11, 12) | 2 | XX | 80 | 00 | 30 | | | | | | | | | | |
| | PPB Status Read | 1 | SA | RD(0) | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |
| Global Volatile Sector Protection Freeze (PPB Lock) | Command Set Entry (Note 5) | 3 | 555 | AA | 2AA | 55 | 555 | 50 | | | | | | | | |
| | PPB Lock Bit Set | 2 | XX | A0 | XX | 00 | | | | | | | | | | |
| | PPB Lock Bit Status Read | 1 | XXX | RD(0) | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |
| Volatile Sector Protection (DYB) | Command Set Entry (Note 5) | 3 | 555 | AA | 2AA | 55 | 555 | E0 | | | | | | | | |
| | DYB Set | 2 | XX | A0 | SA | 00 | | | | | | | | | | |
| | DYB Clear | 2 | XX | A0 | SA | 01 | | | | | | | | | | |
| | DYB Status Read | 1 | SA | RD(0) | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 6) | 1 | XXX | F0 | | | | | | | | | | | | |

***Legend:***

*X = Don't care.*

*RA = Address of the memory location to be read.*

*SA = Sector Address. Any address that falls within a specified sector. See Tables 6–9 for sector address ranges.*

*PWAx = PPB Password address for word0 = 00h, word1 = 01h, word2 = 02h, and word3 = 03h (Sector Address = Word Line = 0).*

*PWDx = Password data word0, word1, word2, and word3.*

*RD(0) = DQ0 protection indicator bit. If protected, DQ0 = 0. If unprotected, DQ0 = 1.*

*Gray vs. White Box = Read vs. Write Operation.*

***Notes:***

1. *All values are in hexadecimal.*
2. *Shaded cells indicate read cycles. All others are write cycles.*
3. *Address and data bits not specified in table, legend, or notes are don't cares (each hex digit implies 4 bits of data).*
4. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*



Downloaded from Arrow.com.

5. *Entry commands are required to enter a specific mode to enable instructions only available within that mode.*
6. *No unlock or command cycles required when bank is reading array data.*
7. *Exit command must be issued to reset the device into read mode; device may otherwise be placed in an unknown state.*
8. *Entire two bus-cycle sequence must be entered for each portion of the password.*
9. *Full address range is required for reading password.*
10. *Password may be unlocked or read in any order. Unlocking requires the full password (all seven cycles).*
11. *ACC must be at $V_{IH}$ when setting PPB or DYB.*
12. *"All PPB Erase" command pre-programs all PPBs before erasure to prevent over-erasure.*

### Table 23. Command Definitions (x8 Mode, BYTE# = $V_{IL}$)

| Command Sequence (Note 1) | | Cycles | Bus Cycles (Notes 2–5) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Read (Note 6) | | 1 | RA | RD | | | | | | | | | | | | |
| Reset (Note 7) | | 1 | XXX | F0 | | | | | | | | | | | | |
| Status Register Read | | 2 | AAA | 70 | XXX (18) | RD | | | | | | | | | | |
| Status Register Clear | | 1 | AAA | 71 | | | | | | | | | | | | |
| Autoselect (Note 8) | Manufacturer ID | 4 | AAA | AA | 555 | 55 | AAA | 90 | X00 | 01 | | | | | | |
| | Device ID (Note 9) | 6 | AAA | AA | 555 | 55 | AAA | 90 | X02 | 7E | X1C | (17) | X1E | (17) | | |
| | Device ID | 4 | AAA | AA | 555 | 55 | AAA | 90 | X02 | (16) | | | | | | |
| | Secure Silicon Region Factory Protect | 4 | AAA | AA | 555 | 55 | AAA | 90 | X06 | (10) | | | | | | |
| | Sector Protect Verify (Note 11) | 4 | AAA | AA | 555 | 55 | AAA | 90 | (SA) X04 | 00/01 | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |
| Program | | 4 | AAA | AA | 555 | 55 | AAA | A0 | PA | PD | | | | | | |
| Write to Buffer (Note 12) | | 3 | AAA | AA | 555 | 55 | SA | 25 | SA | BC | PA | PD | WBL | PD | | |
| Program Buffer to Flash | | 1 | SA | 29 | | | | | | | | | | | | |
| Write to Buffer Abort Reset (Note 13) | | 3 | AAA | AA | 555 | 55 | AAA | F0 | | | | | | | | |
| Chip Erase | | 6 | AAA | AA | 555 | 55 | AAA | 80 | AAA | AA | 555 | 55 | AAA | 10 | | |
| Sector Erase | | 6 | AAA | AA | 555 | 55 | AAA | 80 | AAA | AA | 555 | 55 | SA | 30 | | |
| Unlock Bypass | Enter | 3 | AAA | AA | 555 | 55 | AAA | 20 | | | | | | | | |
| | Program | 2 | XXX | A0 | PA | PD | | | | | | | | | | |
| | Write to Buffer | 4 | SA | 25 | SA | BC | PA | PD | WBL | PD | | | | | | |
| | Sector Erase | 2 | XXX | 80 | SA | 30 | | | | | | | | | | |
| | Chip Erase | 2 | XXX | 80 | XXX | 10 | | | | | | | | | | |
| | Reset | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| Erase Suspend / Program Suspend Legacy Method (Note 15) | | 1 | XXX | B0 | | | | | | | | | | | | |
| Erase Suspend Enhanced Method | | | | | | | | | | | | | | | | |
| Erase Resume / Program Resume Legacy Method (Note 16) | | 1 | XXX | 30 | | | | | | | | | | | | |
| Erase Resume Enhanced Method | | | | | | | | | | | | | | | | |
| Program Suspend Enhanced Method | | 1 | XXX | 51 | | | | | | | | | | | | |
| Program Resume Enhanced Method | | 1 | XXX | 50 | | | | | | | | | | | | |
| Evaluate Erase Status | | 1 | (SA) AAA | 35 | | | | | | | | | | | | |



Downloaded from Arrow.com.

## Table 23. Command Definitions (x8 Mode, BYTE# = $V_{IL}$) (Continued)

| Command Sequence (Note 1) | | Cycles | Bus Cycles (Notes 2–5) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Secure Silicon Region (SSR) ASO | SSR Entry | 3 | AAA | AA | 555 | 55 | AAA | 88 | | | | | | | | |
| | Read (Note 5) | 1 | RA | RD | | | | | | | | | | | | |
| | Word Program | 4 | AAA | AA | 555 | 55 | AAA | A0 | PA | PD | | | | | | |
| | Write to Buffer (Note 12) | 6 | AAA | AA | 555 | 55 | SA | 25 | SA | BC | PA | PD | WBL | PD | | |
| | Program Buffer to Flash (confirm) | 1 | SA | 29 | | | | | | | | | | | | |
| | Write-to-Buffer-Abort Reset (Note 13) | 3 | AAA | AA | 555 | 55 | AAA | F0 | | | | | | | | |
| | SSR Exit | 4 | AAA | AA | 555 | 55 | AAA | 90 | XXX | 00 | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |
| CFI Query (Note 16) | | 1 | AA | 98 | | | | | | | | | | | | |
| CFI Exit | | 1 | XXX | F0 | | | | | | | | | | | | |
| CFI Exit (Alternate) | | 1 | XXX | FF | | | | | | | | | | | | |
| Continuity Check | | 7 | AAA | 71 | AAA | 70 | XXX | RD | 5554AB (19) | FF | 2AAB54(20) | 00 | AAA | 70 | XXX | RD |
| ECC ASO | ECC ASO Entry | 3 | AAA | AA | 555 | 55 | AAA | 75 | | | | | | | | |
| | ECC Status Read | 1 | RA | RD | | | | | | | | | | | | |
| | ECC ASO Exit | 2 | XXX | F0 | | | | | | | | | | | | |

***Legend:***

*X = Don't care.*
*RA = Read Address of memory location to be read.*
*RD = Read Data read from location RA during read operation.*
*PA = Program Address. Addresses latch on falling edge of WE# or CE# pulse, whichever happens later.*
*PD = Program Data for location PA. Data latches on rising edge of WE# or CE# pulse, whichever happens first.*
*SA = Sector Address of sector to be verified (in autoselect mode) or erased. Address bits $A_{MAX}$–A15 uniquely select any sector for uniform mode device and $A_{MAX}$–A12 for boot mode device.*
*WBL = Write Buffer Location. Address must be within same write buffer page as PA.*
*BC = Byte Count. Number of write buffer locations to load minus 1.*

***Notes:***

1. *See Table 5 on page 14 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Shaded cells indicate read cycles. All others are write cycles.*
4. *During unlock and command cycles, when lower address bits are 555 or AAA as shown in table, address bits above A11 are don't care.*
5. *Unless otherwise noted, address bits A21–A11 are don't cares.*
6. *No unlock or command cycles required when device is in read mode.*
7. *Reset command is required to return to read mode (or to erase-suspend-read mode if previously in Erase Suspend) when device is in autoselect mode, or if DQ5 goes high while device is providing status information.*
8. *Fourth cycle of autoselect command sequence is a read cycle. Data bits DQ15–DQ8 are don't care. See Autoselect Command Sequence on page 31 for more information.*
9. *For S29GL064S Device ID must be read in three cycles.*
10. *Refer to Table 10 on page 20, for data indicating Secure Silicon Region factory protect status.*
11. *Data is 00h for an unprotected sector and 01h for a protected sector.*
12. *Total number of cycles in command sequence is determined by number of bytes written to write buffer. Maximum number of cycles in command sequence is 261, including Program Buffer to Flash command.*
13. *Command sequence resets device for next command after aborted write-to-buffer operation.*
14. *System may read and program in non-erasing sectors, or enter autoselect mode, when in Erase Suspend mode. Erase Suspend command is valid only during a sector erase operation.*
15. *Erase Resume command is valid only during Erase Suspend mode.*
16. *Command is valid when device is ready to read array data or when device is in autoselect mode.*
17. *Refer to Table 10 on page 20, for individual Device IDs per device density and model number.*
18. *For x8 mode, status register bits 0-7 are accessed when Address Bit A-1 is 0 and bits 8-15 are accessed when Address Bit A-1 is 1.*
19. *The Address for the fourth cycle depends on the number of address lines supported by the device. See Table 20 on page 41.*
20. *The Address for the fifth cycle depends on the number of address lines supported by the device. See Table 20 on page 41.*



Downloaded from Arrow.com.

### Table 24. Sector Protection Commands (x8)

| Command Sequence (Notes) | | Cycles | Bus Cycles (Notes 2–5) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1st/8th | | 2nd/9th | | 3rd/10th | | 4th/11th | | 5th | | 6th | | 7th | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Lock Register Bits | Command Set Entry (Note 5) | 3 | AAA | AA | 555 | 55 | AAA | 40 | | | | | | | | |
| | Program (Note 6) | 2 | XXX | A0 | XXX | Data | | | | | | | | | | |
| | Read (Note 6) | 1 | 00 | Data | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |
| Password Protection | Command Set Entry (Note 5) | 3 | AAA | AA | 555 | 55 | AAA | 60 | | | | | | | | |
| | Program (Note 8) | 2 | XXX | A0 | PWAx | PWDx | | | | | | | | | | |
| | Read (Note 9) | 8 | 00 | PWD0 | 01 | PWD1 | 02 | PWD2 | 03 | PWD3 | 04 | PWD4 | 05 | PWD5 | 06 | PWD6 |
| | | | 07 | PWD7 | | | | | | | | | | | | |
| | Unlock (Note 10) | 11 | 00 | 25 | 00 | 03 | 00 | PWD0 | 01 | PWD1 | 02 | PWD2 | 03 | PWD3 | 04 | PWD4 |
| | | | 05 | PWD5 | 06 | PWD6 | 07 | PWD7 | 00 | 29 | | | | | | |
| | Command Set Exit (Note 7) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |
| Non-Volatile Sector Protection (PPB) | Command Set Entry (Note 5) | 3 | AAA | AA | 555 | 55 | AAA | C0 | | | | | | | | |
| | PPB Program (Note 11) | 2 | XXX | A0 | SA | 00 | | | | | | | | | | |
| | All PPB Erase (Notes 11, 12) | 2 | XXX | 80 | 00 | 30 | | | | | | | | | | |
| | PPB Status Read | 1 | SA | RD(0) | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |
| Global Volatile Sector Protection Freeze (PPB Lock) | Command Set Entry (Note 5) | 3 | AAA | AA | 555 | 55 | AAA | 50 | | | | | | | | |
| | PPB Lock Bit Set | 2 | XXX | A0 | XXX | 00 | | | | | | | | | | |
| | PPB Lock Bit Status Read | 1 | XXX | RD(0) | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XXX | 90 | XX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |
| Volatile Sector Protection (DYB) | Command Set Entry (Note 5) | 3 | AAA | AA | 555 | 55 | AAA | E0 | | | | | | | | |
| | DYB Set | 2 | XXX | A0 | SA | 00 | | | | | | | | | | |
| | DYB Clear | 2 | XXX | A0 | SA | 01 | | | | | | | | | | |
| | DYB Status Read | 1 | SA | RD(0) | | | | | | | | | | | | |
| | Command Set Exit (Note 7) | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| | Reset / ASO Exit (Note 7) | 1 | XXX | F0 | | | | | | | | | | | | |

***Legend:***
*X = Don't care.*
*RA = Address of the memory location to be read.*
*SA = Sector Address. Any address that falls within a specified sector. See Tables 6–9 for sector address ranges.*
*PWAx = PPB Password address for byte0 = 00h, byte1 = 01h, byte2 = 02h, byte3 = 03h, byte04= 04h, byte5 = 05h, byte6 = 06h, and byte7 = 07h (Sector Address = Word Line = 0).*
*PWDx = Password data byte0, byte1, byte2, byte3, byte4, byte5, byte6, and byte7.*
*RD(0) = DQ0 protection indicator bit. If protected, DQ0 = 0. If unprotected, DQ0 = 1.*
*Gray vs. White Box = Read vs. Write Operation.*



Downloaded from Arrow.com.

***Notes:***

1. *All values are in hexadecimal.*
2. *Shaded cells indicate read cycles. All others are write cycles.*
3. *Address and data bits not specified in table, legend, or notes are don't cares (each hex digit implies 4 bits of data).*
4. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*
5. *Entry commands are required to enter a specific mode to enable instructions only available within that mode.*
6. *No unlock or command cycles required when bank is reading array data.*
7. *Exit command must be issued to reset the device into read mode; device may otherwise be placed in an unknown state.*
8. *Entry two bus-cycle sequence must be entered for each portion of the password.*
9. *Full address range is required for reading password.*
10. *Password may be unlocked or read in any order. Unlocking requires the full password (all seven cycles).*
11. *ACC must be at $V_{IH}$ when setting PPB or DYB.*
12. *"All PPB Erase" command pre-programs all PPBs before erasure to prevent over-erasure.*



Downloaded from Arrow.com.

# 11. Data Integrity

## 11.1 Erase Endurance

**Table 25. Erase Endurance**

| Parameter | Minimum | Unit |
|---|---|---|
| Program/Erase cycles per main Flash array sectors | 100K | PE cycle |
| Program/Erase cycles per PPB array or non-volatile register array (1) | 100K | PE cycle |

***Note:***

1. *Each write command to a non-volatile register causes a PE cycle on the entire non-volatile register array. OTP bits and registers internally reside in a separate array that is not PE cycled.*

## 11.2 Data Retention

**Table 26. Data Retention**

| Parameter | Test Conditions | Minimum Time | Unit |
|---|---|---|---|
| Data Retention Time | 10K Program/Erase Cycles | 20 | Years |
| | 100K Program/Erase Cycles | 2 | Years |

Contact Cypress Sales or FAE representative for additional information on the data integrity. An application note is available at: www.cypress.com/cypressappnotes.



Downloaded from Arrow.com.

# 12. Status Monitoring

There are three methods for monitoring EA status. Previous generations of the S29GL flash family used the methods called Data Polling and Ready/Busy# (RY/BY#) Signal. These methods are still supported by the S29GL-S family. One additional method is reading the Status Register.

## 12.1 Status Register

The status of program and erase operations is provided by a single 16-bit status register. The Status Register Read command is written followed by one read access of the status register information. The contents of the status register is aliased (overlaid) in all locations of the device address space. The overlay is in effect for one read access, specifically the next read access that follows the Status Register Read command. After the one status register access, the Status Register ASO is exited. The CE# or OE# signal must go High following the status register read access for $t_{CEPH}/t_{OEPH}$ time to return to the address space active at the time the Status Register Read command was issued.

The status register contains bits related to the results - success or failure - of the most recently completed Embedded Algorithms (EA):

- Erase Status (bit 5),
- Program Status (bit 4),
- Write Buffer Abort (bit 3),
- Sector Locked Status (bit 1),
- RFU (bit 0).

and, bits related to the current state of any in process EA:

- Device Busy (bit 7),
- Erase Suspended (bit 6),
- Program Suspended (bit 2),

The current state bits indicate whether an EA is in process, suspended, or completed.

The upper 8 bits (bits 15:8) are reserved. These have undefined High or Low value that can change from one status read to another. These bits should be treated as don't care and ignored by any software reading status.

The Clear Status Register Command will clear to 0 the results related bits of the status register but will not affect the current state bits.

Initiation of an embedded operation will first clear the status register bits.



Downloaded from Arrow.com.

**Table 27. Status Register**

| Bit # | 15:8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Description | Reserved | Device Ready Bit | Erase Suspend Status Bit | Erase Status Bit | Program Status Bit | Write Buffer Abort Status Bit | Program Suspend Status Bit | Sector Lock Status Bit | Continuity Check |
| Bit Name | | DRB | ESSB | ESB | PSB | WBASB | PSSB | SLSB | CC |
| Reset Status | X | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Busy Status | Invalid | 0 | Invalid | Invalid | Invalid | Invalid | Invalid | Invalid | Invalid |
| Ready Status | X | 1 | 0=No Erase in Suspension<br>1=Erase in Suspension | 0=Erase successful<br>1=Erase fail | 0=Program successful<br>1=Program fail | 0=Program not aborted<br>1=Program aborted during Write to Buffer command | 0=No Program in suspension<br>1=Program in suspension | 0=Sector not locked during operation<br>1=Sector locked error | 0= Continuity Check Pattern not detected<br>1= Continuity Check Pattern detected |

***Notes:***

1. *Bits 15 thru 8, and 0 are reserved for future use and may display as 0 or 1. These bits should be ignored (masked) when checking status.*
2. *Bit 7 is 1 when there is no Embedded Algorithm in progress in the device.*
3. *Bits 6 thru 1 are valid only if Bit 7 is 1.*
4. *All bits are put in their reset status by cold reset or warm reset.*
5. *Bits 5, 4, 3, and 1 and 0 are cleared to 0 by the Clear Status Register command or Reset command.*
6. *Upon issuing the Erase Suspend Command, the user must continue to read status until DRB becomes 1.*
7. *ESSB is cleared to 0 by the Erase Resume Command.*
8. *ESB reflects success or failure of the most recent erase operation.*
9. *PSB reflects success or failure of the most recent program operation.*
10. *During erase suspend, programming to the suspended sector, will cause program failure and set the Program status bit to 1.*
11. *Upon issuing the Program Suspend Command, the user must continue to read status until DRB becomes 1.*
12. *PSSB is cleared to 0 by the Program Resume Command.*
13. *SLSB indicates that a program or erase operation failed because the sector was locked.*
14. *SLSB reflects the status of the most recent program or erase operation.*

## 12.2 Write Operation Status

The device provides several bits to determine the status of a program or erase operation: DQ2, DQ3, DQ5, DQ6, and DQ7. Table 28 on page 57 and the following subsections describe the function of these bits. DQ7 and DQ6 each offer a method for determining whether a program or erase operation is complete or in progress. The device also provides a hardware-based output signal, RY/BY#, to determine whether an Embedded Program or Erase operation is in progress or is completed.



Downloaded from Arrow.com.

## 12.3 DQ7: Data# Polling

The Data# Polling bit, DQ7, indicates to the host system whether an Embedded Program or Erase algorithm is in progress or completed, or whether the device is in Erase Suspend. Data# Polling is valid after the rising edge of the final WE# pulse in the command sequence.

During the Embedded Program algorithm, the device outputs on DQ7 the complement of the datum programmed to DQ7. This DQ7 status also applies to programming during Erase Suspend. When programming in x8 mode, the DQ7 polling value will be the DATA# of the last byte entered, regardless if the byte is at an even or odd address. When the Embedded Program algorithm is complete, the device outputs the datum programmed to DQ7. The system must provide the program address to read valid status information on DQ7. If a program address falls within a protected sector, Data# Polling on DQ7 is active for approximately $t_{DP}$, then the device returns to the read mode.

During the Embedded Erase algorithm, Data# Polling produces a 0 on DQ7. When the Embedded Erase algorithm is complete, or if the device enters the Erase Suspend mode, Data# Polling produces a 1 on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status information on DQ7.

After an erase command sequence is written, if all sectors selected for erasing are protected, Data# Polling on DQ7 is active for approximately $t_{DP}$, then the device returns to the read mode. If not all selected sectors are protected, the Embedded Erase algorithm erases the unprotected sectors, and ignores the selected sectors that are protected. However, if the system reads DQ7 at an address within a protected sector, the status may not be valid.

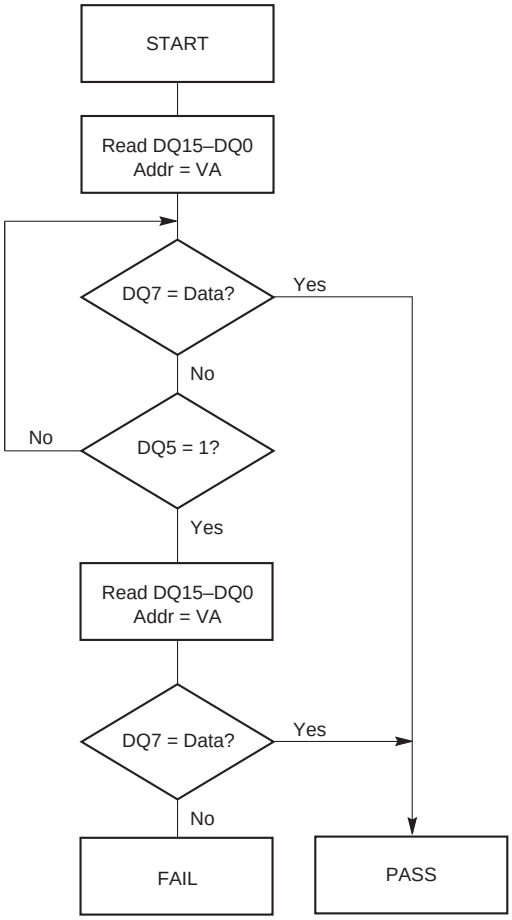
Just prior to the completion of an Embedded Program or Erase operation, DQ7 may change asynchronously with DQ0–DQ6 while Output Enable (OE#) is asserted low. That is, the device may change from providing status information to valid data on DQ7. Depending on when the system samples the DQ7 output, it may read the status or valid data. Even if the device completed the program or erase operation and DQ7 has valid data, the data outputs on DQ0–DQ6 may be still invalid. Valid data on DQ0–DQ7 appears on successive read cycles.

Table 28 on page 57 shows the outputs for Data# Polling on DQ7. Figure 12 on page 53 shows the Data# Polling algorithm. Figure 35 on page 93 shows the Data# Polling timing diagram.



Downloaded from Arrow.com.

**Figure 12. Data# Polling Algorithm**




***Notes:***

1. *VA = Valid address for programming. During a sector erase operation, a valid address is any sector address within the sector being erased. During chip erase, a valid address is any non-protected sector address.*
2. *DQ7 should be rechecked even if DQ5 = 1 because DQ7 may change simultaneously with DQ5.*



Downloaded from Arrow.com.

## 12.4 DQ6: Toggle Bit I

Toggle Bit I on DQ6 indicates whether an Embedded Program or Erase algorithm is in progress or complete, or whether the device entered the Erase Suspend mode. Toggle Bit I may be read at any address, and is valid after the rising edge of the final WE# pulse in the command sequence (prior to the program or erase operation), and during the sector erase time-out.

During an Embedded Program or Erase algorithm operation, successive read cycles to any address cause DQ6 to toggle. The system may use either OE# or CE# to control the read cycles. When the operation is complete, DQ6 stops toggling.

After an erase command sequence is written, if all sectors selected for erasing are protected, DQ6 toggles for approximately $t_{DP}$, then returns to reading array data. If not all selected sectors are protected, the Embedded Erase algorithm erases the unprotected sectors, and ignores the selected sectors that are protected.

The system can use DQ6 and DQ2 together to determine whether a sector is actively erasing or is erase-suspended. When the device is actively erasing (that is, the Embedded Erase algorithm is in progress), DQ6 toggles. When the device enters the Erase Suspend mode, DQ6 stops toggling. However, the system must also use DQ2 to determine which sectors are erasing or erase-suspended. Alternatively, the system can use DQ7 (see DQ7: Data# Polling on page 52).

If a program address falls within a protected sector, DQ6 toggles for approximately $t_{DP}$ after the program command sequence is written, then returns to reading array data.

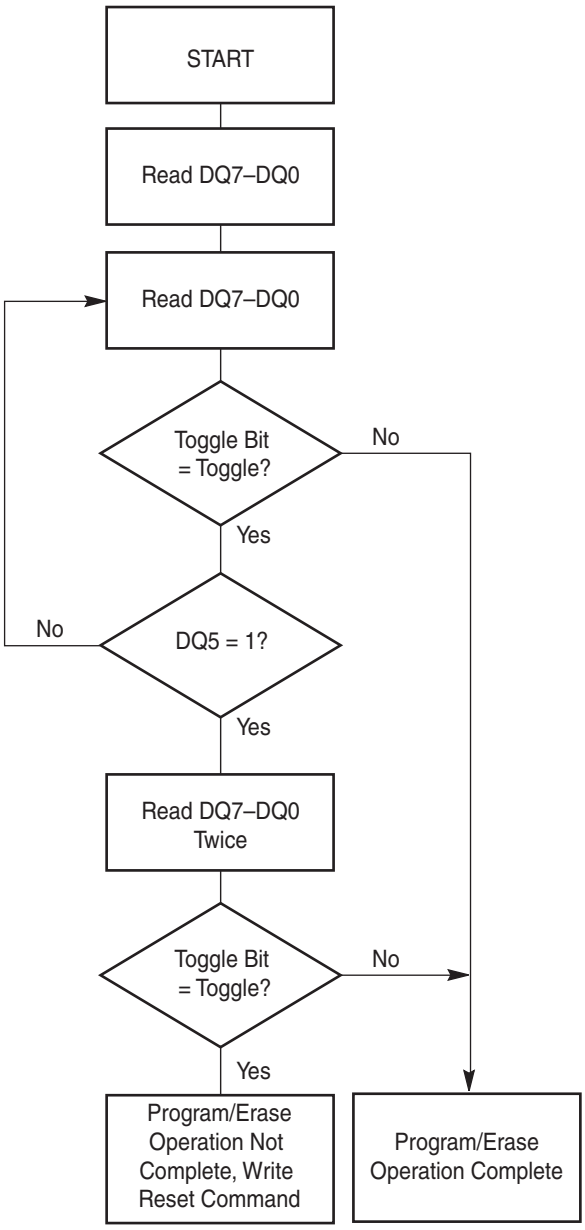
DQ6 also toggles during the erase-suspend-program mode, and stops toggling once the Embedded Program algorithm is complete.

Table 28 on page 57 shows the outputs for Toggle Bit I on DQ6. Figure 13 on page 55 shows the toggle bit algorithm. Figure 36 on page 94 shows the toggle bit timing diagrams. Figure 37 on page 94 shows the differences between DQ2 and DQ6 in graphical form. See also DQ2: Toggle Bit II on page 56.



Downloaded from Arrow.com.

**Figure 13. Toggle Bit Algorithm**




***Note:***
*The system should recheck the toggle bit even if DQ5 = 1 because the toggle bit may stop toggling as DQ5 changes to 1. See Reading Toggle Bits DQ6/DQ2 on page 56 for more information.*



Downloaded from Arrow.com.

## 12.5 DQ2: Toggle Bit II

The “Toggle Bit II” on DQ2, when used with DQ6, indicates whether a particular sector is actively erasing (that is, the Embedded Erase algorithm is in progress), or whether that sector is erase-suspended. (The Toggle Bit II does not apply to the PPB erase command.) Toggle Bit II is valid after the rising edge of the final WE# pulse in the command sequence.

DQ2 toggles when the system reads at addresses within those sectors that were selected for erasure. (The system may use either OE# or CE# to control the read cycles.) But DQ2 cannot distinguish whether the sector is actively erasing or is erase-suspended. DQ6, by comparison, indicates whether the device is actively erasing, or is in Erase Suspend, but cannot distinguish which sectors are selected for erasure. Thus, both status bits are required for sector and mode information. Refer to Table 28 on page 57 to compare outputs for DQ2 and DQ6.

Figure 13 on page 55 shows the toggle bit algorithm in flowchart form. Figure 36 on page 94 shows the toggle bit timing diagram. Figure 37 on page 94 shows the differences between DQ2 and DQ6 in graphical form.

## 12.6 Reading Toggle Bits DQ6/DQ2

Refer to Figure 13 on page 55 for the following discussion. Whenever the system initially begins reading toggle bit status, it must read DQ7–DQ0 at least twice in a row to determine whether a toggle bit is toggling. Typically, the system would note and store the value of the toggle bit after the first read. After the second read, the system would compare the new value of the toggle bit with the first. If the toggle bit is not toggling, the device completed the program or erase operation. The system can read array data on DQ7–DQ0 on the following read cycle.

However, if after the initial two read cycles, the system determines that the toggle bit is still toggling, the system also should note whether the value of DQ5 is high (see DQ5: Exceeded Timing Limits on page 56). If it is, the system should then determine again whether the toggle bit is toggling, since the toggle bit may have stopped toggling just as DQ5 went high. If the toggle bit is no longer toggling, the device successfully completed the program or erase operation. If it is still toggling, the device did not completed the operation successfully, and the system must write the reset command to return to reading array data. It is recommended that data read for polling only be used for polling purposes. Once toggling has stopped array data will be available on subsequent reads.

The remaining scenario is that the system initially determines that the toggle bit is toggling and DQ5 has not gone high. The system may continue to monitor the toggle bit and DQ5 through successive read cycles, determining the status as described in the previous paragraph. Alternatively, it may choose to perform other system tasks. In this case, the system must start at the beginning of the algorithm when it returns to determine the status of the operation (top of Figure 13 on page 55).

## 12.7 DQ5: Exceeded Timing Limits

DQ5 indicates whether the program or erase time exceeded a specified internal pulse count limit. Under these conditions DQ5 produces a 1 indicating that the program or erase cycle was not successfully completed.

In all these cases, the system must write the reset command to return the device to the reading the array (or to erase-suspend-read if the device was previously in the erase-suspend-program mode). In this case, it is possible that the flash will continue to communicate busy for up to $t_{TOR}$ after the reset command is sent.

## 12.8 DQ3: Sector Erase Timer

After writing a sector erase command sequence, the system may read DQ3 to determine whether or not erasure began. (The sector erase timer does not apply to the chip erase command or the PPB erase command.) If additional sectors are selected for erasure, the entire time-out also applies after each additional sector erase command. When the time-out period is complete, DQ3 switches from a 0 to a 1. If the time between additional sector erase commands from the system can be assumed to be less than $t_{SEA}$, the system need not monitor DQ3. See also Sector Erase Command Sequence on page 38.

After the sector erase command is written, the system should read the status of DQ7 (Data# Polling) or DQ6 (Toggle Bit I) to ensure that the device accepted the command sequence, and then read DQ3. If DQ3 is 1, the Embedded Erase algorithm has begun; all further commands (except Erase Suspend) are ignored until the erase operation is complete. If DQ3 is 0, the device accepts additional sector erase commands. To ensure the command is accepted, the system software should check the status of DQ3 prior to and following each subsequent sector erase command. If DQ3 is high on the second status check, the last command might not have been accepted. Table 28 on page 57 shows the status of DQ3 relative to the other status bits.



Downloaded from Arrow.com.

## 12.9 DQ1: Write-to-Buffer Abort

DQ1 indicates whether a Write-to-Buffer operation was aborted. Under these conditions DQ1 produces a 1. The system must issue the Write-to-Buffer-Abort-Reset command sequence to return the device to reading array data. See Write Buffer on page 15 for more details.

**Table 28. Write Operation Status**

| Status | | | DQ7 (Note 2) | DQ6 | DQ5 (Note 1) | DQ3 | DQ2 (Note 2) | DQ1 | RY/BY# |
|---|---|---|---|---|---|---|---|---|---|
| Standard Mode | Embedded Program Algorithm | | DQ7# | Toggle | 0 | N/A | No toggle | 0 | 0 |
| | Embedded Erase Algorithm | | 0 | Toggle | 0 | 1 | Toggle | N/A | 0 |
| Program Suspend Mode | Program-Suspend Read | Program-Suspended Sector | Invalid (not allowed) | | | | | | 1 |
| | | Non-Program Suspended Sector | Data | | | | | | 1 |
| Erase Suspend Mode | Erase-Suspend Read | Erase-Suspended Sector | 1 | No toggle | 0 | N/A | Toggle | N/A | 1 |
| | | Non-Erase Suspended Sector | Data | | | | | | 1 |
| | Erase-Suspend-Program (Embedded Program) (Note 5) | | DQ7# | Toggle | 0 | N/A | N/A | N/A | 0 |
| Write-to-Buffer | Busy (Note 3) | | DQ7# | Toggle | 0 | N/A | N/A | 0 | 0 |
| | Abort (Note 4) | | DQ7# | Toggle | 0 | N/A | N/A | 1 | 0 |

***Notes:***

1. *DQ5 switches to 1 when an Embedded Program, Embedded Erase, or Write-to-Buffer operation exceeded the maximum timing limits. Refer to DQ5: Exceeded Timing Limits on page 56 for more information.*
2. *DQ7 and DQ2 require a valid address when reading status information. Refer to the appropriate subsection for further details.*
3. *The Data# Polling algorithm should be used to monitor the last loaded write-buffer address location.*
4. *DQ1 switches to 1 when the device aborts the write-to-buffer operation.*
5. *DQ6 will not toggle when the sector being polled is a sector selected for sector erase or one of the selected sectors during multi-sector erase.*

## 12.10 RY/BY#: Ready/Busy#

The RY/BY# is a dedicated, open-drain output pin which indicates whether an Embedded Algorithm is in progress or complete. The RY/BY# status is valid after the rising edge of the final WE# pulse in the command sequence. Since RY/BY# is an open-drain output, several RY/BY# pins can be tied together in parallel with a pull-up resistor to $V_{CC}$.

If the output is low (Busy), the device is actively erasing or programming. (This includes programming in the Erase Suspend mode.) If the output is high (Ready), the device is in the read mode, the standby mode, or in the erase-suspend-read mode. Table 28 on page 57 shows the outputs for RY/BY#.

## 12.11 Error Types and Clearing Procedures

There are three types of errors reported by the embedded operation status methods. Depending on the error type, the status reported and procedure for clearing the error status is different. Following is the clearing of error status:

- If an ASO was entered before the error the device remains entered in the ASO awaiting ASO read or a command write.
- If an erase was suspended before the error the device returns to the erase suspended state awaiting flash array read or a command write.
- Otherwise, the device will be in standby state awaiting flash array read or a command write.



Downloaded from Arrow.com.

### 12.11.1 Embedded Operation Error

If an error occurs during an embedded operation (program, erase, evaluate erase status, or password unlock) the device (EAC) remains busy. The RY/BY# output remains Low, data polling status continues to be overlaid on all address locations, and the status register shows ready with valid status bits. The device remains busy until the error status is detected by the host system status monitoring and the error status is cleared.

During embedded algorithm error status the Data Polling status will show the following:

- DQ7 is the inversion of the DQ7 bit in the last word loaded into the write buffer or last word of the password in the case of the password unlock command. DQ7 = 0 for an erase failure
- DQ6 continues to toggle
- DQ5 = 1; Failure of the embedded operation
- DQ4 is RFU and should be treated as don't care (masked)
- DQ3 = 1 to indicate embedded sector erase in progress
- DQ2 continues to toggle, independent of the address used to read status
- DQ1 = 0; Write buffer abort error
- DQ0 is RFU and should be treated as don't care (masked)

During embedded algorithm error status the Status Register will show the following:

- SR[7] = 1; Valid status displayed
- SR[6] = X; May or may not be erase suspended during the EA error
- SR[5] = 1 on erase; else = 0
- SR[4] = 1 on program or password unlock error; else = 0
- SR[3] = 0; Write buffer abort
- SR[2] = 0; Program suspended
- SR[1] = 0; Protected sector
- SR[0] = X; RFU, treat as don't care (masked)

When the embedded algorithm error status is detected, it is necessary to clear the error status in order to return to normal operation, with RY/BY# High, ready for a new read or command write. The error status can be cleared by writing:

- Reset command
- Status Register Clear command

Commands that are accepted during embedded algorithm error status are:

- Status Register Read
- Reset command
- Status Register Clear command



Downloaded from Arrow.com.

### 12.11.2 Protection Error

If an embedded algorithm attempts to change data within a protected area (program, or erase of a protected sector or OTP area) the device (EAC) goes busy for a period of 20 to 100 µs then returns to normal operation. During the busy period the RY/BY# output remains Low, data polling status continues to be overlaid on all address locations, and the status register shows not ready with invalid status bits (SR[7] = 0).

During the protection error status busy period the data polling status will show the following:

- DQ7 is the inversion of the DQ7 bit in the last word loaded into the write buffer. DQ7 = 0 for an erase failure
- DQ6 continues to toggle, independent of the address used to read status
- DQ5 = 0; to indicate no failure of the embedded operation during the busy period
- DQ4 is RFU and should be treated as don't care (masked)
- DQ3 = 1 to indicate embedded sector erase in progress
- DQ2 continues to toggle, independent of the address used to read status
- DQ1 = 0; Write buffer abort error
- DQ0 is RFU and should be treated as don't care (masked)

Commands that are accepted during the protection error status busy period are:

- Status Register Read

When the busy period ends the device returns to normal operation, the data polling status is no longer overlaid, RY/BY# is High, and the status register shows ready with valid status bits. The device is ready for flash array read or write of a new command.

After the protection error status busy period the Status Register will show the following:

- SR[7] = 1; Valid status displayed
- SR[6] = X; May or may not be erase suspended after the protection error busy period
- SR[5] = 1 on erase error, else = 0
- SR[4] = 1 on program error, else = 0
- SR[3] = 0; Program not aborted
- SR[2] = 0; No Program in suspension
- SR[1] = 1; Error due to attempting to change a protected location
- SR[0] = X; RFU, treat as don't care (masked)

Commands that are accepted after the protection error status busy period are:

- Any command



Downloaded from Arrow.com.

### 12.11.3 Write Buffer Abort

If an error occurs during a Write to Buffer command the device (EAC) remains busy. The RY/BY# output remains Low, data polling status continues to be overlaid on all address locations, and the status register shows ready with valid status bits. The device remains busy until the error status is detected by the host system status monitoring and the error status is cleared.

During write to buffer abort (WBA) error status the Data Polling status will show the following:

- DQ7 is the inversion of the DQ7 bit in the last word loaded into the write buffer
- DQ6 continues to toggle, independent of the address used to read status
- DQ5 = 0; to indicate no failure of the programming operation. WBA is an error in the values input by the Write to Buffer command before the programming operation can begin
- DQ4 is RFU and should be treated as don't care (masked)
- DQ3 is don't care after program operation as no erase is in progress. If the Write Buffer Program operation was started after an erase operation had been suspended then DQ3 = 1. If there was no erase operation in progress then DQ3 is a don't care and should be masked.
- DQ2 does not toggle after program operation as no erase is in progress. If the Write Buffer Program operation was started after an erase operation had been suspended then DQ2 will toggle in the sector where the erase operation was suspended and not in any other sector. If there was no erase operation in progress then DQ2 is a don't care and should be masked.
- DQ1 = 1: Write buffer abort error
- DQ0 is RFU and should be treated as don't care (masked)

During embedded algorithm error status the Status Register will show the following:

- SR[7] = 1; Valid status displayed
- SR[6] = X; May or may not be erase suspended during the WBA error status
- SR[5] = 0; Erase successful
- SR[4] = 1; Programming related error
- SR[3] = 1; Write buffer abort
- SR[2] = 0; No Program in suspension
- SR[1] = 0; Sector not locked during operation
- SR[0] = X; RFU, treat as don't care (masked)

When the WBA error status is detected, it is necessary to clear the error status in order to return to normal operation, with RY/BY# High, ready for a new read or command write. The error status can be cleared by writing:

- Write Buffer Abort Reset command
  - Clears the status register and returns to normal operation
- Status Register Clear command

Commands that are accepted during embedded algorithm error status are:

- Status Register Read
  - Reads the status register and returns to WBA busy state
- Write Buffer Abort Reset command
- Status Register Clear command



Downloaded from Arrow.com.

# 13. Command State Transitions

Tables 29 - 55 list the Command State Transitions for the S29GL064S in x16 mode. States highlighted in yellow indicate the state is documented but not recommended.

**Table 29. Read Command State Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Status Register Clear | Unlock 1 | Evaluate Erase Status | CFI Entry | Continuity Entry | Continuity Test |
|---|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h | x555h | (SA)555h | x55h | 2AAA55h | 1555AAh |
| | Data | RD | xF0h | x70h | x71h | xAAh | x35h | x98h | FF00h | 00FFh |
| READ (1) | Read Protect = True | READ | READ | READ | READ | READUL1 | - | CFI (READ) | - | - |
| | Read Protect = False | | | | | | ESS | | CONT | |
| CONT | - | CONT | READ | READ | - | - | - | - | - | READ |

***Note:***
*1. Read Protect = True is defined when LR(5) = 0, LR(2) = 0, and Read Password given does not match the internal password.*

**Table 30. Read Unlock Command State Transition**

| Current State | Command and Condition | Read | Unlock 2 | Word Program Entry | Write to Buffer Enter | Erase Enter | Unlock Bypass Enter | ID (Autoselect) Entry | SSR Entry | Lock Register Entry | Password ASO Entry | PPB ASO Entry | PPB Lock Entry | DYB ASO Entry |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | x2AAh | x555h | (SA)xh | x555h | x555h | x555h | (SA)555h | x555h | x555h | x555h | x555h | x555h |
| | Data | RD | x55h | xA0h | x25h | x80h | x20h | x90h | x88h | x40h | x60h | xC0h | x50h | xE0h |
| READUL1 | - | READUL1 | READUL2 | - | - | - | - | - | - | - | - | - | - | - |
| READUL2 (1) | Read Protect = True | READUL2 | - | - | - | - | - | AS (READ) | - | - | PP | - | - | - |
| | Read Protect = False | | | PG1 | WB | ER | UB | | SSR (READ) | LR | | PPB (READ) | PPBLB | DYB (READ) |

***Note:***
*1. Read Protect = True is defined when LR(5) = 0, LR(2) = 0, and Read Password given does not match the internal password*



Downloaded from Arrow.com.

**Table 31. Erase State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Unlock 1 | Unlock 2 | Chip Erase Start | Sector Erase Start | Erase Suspend Enhanced Method (4) |
|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h | x2AAh | x555h | (SA)xh | xh |
| | Data | RD | xF0h | x70h | xAAh | x55h | x10h | x30h | xB0h |
| ER | - | ER | - | READ | ERUL1 | - | - | - | - |
| ERUL1 | - | ERUL1 | - | READ | - | ERUL2 | - | - | - |
| ERUL2 | - | ERUL2 | - | READ | - | - | CER | SER | - |
| CER (3) | SR(7) = 0 | CER | - | CER | - | - | - | - | - |
| | SR(7) = 1 and DQ5 = 1 (6) | | READ | | | | | | |
| SER (1) (3) | SR(7) = 0 and DQ3 = 0 | SER | - | SER | - | - | - | SER | ESR |
| | SR(7) = 0 and DQ3 = 1 | | | | - | - | | - | |
| | SR(7) = 1 and DQ5 = 1 (6) | | READ | | | | | | - |
| ESR (5) | - | ESR | - | ESR | - | - | - | - | - |
| ESS (3) | SR(7) = 0 | ESS | - | ESS | - | - | - | - | - |
| | SR(7) = 1 and DQ5 = 1 | | READ | | | | | | |

***Notes:***

1. *Issuing a suspend command during the DQ3 = 0 period will force DQR to 1 and queuing of additional sectors will not be allowed after the resume.*
2. *SR Clear will only clear the SR, not the DQ bits.*
3. *State will automatically move to READ state at the successful completion of the operation.*
4. *Also known as Erase Suspend / Program Suspend Legacy Method.*
5. *State will automatically move to ES state by $t_{ESL}$.*
6. *Hang State (time out) only. Sector Protection will have returned to READ state.*

**Table 32. Erase Suspend State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Status Register Clear | Unlock 1 | Erase Resume Enhanced Method (1) |
|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h | x555h | xh |
| | Data | RD | xF0h | x70h | x71h | xAAh | x30h |
| ES | - | ES | ES | ES | ES | ESUL1 | SER |

***Note:***

1. *Also known as Erase Resume / Program Resume Legacy Method.*

**Table 33. Erase Suspend Unlock State Command Transition**

| Current State | Command and Condition | Read | Unlock 2 | Word Program Entry | Write to Buffer Enter |
|---|---|---|---|---|---|
| | Address | RA | x2AAh | x555h | (SA)xh |
| | Data | RD | x55h | xA0h | x25h |
| ESUL1 | - | ESUL1 | ESUL2 | - | - |
| ESUL2 | - | ESUL2 | - | ESPG1 | ES_WB |



Downloaded from Arrow.com.

**Table 34. Erase Suspend - Program Command State Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Program Buffer to Flash (confirm) | Erase Suspend Enhanced Method (1) | Program Suspend Enhanced Method | Write Data |
|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | (SA)xh | xh | xh | xh |
| | Data | RD | xF0h | x70h | x29h | xB0h | x51h | xh |
| ES_WB (6) | WC > 127 or SAe ≠ SAc | ES_WB | - | - | - | - | - | PGE (ES) |
| | WC ≤ 127 and SAe = SAc | | | | | | | ES_WB_D |
| ES_WB_D (6) | WC = -1 and SAe ≠ SAc | ES_WB_D | - | - | - | - | - | PGE (ES) |
| | WC = -1 and SAe = SAc | | | | ESPG | | | |
| | WC ≥ 0 and Write Buffer ≠ Write Buffer | | | | - | | | |
| | WC ≥ 0 and Write Buffer = Write Buffer | | | | | | | ES_WB_D (5) |
| ESPG1 | - | ESPG1 | - | - | - | - | - | ESPG |
| ESPG (2) | SR(7) = 0 | ESPG | - | ESPG | - | ESPSR | ESPSR | - |
| | SR(7) = 1 (3) | | ES | | | - | - | |
| ESPSR (4) | - | ESPSR | - | ESPSR | - | - | - | - |

***Notes:***

1. *Also known as Erase Suspend / Program Suspend Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*
2. *When Program operation is completed with no errors then it will return to Erase Suspend.*
3. *Hang State (time out) only. Sector Protection will have returned to ES state.*
4. *State will automatically move to ESPS state by $t_{PSL}$.*
5. *WC counter will automatically decrement by 1.*
6. *SAe = SAc: SAe is the SA entered with programming command. SAc is the current command.*

**Table 35. Erase Suspend - Program Suspend Command State Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Erase Resume Enhanced Method (1) | Program Resume Enhanced Method |
|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | xh | xh |
| | Data | RD | xF0h | x70h | x30h | x50h |
| ESPS | - | ESPS | ESPS | ESPS | ESPG | ESPG |

***Note:***

1. *Also known as Erase Resume / Program Resume Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*



Downloaded from Arrow.com.

### Table 36. Program State Command Transition

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Program Buffer to Flash (confirm) | Erase Suspend Enhanced Method (2) | Program Suspend Enhanced Method | Write Data |
|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | (SA)xh | xh | xh | xh |
| | Data | RD | xF0h | x70h | x29h | xB0h | x51h | xh |
| WB (6) | WC > 127 or SAe ≠ SAc | WB | - | - | - | - | - | PGE (READ) |
| | WC ≤ 127 and SAe = SAc | | | | | | | WB_D |
| WB_D (6) | WC = -1 and SAe ≠ SAc | WB_D | - | - | - | - | - | PGE (READ) |
| | WC = -1 and SAe = SAc | | | | PG | | | |
| | WC ≥ 0 and Write Buffer ≠ Write Buffer | | | | - | | | |
| | WC ≥ 0 and Write Buffer = Write Buffer | | | | | | | WB_D (5) |
| PG1 | - | PG1 | - | - | - | - | - | PG |
| PG (1) | SR(7) = 0 | PG | - | PG | - | PSR | PSR | - |
| | SR(7) = 1 (3) | | READ | | | - | - | |
| PSR (4) | - | PSR | - | PSR | - | - | - | - |

***Notes:***

1. *State will automatically move to READ state at the completion of the operation.*
2. *Also known as Erase Suspend / Program Suspend Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*
3. *Hang State (time out) only. Sector Protection will have returned to READ state.*
4. *State will automatically move to PS state by $t_{PSL}$.*
5. *WC counter will automatically decrement by 1.*
6. *SAe = SAc: SAe is the SA entered with programming command. SAc is the current command.*

### Table 37. Program Suspend State Command Transition

| Current State | Command and Condition | Read | Status Register Read Enter | Erase Resume Enhanced Method (1) | Program Resume Enhanced Method |
|---|---|---|---|---|---|
| | Address | RA | x555h | xh | xh |
| | Data | RD | x70h | x30h | x50h |
| PS | - | PS | PS | PG | PG |

***Note:***

1. *Also known as Erase Resume / Program Resume Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*



Downloaded from Arrow.com.

**Table 38. Program Abort Command State Transition**

| Current State | Command and Condition | Read | Status Register Read Enter | Unlock 1 | Unlock 2 | Write-To-Buffer Abort Reset |
|---|---|---|---|---|---|---|
| | Address | RA | x555h | x555h | x2AAh | x555h |
| | Data | RD | x70h | xAAh | x55h | xF0h |
| PGE | - | PGE (-) | PGE | PGEUL1 | - | - |
| PGEUL1 | - | PGEUL1 | - | - | PGEUL2 | - |
| PGEUL2 | - | PGEUL2 | - | - | - | (return) |

**Table 39. Lock Register State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Status Register Clear | Command Set Exit Entry | Command Set Exit | PPB Lock Bit Set Entry | Write Data |
|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h | xh | xh | xh | xh |
| | Data | RD | xF0h | x70h | x71h | x90h | x00h | xA0h | xh |
| LR | - | LR | READ | LR | LR | LREXT | - | LRPG1 | - |
| LRPG1 | - | LRPG1 | - | - | - | - | - | - | LRPG |
| LRPG (1) | SR(7) = 0 | LRPG | - | LRPG | - | - | - | - | - |
| | SR(7) = 1 (2) | | LR | | - | | | | |
| LREXT | - | LREXT | READ | - | - | - | READ | - | - |

***Notes:***

1. *State will automatically move to LR state at the completion of the operation.*
2. *Hang state (time out) only.*

**Table 40. CFI State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | CFI Exit |
|---|---|---|---|---|
| | Address | RA | xh | xh |
| | Data | RD | xF0h | xFFh |
| CFI | - | CFI | (return) | (return) |

**Table 41. Autoselect State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit |
|---|---|---|---|
| | Address | RA | xh |
| | Data | RD | xF0h |
| AS | - | AS | (return) |

**Table 42. Secure Silicon Sector State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Unlock 1 |
|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h |
| | Data | RD | xF0h | x70h | xAAh |
| SSR | - | SSR | (return) | SSR | SSRUL1 |



Downloaded from Arrow.com.

**Table 43. Secure Silicon Sector Unlock State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Unlock 2 | Word Program Entry | Write to Buffer Enter | SSR Exit Entry | SSR Exit |
|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x2AAh | x555h | (SA)xh | x555h | xh |
| | Data | RD | xF0h | x55h | xA0h | x25h | x90h | x00h |
| SSRUL1 | - | SSRUL1 | - | SSRUL2 | - | - | - | - |
| SSRUL2 (1) | SR(2) = 0 | SSRUL2 | - | - | SSRPG1 | SSR_WB | SSREXT | - |
| | SR(2) = 1 | | | | - | - | | |
| SSREXT | - | SSREXT | (return) | - | - | - | - | (return) |

***Note:***
1. *SSR's are protected from programming once SSR protect bit is set.*

**Table 44. Secure Silicon Sector Program State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Program Buffer to Flash (confirm) | Write Data |
|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | (SA)xh | xh |
| | Data | RD | xF0h | x70h | x29h | xh |
| SSR_WB (4) | WC > 127 or SAe ≠ SAc | SSR_WB | - | - | - | PGE (SSR) |
| | WC ≤ 127 and SAe = SAc | | | | | SSR_WB_D |
| SSR_WB_D (4) | WC = -1 and SAe ≠ SAc | SSR_WB_D | - | - | - | - |
| | WC = -1 and SAe = SAc | | | | SSRPG | |
| | WC ≥ 0 and Write Buffer ≠ Write Buffer | | | | - | |
| | WC ≥ 0 and Write Buffer = Write Buffer | | | | | SSR_WB_D (3) |
| SSRPG (1) | SR(7) = 0 | SSRPG | - | SSRPG | - | - |
| | SR(7) = 1 (2) | | SSR | | | |
| SSRPG1 | - | SSRPG1 | - | - | - | - |

***Notes:***
1. *When Program operation is completed with no errors then it will return to SSR State.*
2. *Hang State (time out) only. Sector Protection will have returned to SSR state.*
3. *WC counter will automatically decrement by 1.*
4. *SAe = SAc: SAe is the SA entered with programming command. SAc is the current command.*



Downloaded from Arrow.com.

**Table 45. Password Protection Command State Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Status Register Clear | Password ASO Unlock Enter | Password ASO Unlock Start | Command Set Exit Entry | Command Set Exit | Program Entry | Password Word Count | Write Data |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h | 0h | 0h | xh | xh | xh | 0h | xh |
| | Data | RD | xF0h | x70h | x71h | x25h | x29h | x90h | x00h | xA0h | x03h | xh |
| PP (4) | Read Protect = True | PP | READ | PP | PP | PPWB25 | - | PPEXT | - | - | - | PP |
| | Read Protect = False | | | | | | | | | PPPG1 | | |
| PPWB25 | A10:A0 = 0 | PPWB25 | - | - | - | - | - | - | - | - | PPD | PGE (PP) |
| | A10:A0 ≠ 0 | | | | | | | | | | PGE (PP) | |
| PPD (2) (3) | Last Password Loaded and PWD's match | PPD | - | - | - | - | PPV | - | - | - | - | PGE (PP) |
| | Last Password Loaded and PWD's don't match | | | | | | PPH (5) | | | | | |
| | Not Last Password Loaded and Addresses match | | | | | | - | | | | | PPD |
| | Addresses don't match | | | | | | | | | | | PGE (PP) |
| PPV (6) | SR(7) = 0 | PPV | - | PPV | - | - | - | - | - | - | - | - |
| PPH | SR(7) = 1 (7) | PPH | PP | PPH | - | - | - | - | - | - | - | - |
| PPPG1 | - | PPPG1 | - | - | - | - | - | - | - | - | - | PPPG |
| PPPG (1) | SR(7) = 0 (8) | PPPG | - | PPPG | - | - | - | - | - | - | - | - |
| | SR(7) = 1 (7) | | PP | | - | | | | | | | |
| PPEXT | - | PPEXT | READ | PP | - | - | - | - | READ | - | - | - |

***Notes:***

1. *When program operation is completed with no errors then the device will return to PP State.*
2. *In x16 mode, 4 write cycles are required to load password. In x8 mode, 8 write cycles are required to load password.*
3. *On the 1st cycle SA is compared to the SA given during PPWB25. During all other password load cycles SA and WLB are compared to the prior cycle.*
4. *Read Protect = True is defined when LR(5) = 0, LR(2) = 0, and Read Password given does not match the internal password.*



Downloaded from Arrow.com.

5. *If the password data does not match the hidden internal one, device goes into hang state (SR=x90h).*
6. *Well before the completion of $t_{PPB}$ the device will move to the PP State.*
7. *SR(7) will initially be 0. SR(7) will transition to 1 at the completion of $t_{PPB}$.*
8. *If LR(2) = 0 RDY busy will go low for a short time and then the device goes to the PP state and reports it as a security violation.*

### Table 46. Non-Volatile Protection Command State Transition

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Status Register Clear | Command Set Exit Entry | Command Set Exit | Program Entry | PPB Set Start | All PPB Erase Enter | All PPB Erase Start |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | x555h | xh | xh | xh | (SA)xh | Xh | 0h |
| | Data | RD | xF0h | x70h | x71h | x90h | x00h | xA0h | x00h | x80h | x30h |
| PPB | LR(3) = 0 | PPB | (return) | PPB | PPB | PPBEXT | - | PPBPG1 | - | - | - |
| | LR(3) = 1 | | | | | | | | | PPBBER | |
| PPBPG1 | - | PPBPG1 | PPB | - | - | - | PPBPG | - | PPBPG | - | - |
| PPBPG (2) | SR(7) = 0 | PPBPG | - | PPBPG | - | - | - | - | - | - | - |
| | SR(7) = 1 (1) | | PPB | | - | | | | | | |
| PPBBER | - | PPBBER | PPB | PPB | - | - | - | - | - | - | PPBSER |
| PPBSER (2) | SR(7) = 0 | PPBSER | - | PPBSER | - | - | - | - | - | - | - |
| | SR(7) = 1 (1) | | PPB | | - | | | | | | |
| PPBEXT | - | PPBEXT | - | - | - | - | (return) | - | (return) | - | - |

***Notes:***
1. *Hang State (time out) only. Locked PPB's will have returned to PPB state.*
2. *State will automatically move to PPB state at the completion of the operation.*

### Table 47. PPB Lock Bit Command State Transition

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Command Set Exit Entry | Command Set Exit (1) | Program Entry | PPB Set |
|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | xh | xh | xh | xh |
| | Data | RD | xF0h | x70h | x90h | x00h | xA0h | x00h |
| PPBLB | - | PPBLB | READ | PPBLB | PPBLBEXT | - | PPBLBSET | - |
| PPBLBSET | - | PPBLBSET | - | - | - | PPBLB | - | PPBLB |
| PPBLBEXT | - | PPBLBEXT | - | - | - | READ | - | READ |

***Note:***
1. *Matches only valid data for the set command.*

### Table 48. Volatile Sector Protection Command State Transition

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Command Set Exit Entry | Command Set Exit | Program Entry | DYB Set Start | DYB Clear Start |
|---|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | xh | xh | xh | (SA)xh | (SA)xh |
| | Data | RD | xF0h | x70h | x90h | x00h | xA0h | x00h | x01h |
| DYB | - | DYB | (return) | DYB | DYBEXT | - | DYBSET | - | - |
| DYBSET | - | DYBSET | - | - | - | DYB (-) | - | DYB (-) | DYB (-) |
| DYBEXT | - | DYBEXT | - | - | - | (return) | - | (return) | - |



Downloaded from Arrow.com.

**Table 49. Unlock Bypass Command Transition**

| Current State | Command and Condition | Read | Unlock Bypass Word Program Entry | Unlock Bypass Write to Buffer Entry | Unlock Bypass Erase Entry | Unlock Bypass Reset Entry | Unlock Bypass Reset |
|---|---|---|---|---|---|---|---|
| | Address | RA | xh | PA | xh | xh | xh |
| | Data | RD | xA0h | x25h | x80h | x90h | x0h |
| UB | - | UB | UBPG1 | UBWB | UBER | UBRST | - |
| UBRST | - | UBRST | - | - | - | - | READ |

**Table 50. Unlock Bypass Erase State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Chip Erase Start | Sector Erase Start | Erase Suspend Enhanced Method (3) |
|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | xh | (SA)xh | xh |
| | Data | RD | xF0h | x70h | x10h | x30h | xB0h |
| UBER | - | UBER | - | UB | UBCER | UBSER | - |
| UBCER (1) | SR(7) = 0 | UBCER | - | UBCER | - | - | - |
| | SR(7) = 1 (2) | | UB | | | | |
| UBSER (1) | SR(7) = 0 and DQ3 = 0 | UBSER | - | UBSER | - | UBSER | UBESR |
| | SR(7) = 0 and DQ3 =1 | | | | | - | |
| | SR(7) = 1 and DQ5 = 1 (2) | | UB | | | | |
| UBESR (4) | - | UBESR | - | UBESR | - | - | - |

***Notes:***

1. *State will automatically move to UB state at the completion of the operation.*
2. *Hang State (time out) only. Sector Protection will have returned to UB state.*
3. *Also known as Erase Suspend / Program Suspend Legacy Method.*
4. *State will automatically move to UBES state by $t_{ESL}$.*

**Table 51. Unlock Bypass Erase Suspend State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Erase Resume Enhanced Method (1) | Word Program Entry | Write to Buffer Enter |
|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | xh | xh | (SA)xh |
| | Data | RD | xF0h | x70h | x30h | xA0h | x25h |
| UBES | - | UBES | UBES | UBES | UBSER | UBESPG1 | UBES_WB |

***Note:***

1. *Also known as Erase Resume / Program Resume Legacy Method.*



Downloaded from Arrow.com.

**Table 52. Unlock Bypass Erase Suspend - Program Command State Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Program Buffer to Flash (confirm) | Erase Suspend Enhanced Method (1) | Program Suspend Enhanced Method | Write Data |
|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | (SA)xh | xh | xh | xh |
| | Data | RD | xF0h | x70h | x29h | xB0h | x51h | xh |
| UBES_WB (6) | WC > 127 or SAe ≠ SAc | UBES_WB | - | - | - | - | - | PGE (UBES) |
| | WC ≤ 127 and SAe = SAc | | | | | | | UBES_WB_D |
| UBES_WB_D (6) | WC = -1 and SAe ≠ SAc | UBES_WB_D | - | - | - | - | - | PGE (UBES) |
| | WC = -1 and SAe = SAc | | | | UBESPG | | | |
| | WC ≥ 0 and Write Buffer ≠ Write Buffer | | | | - | | | |
| | WC ≥ 0 and Write Buffer = Write Buffer | | | | | | | UBES_WB_D (5) |
| UBESPG1 | - | UBESPG1 | - | - | - | - | - | UBESPG |
| UBESPG (2) | SR(7) = 0 | UBESPG | - | UBESPG | - | UBESPSR | UBESPSR | - |
| | SR(7) = 1 (3) | | UBES | | | - | - | |
| UBESPSR (4) | - | UBESPSR | - | UBESPSR | - | - | - | - |

***Notes:***

1. *Also known as Erase Suspend / Program Suspend Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*
2. *When Program operation is completed with no errors then it will return to Unlock Bypass Erase Suspend.*
3. *Hang State (time out) only. Sector Protection will have returned to UBES state.*
4. *State will automatically move to UBESPS state by $t_{PSL}$.*
5. *WC counter will automatically decrement by 1.*
6. *SAe = SAc: SAe is the SA entered with programming command. SAc is the current command.*

**Table 53. Unlock Bypass Erase Suspend - Program Suspend Command State Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Erase Resume Enhanced Method (1) | Program Resume Enhanced Method |
|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | xh | xh |
| | Data | RD | xF0h | x70h | x30h | x50h |
| UBESPS | - | UBESPS | UBESPS | UBESPS | UBESPG | UBESPG |

***Note:***

1. *Also known as Erase Resume / Program Resume Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*



Downloaded from Arrow.com.

**Table 54. Unlock Bypass Program State Command Transition**

| Current State | Command and Condition | Read | Software Reset / ASO Exit | Status Register Read Enter | Program Buffer to Flash (confirm) | Erase Suspend Enhanced Method (2) | Program Suspend Enhanced Method | Write Data |
|---|---|---|---|---|---|---|---|---|
| | Address | RA | xh | x555h | (SA)xh | xh | xh | xh |
| | Data | RD | xF0h | x70h | x29h | xB0h | x51h | xh |
| UBWB (6) | WC > 127 or SAe ≠ SAc | UBWB | - | - | - | - | - | PGE (UB) |
| | WC ≤ 127 and SAe = SAc | | | | | | | UBWB_D |
| UBWB_D (6) | WC = -1 and SAe ≠ SAc | UBWB_D | - | - | - | - | - | PGE (UB) |
| | WC = -1 and SAe = SAc | | | | UBPG | | | |
| | WC ≥ 0 and Write Buffer ≠ Write Buffer | | | | - | | | |
| | WC ≥ 0 and Write Buffer = Write Buffer | | | | | | | UBWB_D (5) |
| UBPG1 | - | UBPG1 | - | - | - | - | - | UBPG |
| UBPG (1) | SR(7) = 0 | UBPG | - | UBPG | - | UBPSR | UBPSR | - |
| | SR(7) = 1 (3) | | UB | | | - | - | |
| UBPSR (4) | - | UBPSR | - | UBPSR | - | - | - | - |

***Notes:***

1. *State will automatically move to UB state at the completion of the operation.*
2. *Also known as Erase Suspend / Program Suspend Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*
3. *Hang State (time out) only. Sector Protection will have returned to UB state.*
4. *State will automatically move to UBPS state by $t_{PSL}$.*
5. *WC counter will automatically decrement by 1.*
6. *SAe = SAc: SAe is the SA entered with programming command. SAc is the current command.*

**Table 55. Unlock Bypass Program Suspend State Command Transition**

| Current State | Command and Condition | Read | Status Register Read Enter | Erase Resume Enhanced Method (1) | Program Resume Enhanced Method |
|---|---|---|---|---|---|
| | Address | RA | x555h | xh | xh |
| | Data | RD | x70h | x30h | x50h |
| UBPS | - | UBPS | UBPS | UBPG | UBPG |

***Note:***

1. *Also known as Erase Resume / Program Resume Legacy Method. Not recommend due to the potential of nested loop errors. Instead Program Suspend Enhanced Method is recommended.*



Downloaded from Arrow.com.

**Table 56. Next State Table Lookup**

| Current State | Command Transition | Definition |
|---|---|---|
| AS | Table 41 | ID (Autoselect) |
| CER | Table 31 | Chip Erase Start |
| CFI | Table 40 | CFI Entry |
| CONT | Table 29 | Continuity Enter |
| DYB | Table 48 | DYB ASO |
| DYBEXT | Table 48 | DYB ASO - Command Exit |
| DYBSET | Table 48 | DYB ASO - Set |
| ER | Table 31 | Erase Enter |
| ERUL1 | Table 31 | Erase - Unlock Cycle 1 |
| ERUL2 | Table 31 | Erase - Unlock Cycle 2 |
| ES | Table 32 | Erase Suspended |
| ESPG | Table 34 | Erase Suspended - Program |
| ESPG1 | Table 34 | Erase Suspended - Word Program |
| ESPS | Table 35 | Erase Suspended - Program Suspended |
| ESPSR | Table 34 | Erase Suspended - Program Suspend |
| ESS | Table 31 | Evaluate Erase Status |
| ESR | Table 31 | Erase Suspend Request |
| ESUL1 | Table 33 | Erase Suspended - Unlock Cycle 1 |
| ESUL2 | Table 33 | Erase Suspended - Unlock Cycle 2 |
| ES_WB | Table 34 | Erase Suspended - Write to Buffer |
| ES_WB_D | Table 34 | Erase Suspended - Write to Buffer Data |
| LR | Table 39 | Lock Register |
| LREXT | Table 39 | Lock Register - Command Exit |
| LRPG | Table 39 | Lock Register - Program |
| LRPG1 | Table 39 | Lock Register - Program Start |
| PG | Table 36 | Program |
| PG1 | Table 36 | Word Program |
| PGE | Table 38 | Programming Error |
| PGEUL1 | Table 38 | Programming Error - Unlock 1 |
| PGUUL2 | Table 38 | Programming Error - Unlock 2 |
| PP | Table 45 | Password ASO |
| PPB | Table 46 | PPB |
| PPBBER | Table 46 | PPB - Erase |
| PPBEXT | Table 46 | PPB - Command Exit |
| PPBLB | Table 47 | PPB Lock Bit |
| PPBLBEXT | Table 47 | PPB Lock Bit - Command Exit |
| PPBLBSET | Table 47 | PPB Lock Bit - Set |
| PPBPG | Table 46 | PPB - Program |
| PPBPG1 | Table 46 | PPB - Program Request |
| PPBSER | Table 46 | PPB - Erase Start |
| PPD | Table 45 | Password ASO - Data |
| PPEXT | Table 45 | Password ASO - Command Exit |



Downloaded from Arrow.com.

**Table 56. Next State Table Lookup** (Continued)

| Current State | Command Transition | Definition |
|---|---|---|
| PPH | Table 45 | Password ASO - Hang |
| PPPG | Table 45 | Password ASO - Program |
| PPPG1 | Table 45 | Password ASO - Program Request |
| PPV | Table 45 | Password ASO - Valid |
| PPWB25 | Table 45 | Password ASO - Unlock |
| PS | Table 37 | Program Suspended |
| PSR | Table 36 | Program Suspend Request |
| READ | Table 29 | Read Array |
| READUL1 | Table 30 | Read - Unlock Cycle 1 |
| READUL2 | Table 30 | Read - Unlock Cycle 2 |
| SER | Table 31 | Sector Erase Start |
| SSR | Table 42 | Secure Silicon |
| SSREXT | Table 43 | Secure Silicon - Command Exit |
| SSRPG | Table 44 | Secure Silicon - Program |
| SSRPG1 | Table 44 | Secure Silicon - Word Program |
| SSRUL1 | Table 43 | Secure Silicon - Unlock Cycle 1 |
| SSRUL2 | Table 43 | Secure Silicon - Unlock Cycle 2 |
| SSR_WB | Table 44 | Secure Silicon - Write to Buffer |
| SSR_WB_D | Table 44 | Secure Silicon - Write to Buffer - Write Data |
| UB | Table 49 | Unlock Bypass - Enter |
| UBCER | Table 50 | Unlock Bypass - Chip Erase Start |
| UBER | Table 50 | Unlock Bypass - Erase Enter |
| UBES | Table 51 | Unlock Bypass Erase Suspended |
| UBESR | Table 50 | Unlock Bypass Erase Suspend Request |
| UBESPG | Table 52 | Unlock Bypass Erase Suspended - Program |
| UBESPG1 | Table 52 | Unlock Bypass Erase Suspended - Word Program |
| UBESPS | Table 53 | Unlock Bypass Erase Suspended - Program Suspended |
| UBESPSR | Table 52 | Unlock Bypass Erase Suspended - Program Suspend |
| UBES_WB | Table 52 | Unlock Bypass Erase Suspended - Write to Buffer |
| UBES_WB_D | Table 52 | Unlock Bypass Erase Suspended - Write to Buffer Data |
| UBPS | Table 55 | Unlock Bypass Program Suspended |
| UBPSR | Table 54 | Unlock Bypass Program Suspended Request |
| UBRST | Table 49 | Unlock Bypass- Reset |
| UBSER | Table 50 | Unlock Bypass - Sector Erase Start |
| UBWB | Table 54 | Unlock Bypass - Write to Buffer |
| UBWB_D | Table 54 | Unlock Bypass - Write to Buffer Write Data |
| UBPG1 | Table 54 | Unlock Bypass - Word Program |
| UBPG | Table 54 | Unlock Bypass - Program |
| WB | Table 36 | Write to Buffer |
| WB_D | Table 36 | Write to Buffer Write Data |



Downloaded from Arrow.com.

# 14. Electrical Specifications

## 14.1 Absolute Maximum Ratings

| Parameter | | Rating |
|---|---|---|
| Storage Temperature, Plastic Packages | | –65°C to +150°C |
| Ambient Temperature with Power Applied | | –65°C to +125°C |
| Voltage with Respect to Ground | $V_{CC}$ (Note 1) | –0.5V to +4.0V |
| | $V_{IO}$ (Note 1) | –0.5V to +4.0V |
| | A9 and ACC (Note 2) | –0.5V to +12.5V |
| | All other pins (Note 1) | –0.5V to $V_{IO}$+0.5V |
| Output Short Circuit Current (Note 3) | | 200 mA |

***Notes:***

1. *Minimum DC voltage on input or I/Os is –0.5V. During voltage transitions, inputs or I/Os may overshoot $V_{SS}$ to –2.0V for periods of up to 20 ns. See Figure 16. Maximum DC voltage on input or I/Os is $V_{CC}$ + 0.5V. During voltage transitions, input or I/O pins may overshoot to $V_{CC}$ + 2.0V for periods up to 20 ns. See Figure 17.*
2. *Minimum DC input voltage on pins A9 and ACC is –0.5V. During voltage transitions, A9 and ACC may overshoot $V_{SS}$ to –2.0V for periods of up to 20 ns. See Figure 16. Maximum DC input voltage on pin A9 and ACC is +12.5V which may overshoot to +14.0V for periods up to 20 ns.*
3. *No more than one output may be shorted to ground at a time. Duration of the short circuit should not be greater than one second.*
4. *Stresses above those listed under* Absolute Maximum Ratings *may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this data sheet is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.*

## 14.2 Latchup Characteristics

This product complies with JEDEC standard JESD78C latchup testing requirements.

## 14.3 Thermal Resistance

**Table 57. Thermal Resistance**

| Parameter | Description | TS056 | TS048 | LAE064 | LAA064 | Unit |
|---|---|---|---|---|---|---|
| Theta Ja | Thermal resistance (junction to ambient) | 52.5 | 45.8 | 38.0 | 28.5 | °C/W |

## 14.4 Operating Ranges

Operating ranges define those limits between which the functionality of the device is guaranteed.

### 14.4.1 Temperature Ranges

| Parameter | Symbol | Device | Spec | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| Ambient Temperature | $T_A$ | Industrial (I) | –40 | +85 | °C |
| | | Industrial Plus (V) | –40 | +105 | |
| | | Extended (N) | –40 | +125 | |

### 14.4.2 Power Supply Voltages

| | |
|---|---|
| $V_{CC}$ | 2.7V to 3.6V |
| $V_{IO}$ | 1.65V to $V_{CC}$ + 200 mV |



Downloaded from Arrow.com.

### 14.4.3 Power-Up and Power-Down

During power-up or power-down $V_{CC}$ must always be greater than or equal to $V_{IO}$ ($V_{CC} \geq V_{IO}$).

The device ignores all inputs until a time delay of $t_{VCS}$ has elapsed after the moment that $V_{CC}$ and $V_{IO}$ both rise above, and stay above, the minimum $V_{CC}$ and $V_{IO}$ thresholds. During $t_{VCS}$ the device is performing power on reset operations.

During power-down or voltage drops below $V_{CC}$ Lockout maximum ($V_{LKO}$), the $V_{CC}$ and $V_{IO}$ voltages must drop below $V_{CC}$ Reset ($V_{RST}$) minimum for a period of $t_{PD}$ for the part to initialize correctly when $V_{CC}$ and $V_{IO}$ again rise to their operating ranges. See Figure 15 on page 75. If during a voltage drop the $V_{CC}$ stays above $V_{LKO}$ maximum the part will stay initialized and will work correctly when $V_{CC}$ is again above $V_{CC}$ minimum. If the part locks up from improper initialization, a hardware reset can be used to initialize the part correctly.

Normal precautions must be taken for supply decoupling to stabilize the $V_{CC}$ and $V_{IO}$ power supplies. Each device in a system should have the $V_{CC}$ and $V_{IO}$ power supplies decoupled by a suitable capacitor close to the package connections (this capacitor is generally on the order of 0.1 µF). At no time should $V_{IO}$ be greater then 200 mV above $V_{CC}$ ($V_{CC} \geq V_{IO}$ - 200 mV).

**Table 58. Power-Up / Power-Down Voltage and Timing**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $V_{CC}$ | $V_{CC}$ Power Supply | 2.7 | 3.6 | V |
| $V_{LKO}$ | $V_{CC}$ level below which re-initialization is required (Note 1) | | 2.5 | V |
| $V_{RST}$ | $V_{CC}$ and $V_{IO}$ Low voltage needed to ensure initialization will occur (Note 1) | 1.0 | | V |
| $t_{PD}$ | Duration of $V_{CC} \leq V_{RST}$(min) (Note 1) | 15 | | µs |

***Note:***
*1. Not 100% tested.*

**Figure 14. Power-Up**

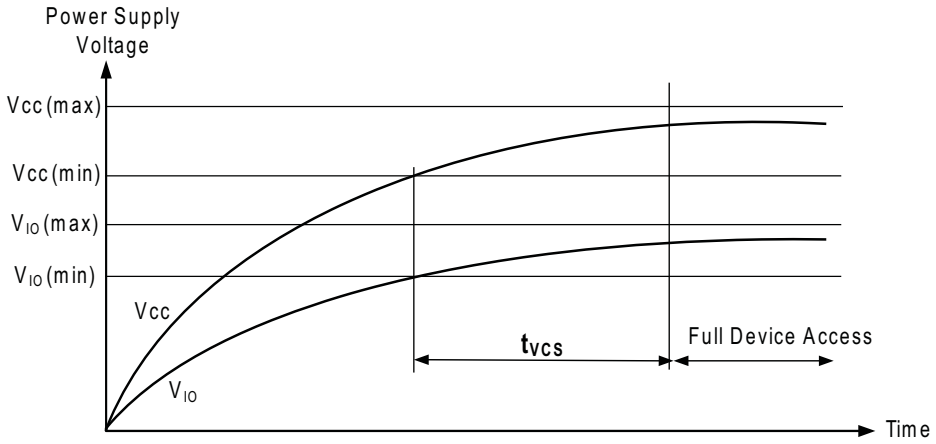
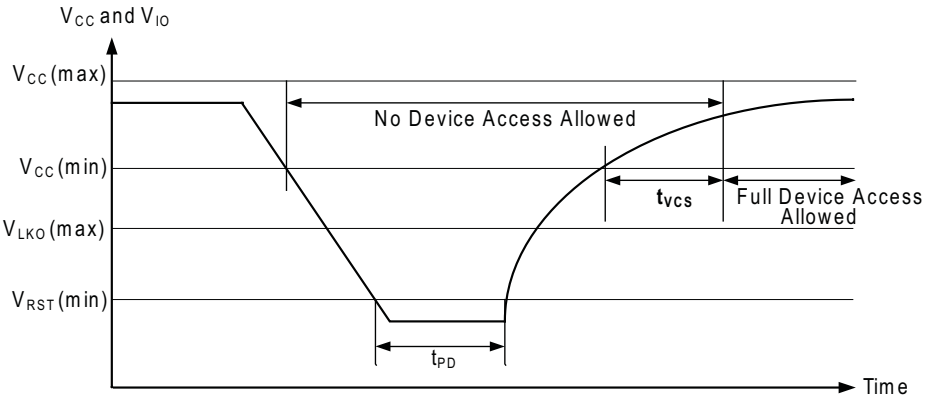



**Figure 15. Power-Down and Voltage Drop**






Downloaded from Arrow.com.

### 14.4.4 Input Signal Overshoot

**Figure 16. Maximum Negative Overshoot Waveform**

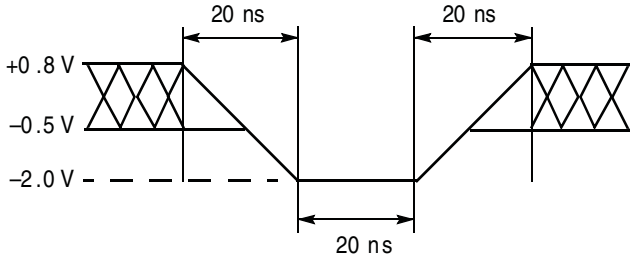
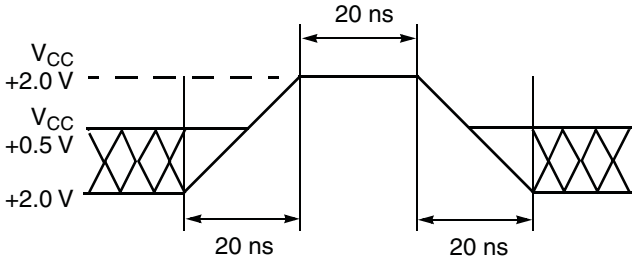



**Figure 17. Maximum Positive Overshoot Waveform**






Downloaded from Arrow.com.

# 15. DC Characteristicst

**Table 59. DC Characteristics**

| Parameter Symbol | Parameter Description (Notes) | Test Conditions | | Min | Typ (Note 1) | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Load Current (Note 2) | $V_{IN} = V_{SS}$ to $V_{IO}$, $V_{CC} = V_{CC\ max}$ | ACC | | | ±2.0 | µA |
| | | | Others | | | ±1.0 | |
| $I_{LIT}$ | A9 Input Load Current | $V_{CC} = V_{CC\ max}$, A9 = 12.5V | | | | 35 | µA |
| $I_{LO}$ | Output Leakage Current | $V_{OUT} = V_{SS}$ to $V_{IO}$, $V_{CC} = V_{CC\ max}$ | | | ±0.02 | ±1.0 | µA |
| $I_{CC1}$ | $V_{CC}$ Initial Read Current (Note 2) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$, Address Switching @ 1 MHz | | | 6.0 | 10 | mA |
| | | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$, Address Switching @ 5 MHz | | | 25 | 30 | |
| | | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$, Address Switching @ 10 MHz | | | 45 | 50 | |
| $I_{IO2}$ | $V_{IO}$ Non-Active Output | CE# = $V_{IL}$, OE# = $V_{IH}$ | | | 0.2 | 10 | mA |
| $I_{CC2}$ | $V_{CC}$ Intra-Page Read Current (Note 2) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$ Address Switching @ 33 MHz | | | 7.5 | 20 | mA |
| $I_{CC3}$ | $V_{CC}$ Active Erase / Program Current (Notes 3, 4) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$ | | | 50 | 60 | mA |
| $I_{CC4}$ | $V_{CC}$ Standby Current | CE#, RESET# = $V_{IH}$, OE# = $V_{IH}$, $V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$ | | | 40 | 100 | µA |
| $I_{CC5}$ | $V_{CC}$ Reset Current (Notes 4, 9) | RESET# = $V_{IH}$,$V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$ | | | 10 | 20 | mA |
| $I_{CC6}$ | Automatic Sleep Mode (Note 5) | ACC = $V_{IH}$, $V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$, $t_{ACC}$ + 30 ns | | | 3 | 6 | mA |
| | | ACC = $V_{IH}$, $V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$, $t_{ASSB}$ | | | 40 | 100 | µA |
| $I_{CC7}$ | $V_{CC}$ Current during power up (Notes 4, 8) | RESET# = $V_{IO}$, CE# = $V_{IO}$, OE# = $V_{IO}$, $V_{CC} = V_{CC}max$ | | | 53 | 80 | mA |
| $I_{ACC}$ | ACC Accelerated Program Current | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC}max$, ACC = $V_{HH}$ | ACC | | 10 | 20 | mA |
| | | | $V_{CC}$ | | 50 | 60 | mA |
| $V_{IL}$ | Input Low Voltage (Note 6) | | | –0.5 | | 0.3 x $V_{IO}$ | V |
| $V_{IH}$ | Input High Voltage (Note 6) | | | 0.7 x $V_{IO}$ | | $V_{IO}$ + 0.4 | V |
| $V_{HH}$ | Voltage for ACC Program Acceleration | $V_{CC}$ = 2.7 –3.6V | | 11.5 | | 12.5 | V |
| $V_{ID}$ | Voltage for Autoselect | $V_{CC}$ = 2.7 –3.6V | | 11.5 | | 12.5 | V |
| $V_{OL}$ | Output Low Voltage (Notes 6, 10) | $I_{OL}$ = 100 µA for DQ15-DQ0<br>$I_{OL}$ = 2 mA for RY/BY# | | | | 0.15 x $V_{IO}$ | V |
| $V_{OH}$ | Output High Voltage (Note 6) | $I_{OH}$ = –100 µA | | 0.85 x $V_{IO}$ | | | V |
| $V_{LKO}$ | Low $V_{CC}$ Lock-Out Voltage (Note 4) | | | 2.3 | | 2.5 | V |
| $V_{RST}$ | Low $V_{CC}$ Power on Reset Voltage (Note 4) | | | | 1.0 | | V |

***Notes:***

1. Temperature = +25°C, $V_{CC}$ = 3V.
2. $I_{CC}$ current listed is typically less than 2 mA / MHz, with OE# at $V_{IH}$.
3. $I_{CC}$ active while Embedded Erase, Embedded Program, or Write Buffer Programming is in progress.
4. Not 100% tested.
5. Automatic sleep mode enables the low power mode when addresses remain stable for $t_{ACC}$ + 30 ns.
6. $V_{IO}$ = 1.65V–$V_{CC}$ or 2.7V–$V_{CC}$.
7. $V_{CC}$ = 3V and $V_{IO}$ = 3V or 1.8V. When $V_{IO}$ is at 1.8V, I/Os cannot operate at 3V.
8. During power-up there are spikes of current demand, the system needs to be able to supply this current to insure the part initializes correctly.



Downloaded from Arrow.com.

9. *If an embedded operation is in progress at the start of reset, the current consumption will remain at the embedded operation specification until the embedded operation is stopped by the reset. If no embedded operation is in progress when reset is started, or following the stopping of an embedded operation, $I_{CC5}$ will be drawn during the remainder of $t_{RPH}$. After the end of $t_{RPH}$ the device will go to standby mode until the next read or write.*
10. *The recommended pull-up resistor for RY/BY# output is 5k to 10k Ohms.*

**Table 60. DC Characteristics, CMOS Compatible In Cabin Temperature (-40°C to +105°C)**

| Parameter Symbol | Parameter Description (Notes) | Test Conditions | | Min | Typ (Note 1) | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Load Current (Note 2) | $V_{IN} = V_{SS}$ to $V_{IO}$, $V_{CC} = V_{CC\ max}$ | ACC | | | ±2.0 | µA |
| | | | Others | | | ±1.0 | |
| $I_{LIT}$ | A9 Input Load Current | $V_{CC} = V_{CC\ max}$, A9 = 12.5V | | | | 35 | µA |
| $I_{LO}$ | Output Leakage Current | $V_{OUT} = V_{SS}$ to $V_{IO}$, $V_{CC} = V_{CC\ max}$ | | | ±0.02 | ±1.0 | µA |
| $I_{CC1}$ | $V_{CC}$ Initial Read Current (Note 2) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$, Address Switching @ 1 MHz | | | 6.0 | 10 | mA |
| | | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$, Address Switching @ 5 MHz | | | 25 | 30 | |
| | | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$, Address Switching @ 10 MHz | | | 45 | 50 | |
| $I_{IO2}$ | $V_{IO}$ Non-Active Output | CE# = $V_{IL}$, OE# = $V_{IH}$ | | | 0.2 | 10 | mA |
| $I_{CC2}$ | $V_{CC}$ Intra-Page Read Current (Note 2) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$ Address Switching @ 33 MHz | | | 7.5 | 20 | mA |
| $I_{CC3}$ | $V_{CC}$ Active Erase / Program Current (Notes 3, 4) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC\ max}$ | | | 50 | 60 | mA |
| $I_{CC4}$ | $V_{CC}$ Standby Current | CE#, RESET# = $V_{IH}$, OE# = $V_{IH}$, $V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$ | | | 40 | <200 | µA |
| $I_{CC5}$ | $V_{CC}$ Reset Current (Notes 4, 9) | RESET# = $V_{IH}$,$V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$ | | | 10 | 20 | mA |
| $I_{CC6}$ | Automatic Sleep Mode (Note 5) | ACC = $V_{IH}$, $V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$, $t_{ACC}$ + 30 ns | | | 3 | 6 | mA |
| | | ACC = $V_{IH}$, $V_{IL} = V_{SS}$, $V_{IH} = V_{IO}$, $V_{CC} = V_{CC\ max}$, $t_{ASSB}$ | | | 40 | <200 | µA |
| $I_{CC7}$ | $V_{CC}$ Current during power up (Notes 4, 8) | RESET# = $V_{IO}$, CE# = $V_{IO}$, OE# = $V_{IO}$, $V_{CC} = V_{CC}$max | | | 53 | 80 | mA |
| $I_{ACC}$ | ACC Accelerated Program Current | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC}$max, ACC = $V_{HH}$ | ACC | | 10 | 20 | mA |
| | | | $V_{CC}$ | | 50 | 60 | mA |
| $V_{IL}$ | Input Low Voltage (Note 6) | | | –0.5 | | 0.3 x $V_{IO}$ | V |
| $V_{IH}$ | Input High Voltage (Note 6) | | | 0.7 x $V_{IO}$ | | $V_{IO}$ + 0.4 | V |
| $V_{HH}$ | Voltage for ACC Program Acceleration | $V_{CC}$ = 2.7 –3.6V | | 11.5 | | 12.5 | V |
| $V_{ID}$ | Voltage for Autoselect | $V_{CC}$ = 2.7 –3.6V | | 11.5 | | 12.5 | V |
| $V_{OL}$ | Output Low Voltage (Notes 7, 10) | $I_{OL}$ = 100 µA for DQ15-DQ0<br>$I_{OL}$ = 2 mA for RY/BY# | | | | 0.15 x $V_{IO}$ | V |
| $V_{OH}$ | Output High Voltage (Note 6) | $I_{OH}$ = –100 µA | | 0.85 x $V_{IO}$ | | | V |
| $V_{LKO}$ | Low $V_{CC}$ Lock-Out Voltage (Note 4) | | | 2.3 | | 2.5 | V |
| $V_{RST}$ | Low $V_{CC}$ Power on Reset Voltage (Note 4) | | | | 1.0 | | V |

***Notes:***
1. Temperature = +25°C, $V_{CC}$ = 3V.
2. $I_{CC}$ current listed is typically less than 2 mA / MHz, with OE# at $V_{IH}$.
3. $I_{CC}$ active while Embedded Erase, Embedded Program, or Write Buffer Programming is in progress.



Downloaded from Arrow.com.

4. *Not 100% tested.*
5. *Automatic sleep mode enables the low power mode when addresses remain stable for $t_{ACC}$ + 30 ns.*
6. *$V_{IO}$ = 1.65 –$V_{CC}$ or 2.7 –$V_{CC}$.*
7. *$V_{CC}$ = 3V and $V_{IO}$ = 3V or 1.8V. When $V_{IO}$ is at 1.8V, I/Os cannot operate at 3V.*
8. *During power-up there are spikes of current demand, the system needs to be able to supply this current to insure the part initializes correctly.*
9. *If an embedded operation is in progress at the start of reset, the current consumption will remain at the embedded operation specification until the embedded operation is stopped by the reset. If no embedded operation is in progress when reset is started, or following the stopping of an embedded operation, $I_{CC5}$ will be drawn during the remainder of $t_{RPH}$. After the end of $t_{RPH}$ the device will go to standby mode until the next read or write.*
10. *The recommended pull-up resistor for RY/BY# output is 5k to 10k Ohms.*

## 15.1 Capacitance Characteristics

### Table 61. Connector Capacitance for FBGA (LAA) Package

| Parameter Symbol | Parameter Description | Test Setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0 | 4 | 7 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0 | 4 | 7 | pF |
| $C_{IN2}$ | Control Pin Capacitance | $V_{IN}$ = 0 | 6 | 8 | pF |
| $C_{IN3}$ | ACC or WP#/ACC Pin Capacitance | $V_{IN}$ = 0 | 4 | 8 | pF |
| RY/BY# | Output Capacitance | $V_{OUT}$ = 0 | 3 | 5 | pF |

***Notes:***
1. *Sampled, not 100% tested.*
2. *Test conditions $T_A$ = 25°C, f = 1.0 MHz.*

### Table 62. Connector Capacitance for FBGA (LAE) Package

| Parameter Symbol | Parameter Description | Test Setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0 | 4 | 6 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0 | 4 | 6 | pF |
| $C_{IN2}$ | Control Pin Capacitance | $V_{IN}$ = 0 | 6 | 7 | pF |
| $C_{IN3}$ | ACC or WP#/ACC Pin Capacitance | $V_{IN}$ = 0 | 4 | 8 | pF |
| RY/BY# | Output Capacitance | $V_{OUT}$ = 0 | 3 | 4 | pF |

***Notes:***
1. *Sampled, not 100% tested.*
2. *Test conditions $T_A$ = 25°C, f = 1.0 MHz.*

### Table 63. Connector Capacitance for FBGA (VBK) Package

| Parameter Symbol | Parameter Description | Test Setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0 | 4 | 6 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0 | 4 | 6 | pF |
| $C_{IN2}$ | Control Pin Capacitance | $V_{IN}$ = 0 | 6 | 7 | pF |
| $C_{IN3}$ | ACC or WP#/ACC Pin Capacitance | $V_{IN}$ = 0 | 4 | 8 | pF |
| RY/BY# | Output Capacitance | $V_{OUT}$ = 0 | 3 | 4 | pF |

***Notes:***
1. *Sampled, not 100% tested.*
2. *Test conditions $T_A$ = 25°C, f = 1.0 MHz.*



Downloaded from Arrow.com.

**Table 64. Connector Capacitance for 56-Pin TSOP and 48-Pin Packages**

| Parameter Symbol | Parameter Description | Test Setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0 | 4 | 7 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0 | 4 | 7 | pF |
| $C_{IN2}$ | Control Pin Capacitance | $V_{IN}$ = 0 | 6 | 8 | pF |
| $C_{IN3}$ | ACC or WP#/ACC Pin Capacitance | $V_{IN}$ = 0 | 4 | 8 | pF |
| RY/BY# | Output Capacitance | $V_{OUT}$ = 0 | 3 | 5 | pF |

***Notes:***

1. *Sampled, not 100% tested.*
2. *Test conditions $T_A$ = 25°C, f = 1.0 MHz.*



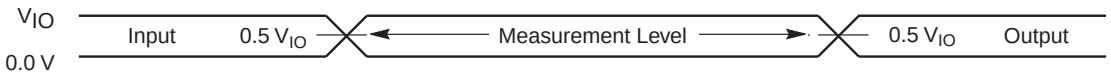
Downloaded from Arrow.com.

# 16. Test Specifications

## 16.1 Key to Switching Waveforms

| Waveform | Inputs | Outputs |
|---|---|---|
| | Steady | |
| | Changing from H to L | |
| | Changing from L to H | |
| | Don't Care, Any Change Permitted | Changing, State Unknown |
| | Does Not Apply | Center Line is High Impedance State (High-Z) |

**Figure 18. Input Waveforms and Measurement Levels**




## 16.2 AC Test Conditions

**Figure 19. Test Setup**

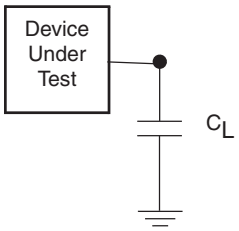



**Table 65. Test Specifications**

| Test Condition | All Speeds | Unit |
|---|---|---|
| Output Load Capacitance, $C_L$ (including jig capacitance) | 30 | pF |
| Input Rise and Fall Times (Note 1) | 1.5 | ns |
| Input Pulse Levels | 0.0 or $V_{IO}$ | V |
| Input timing measurement reference levels | 0.5 $V_{IO}$ | V |
| Output timing measurement reference levels | 0.5 $V_{IO}$ | V |

***Note:***
*1. Measured between $V_{IL}$ max and $V_{IH}$ min.*



Downloaded from Arrow.com.

## 16.3 Power-On Reset (POR) and Warm Reset

Normal precautions must be taken for supply decoupling to stabilize the $V_{CC}$ and $V_{IO}$ power supplies. Each device in a system should have the $V_{CC}$ and $V_{IO}$ power supplies decoupled by a suitable capacitor close to the package connections (this capacitor is generally on the order of 0.1 µF).

**Table 66. Power-On and Reset Parameters**

| Parameter | Description | Limit | Value | Unit |
|---|---|---|---|---|
| $t_{VCS}$ | $V_{CC}$ Setup Time to first access (Notes 1, 2) | Min | 50 | µs |
| $t_{VIOS}$ | $V_{IO}$ Setup Time to first access (Notes 1, 2) | Min | 50 | µs |
| $t_{RPH}$ | RESET# Low to CE# Low | Min | 50 | µs |
| $t_{RP}$ | RESET# Pulse Width | Min | 200 | ns |
| $t_{RH}$ | Time between RESET# (High) and CE# (low) | Min | 50 | ns |
| $t_{RB}$ | RY/BY# output High to CE#, OE# pin Low or Address transition | Min | 0 | ns |
| $t_{CEH}$ | CE# Pulse Width High | Min | 20 | ns |

***Notes:***

1. *Not 100% tested.*
2. *Timing measured from $V_{CC}$ reaching $V_{CC}$ minimum and $V_{IO}$ reaching $V_{IO}$ minimum to $V_{IH}$ on Reset and $V_{IL}$ on CE#.*
3. *RESET# Low is optional during POR. If RESET is asserted during POR, the later of $t_{RPH}$, $t_{VIOS}$, or $t_{VCS}$ will determine when CE# may go Low. If RESET# remains Low after $t_{VIOS}$, or $t_{VCS}$ is satisfied, $t_{RPH}$ is measured from the end of $t_{VIOS}$, or $t_{VCS}$. RESET must also be High $t_{RH}$ before CE# goes Low.*
4. *$V_{CC} \geq V_{IO}$ - 200 mV during power-up.*
5. *$V_{CC}$ and $V_{IO}$ ramp rate can be non-linear.*
6. *Sum of $t_{RP}$ and $t_{RH}$ must be equal to or greater than $t_{RPH}$.*

### 16.3.1 Power-On (Cold) Reset (POR)

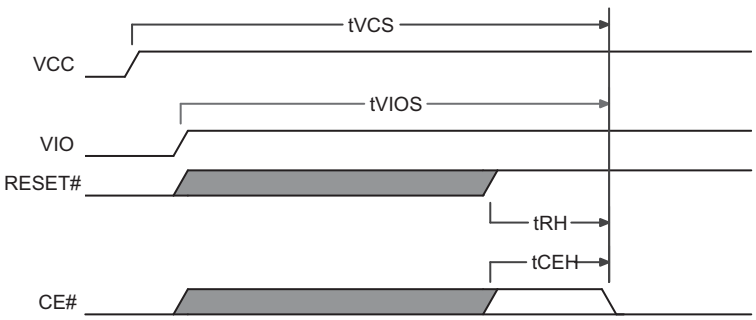
During the rise of power supplies the $V_{IO}$ supply voltage must remain less than or equal to the $V_{CC}$ supply voltage. $V_{IH}$ also must remain less than or equal to the $V_{IO}$ supply.

The Cold Reset Embedded Algorithm requires a relatively long, hundreds of µs, period ($t_{VCS}$) to load all of the EAC algorithms and default state from non-volatile memory. During the Cold Reset period all control signals including CE# and RESET# are ignored. If CE# is Low during $t_{VCS}$ the device may draw higher than normal POR current during $t_{VCS}$ but the level of CE# will not affect the Cold Reset EA. CE# or OE# must transition from High to Low or there must be an address transition after $t_{VCS}$ for a valid read operation ($t_{ACC}$ or $t_{CE}$ is required after $t_{RH}$). RESET# may be High or Low during $t_{VCS}$. If RESET# is Low during $t_{VCS}$ it may remain Low at the end of $t_{VCS}$ to hold the device in the Hardware Reset state. If RESET# is High at the end of $t_{VCS}$ the device will go to the Standby state.

When power is first applied, with supply voltage below $V_{RST}$ then rising to reach operating range minimum, internal device configuration and warm reset activities are initiated. CE# is ignored for the duration of the POR operation ($t_{VCS}$ or $t_{VIOS}$). RESET# Low during this POR period is optional. If RESET# is driven Low during POR it must satisfy the Hardware Reset parameters $t_{RP}$ and $t_{RPH}$. In which case the Reset operations will be completed at the later of $t_{VCS}$ or $t_{VIOS}$ or $t_{RPH}$.

During Cold Reset the device will draw $I_{CC7}$ current.

**Figure 20. Power-Up Diagram**






Downloaded from Arrow.com.

### 16.3.2 Hardware (Warm) Reset

During Hardware Reset ($t_{RPH}$) the device will draw $I_{CC5}$ current.

When RESET# continues to be held at $V_{SS}$, the device draws CMOS standby current ($I_{CC4}$). If RESET# is held at $V_{IL}$, but not at $V_{SS}$, the standby current is greater.

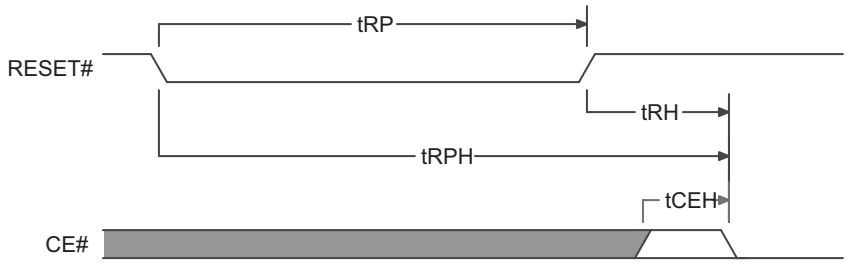
If a Cold Reset has not been completed by the device when RESET# is asserted Low after $t_{VCS}$, the Cold Reset# EA will be performed instead of the Warm RESET#, requiring $t_{VCS}$ time to complete. See Figure 21. Hardware Reset on page 83.

After the device has completed POR and entered the Standby state, any later transition to the Hardware Reset state will initiate the Warm Reset Embedded Algorithm. A Warm Reset is much shorter than a Cold Reset, taking tens of µs ($t_{RPH}$) to complete. During the Warm Reset EA, any in progress Embedded Algorithm is stopped and the EAC is returned to its POR state without reloading EAC algorithms from non-volatile memory. After the Warm Reset EA completes, the interface will remain in the Hardware Reset state if RESET# remains Low. When RESET# returns High the interface will transit to the Standby state. If RESET# is High at the end of the Warm Reset EA, the interface will directly transit to the Standby state. CE# or OE# must transition from High to Low or there must be an address transition after $t_{VCS}$ for a valid read operation ($t_{ACC}$ or $t_{CE}$ is required).

If POR has not been properly completed by the end of $t_{VCS}$, a later transition to the Hardware Reset state will cause a transition to the Power-on Reset interface state and initiate the Cold Reset Embedded Algorithm. This ensures the device can complete a Cold Reset even if some aspect of the system Power-On voltage ramp-up causes the POR to not initiate or complete correctly. The RY/BY# pin is Low during cold or warm reset as an indication that the device is busy performing reset operations.

Hardware Reset is initiated by the RESET# signal going to $V_{IL}$.

**Figure 21. Hardware Reset**






Downloaded from Arrow.com.

# 17. AC Characteristics

## 17.1 Read-Only Operations

**Table 67. Read-Only Operations Industrial Temperature (-40°C to +85°C)**

| Parameter | | Description | | Test Setup | | Speed Options | | Unit |
|---|---|---|---|---|---|---|---|---|
| JEDEC | Std. | | | | | 70 | 80 | |
| $t_{AVAV}$ | $t_{RC}$ | Read Cycle Time (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Min | 70 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 80 | |
| $t_{AVQV}$ | $t_{ACC}$ | Address to Output Delay | CE#, OE# = $V_{IL}$ | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 70 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 80 | |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to Output Delay | OE# = $V_{IL}$ | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 70 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 80 | |
| | $t_{PACC}$ | Page Access Time | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 25 | | |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to Output Delay | Read | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 25 | | |
| | | | Poll | $V_{IO}$ = 2.7V to $V_{CC}$ | | 25 | | |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 35 | | |
| | $t_{ASO}$ | Address Setup Time | Poll | | Min | 15 | | ns |
| | $t_{AHT}$ | Address Hold Time | Poll | | Min | 0 | | ns |
| | $t_{CEPH}$ | CE# High | Poll | | Min | 20 | | ns |
| | $t_{OEP}$ | OE# Low | Poll | | Min | 25 | | ns |
| | $t_{OEPH}$ | OE# High | Poll | | Min | 20 | | ns |
| | $t_{OEC}$ | OE# Cycle Time | Poll | | Min | 60 | | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable to Output High-Z (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 20 | | ns |
| $t_{GHQZ}$ | $t_{DF}$ | Output Enable to Output High-Z (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 20 | | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold Time From Addresses, CE# or OE#, Whichever Occurs First | | | Min | 0 | | ns |
| | $t_{OEH}$ | Output Enable Hold Time (Note 1) | Read | | Min | 0 | | ns |
| | | | Toggle and Data# Polling | | Min | 10 | | ns |
| | $t_{ASSB}$ | Automatic Sleep to Standby time (Note 2) | | CE# = $V_{IL}$, Address stable | Typ | 5 | | µs |
| | | | | | Max | 8 | | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# Low to CE# | | | Max | 10 | | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# High to CE# | | | Max | 10 | | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# Low to Output High-Z (Note 1) | | | Max | 10 | | ns |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# High to Output Delay | | | Max | 10 | | ns |

***Notes:***

1. *Not 100% tested.*
2. *See Figure 19 on page 81 and Table 65 on page 81 for test specifications.*



Downloaded from Arrow.com.

**Table 68. Read-Only Operations Industrial Plus Temperature (-40°C to +105°C)**

| Parameter | | Description | | Test Setup | | Speed Options | | Unit |
|---|---|---|---|---|---|---|---|---|
| JEDEC | Std. | | | | | 80 | 90 | |
| $t_{AVAV}$ | $t_{RC}$ | Read Cycle Time (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Min | 80 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 90 | |
| $t_{AVQV}$ | $t_{ACC}$ | Address to Output Delay | CE#, OE# = $V_{IL}$ | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 80 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 90 | |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to Output Delay | OE# = $V_{IL}$ | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 80 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 90 | |
| | $t_{PACC}$ | Page Access Time | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 25 | | |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to Output Delay | Read | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 25 | | |
| | | | Poll | $V_{IO}$ = 2.7V to $V_{CC}$ | | 25 | | |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 35 | | |
| | $t_{ASO}$ | Address Setup Time | Poll | | Min | 15 | | ns |
| | $t_{AHT}$ | Address Hold Time | Poll | | Min | 0 | | ns |
| | $t_{CEPH}$ | CE# High | Poll | | Min | 20 | | ns |
| | $t_{OEP}$ | OE# Low | Poll | | Min | 25 | | ns |
| | $t_{OEPH}$ | OE# High | Poll | | Min | 20 | | ns |
| | $t_{OEC}$ | OE# Cycle Time | Poll | | Min | 60 | | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable to Output High-Z (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 20 | | ns |
| $t_{GHQZ}$ | $t_{DF}$ | Output Enable to Output High-Z (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 20 | | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold Time From Addresses, CE# or OE#, Whichever Occurs First | | | Min | 0 | | ns |
| | $t_{OEH}$ | Output Enable Hold Time (Note 1) | Read | | Min | 0 | | ns |
| | | | Toggle and Data# Polling | | Min | 10 | | ns |
| | $t_{ASSB}$ | Automatic Sleep to Standby time (Note 1) | | CE# = $V_{IL}$, Address stable | Typ | 5 | | µs |
| | | | | | Max | 8 | | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# Low to CE# | | | Max | 10 | | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# High to CE# | | | Max | 10 | | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# Low to Output High-Z (Note 1) | | | Max | 10 | | ns |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# High to Output Delay | | | Max | 10 | | ns |

***Notes:***

1. *Not 100% tested.*
2. *See Figure 19 on page 81 and Table 65 on page 81 for test specifications.*



Downloaded from Arrow.com.

**Table 69. Read-Only Operations Extended Temperature (-40°C to +125°C)**

| Parameter | | Description | | Test Setup | | Speed Options | | Unit |
|---|---|---|---|---|---|---|---|---|
| JEDEC | Std. | | | | | 80 | 90 | |
| $t_{AVAV}$ | $t_{RC}$ | Read Cycle Time (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Min | 80 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 90 | |
| $t_{AVQV}$ | $t_{ACC}$ | Address to Output Delay | CE#, OE# = $V_{IL}$ | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 80 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 90 | |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to Output Delay | OE# = $V_{IL}$ | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 80 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | | 90 | |
| | $t_{PACC}$ | Page Access Time | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 25 | | |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to Output Delay | Read | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 25 | | |
| | | | Poll | $V_{IO}$ = 2.7V to $V_{CC}$ | | 25 | | |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 35 | | |
| | $t_{ASO}$ | Address Setup Time | Poll | | Min | 15 | | ns |
| | $t_{AHT}$ | Address Hold Time | Poll | | Min | 0 | | ns |
| | $t_{CEPH}$ | CE# High | Poll | | Min | 20 | | ns |
| | $t_{OEP}$ | OE# Low | Poll | | Min | 25 | | ns |
| | $t_{OEPH}$ | OE# High | Poll | | Min | 20 | | ns |
| | $t_{OEC}$ | OE# Cycle Time | Poll | | Min | 60 | | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable to Output High-Z (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 20 | | ns |
| $t_{GHQZ}$ | $t_{DF}$ | Output Enable to Output High-Z (Note 1) | | $V_{IO}$ = 2.7V to $V_{CC}$ | Max | 15 | | ns |
| | | | | $V_{IO}$ = 1.65V to $V_{CC}$ | | 20 | | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold Time From Addresses, CE# or OE#, Whichever Occurs First | | | Min | 0 | | ns |
| | $t_{OEH}$ | Output Enable Hold Time (Note 1) | Read | | Min | 0 | | ns |
| | | | Toggle and Data# Polling | | Min | 10 | | ns |
| | $t_{ASSB}$ | Automatic Sleep to Standby time (Note 1) | | CE# = $V_{IL}$, Address stable | Typ | 5 | | µs |
| | | | | | Max | 8 | | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# Low to CE# | | | Max | 10 | | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# High to CE# | | | Max | 10 | | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# Low to Output High-Z (Note 1) | | | Max | 10 | | ns |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# High to Output Delay | | | Max | 10 | | ns |

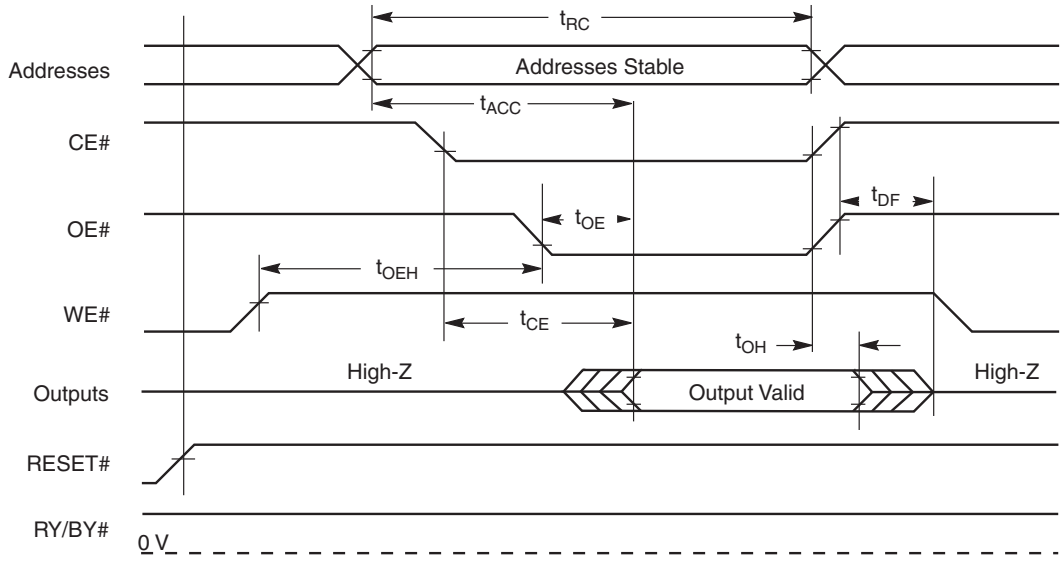
***Notes:***

1. *Not 100% tested.*
2. *See Figure 19 on page 81 and Table 65 on page 81 for test specifications.*



Downloaded from Arrow.com.

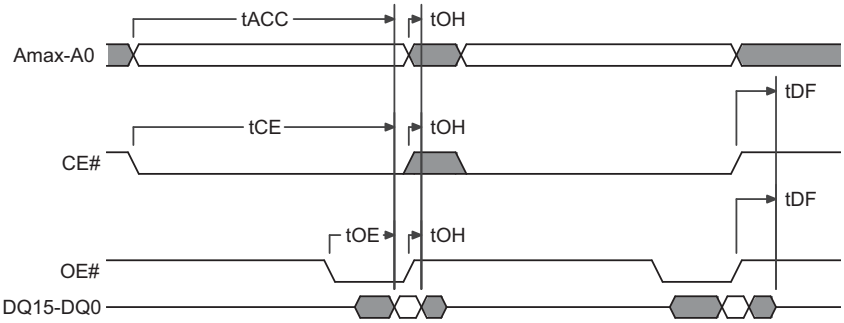
**Figure 22. Read Operation Timings**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

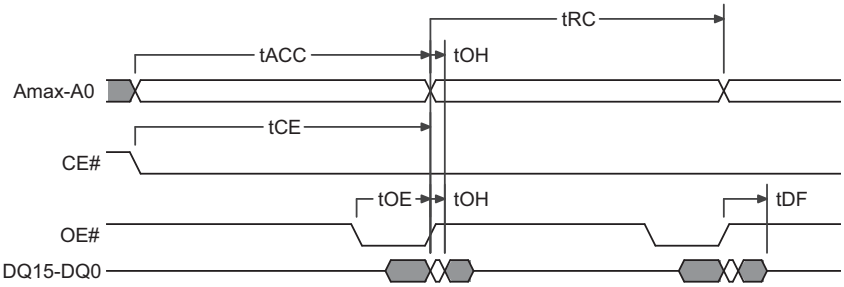
**Figure 23. Back to Back Read ($t_{ACC}$) Operation Timing Diagram**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

**Figure 24. Back to Back Read Operation ($t_{RC}$)Timing Diagram**



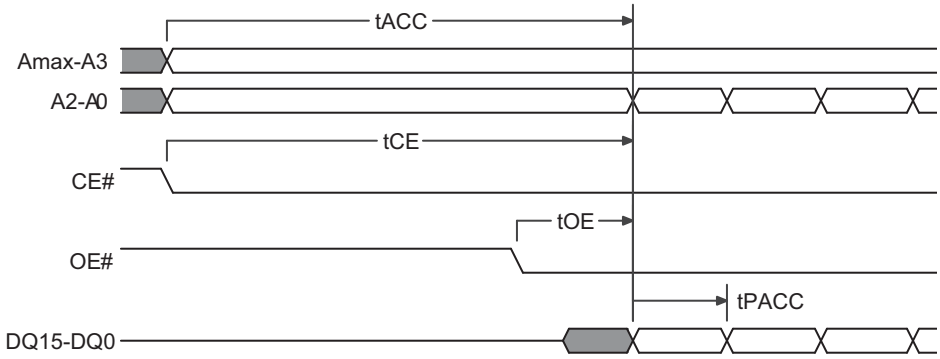

***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*
3. *Back to Back operations, in which CE# remains Low between accesses, requires an address change to initiate the second access.*



Downloaded from Arrow.com.

**Figure 25. Page Read Timing Diagram**




***Notes:***
1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

## 17.2 Asynchronous Write Operations

**Table 70. Write Operations**

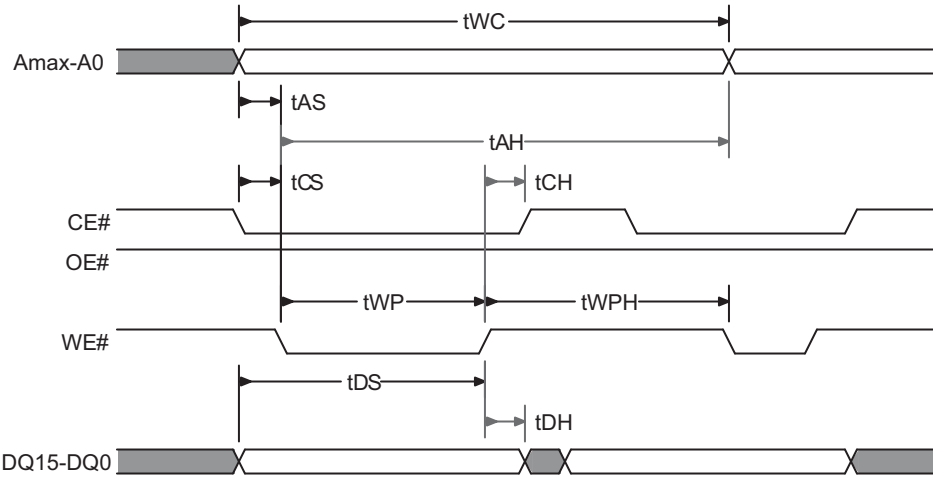
| Parameter | | Description | | $V_{IO}$ = 2.7V to $V_{CC}$ | $V_{IO}$ = 1.65V to $V_{CC}$ | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Std. | | | | | |
| $t_{AVAV}$ | $t_{WC}$ | Write Cycle Time (Note 1) | Min | 60 | | ns |
| $t_{AVWL}$ | $t_{AS}$ | Address Setup Time | Min | 0 | | ns |
| $t_{WLAX}$ | $t_{AH}$ | Address Hold Time | Min | 45 | | ns |
| $t_{DVWH}$ | $t_{DS}$ | Data Setup Time | Min | 30 | | ns |
| $t_{WHDX}$ | $t_{DH}$ | Data Hold Time | Min | 0 | | ns |
| $t_{GHWL}$ | $t_{GHWL}$ | Read Recovery Time Before Write (OE# High to WE# Low) | Min | 0 | | ns |
| $t_{ELWL}$ | $t_{CS}$ | CE# Setup Time | Min | 0 | | ns |
| $t_{WHEH}$ | $t_{CH}$ | CE# Hold Time | Min | 0 | | ns |
| $t_{WLWH}$ | $t_{WP}$ | Write Pulse Width | Min | 25 | | ns |
| $t_{WHDL}$ | $t_{WPH}$ | Write Pulse Width High | Min | 20 | | ns |
| | $t_{SEA}$ | Sector Erase Time-Out | Min | 50 | | µs |

***Notes:***
1. *Not 100% tested.*
2. *See the Erase and Programming Performance on page 97 for more information.*



Downloaded from Arrow.com.

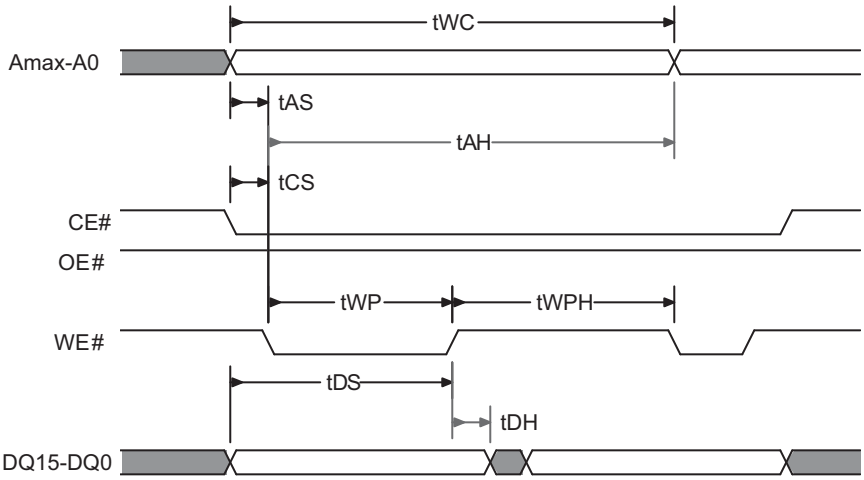
**Figure 26. Back to Back Write Operation Timing Diagram**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

**Figure 27. Back to Back (CE# $V_{IL}$) Write Operation Timing Diagram**



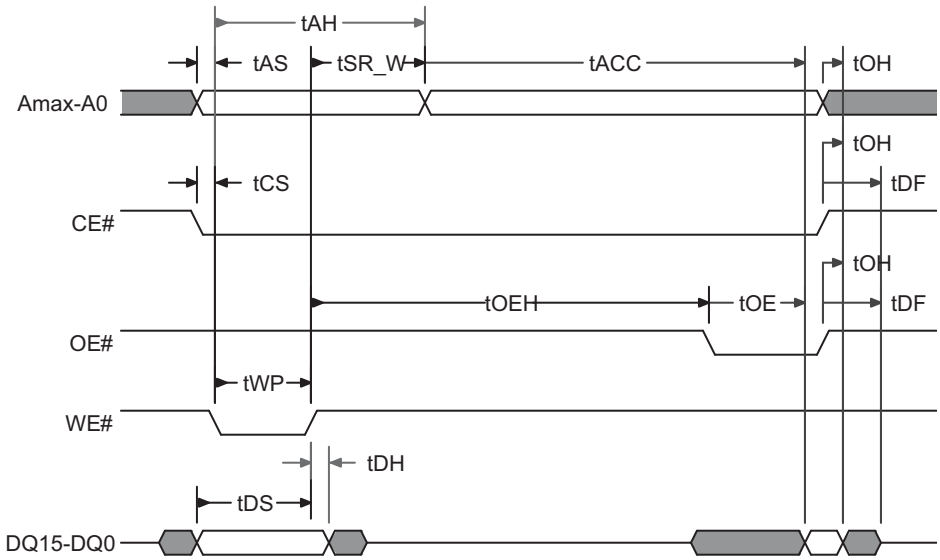

***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*



Downloaded from Arrow.com.

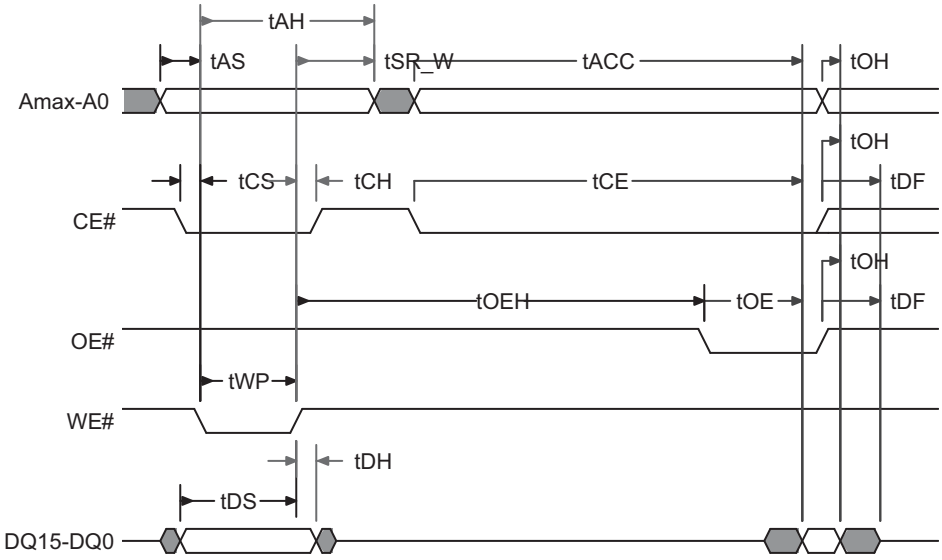
**Figure 28. Write to Read ($t_{ACC}$) Operation Timing Diagram**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

**Figure 29. Write to Read ($t_{CE}$) Operation Timing Diagram**



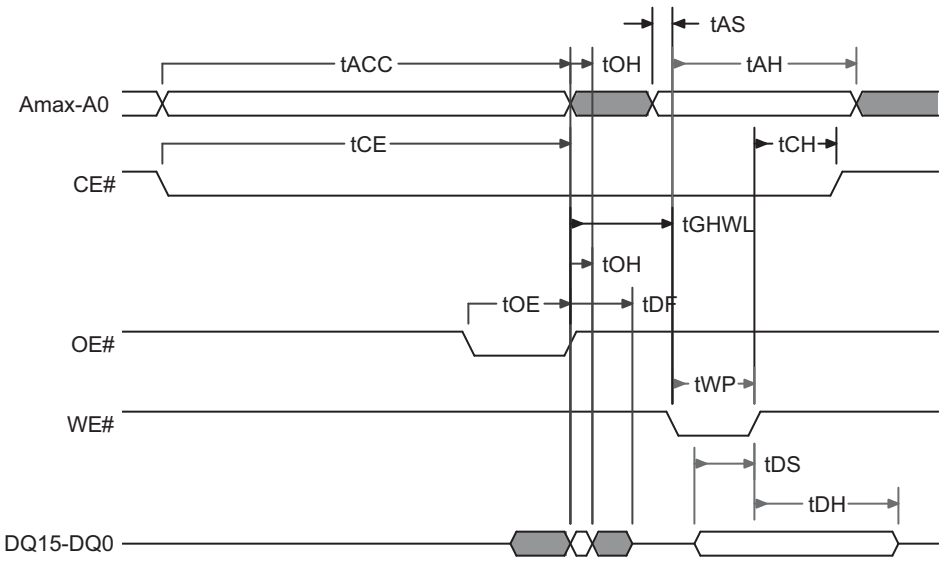

***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*



Downloaded from Arrow.com.

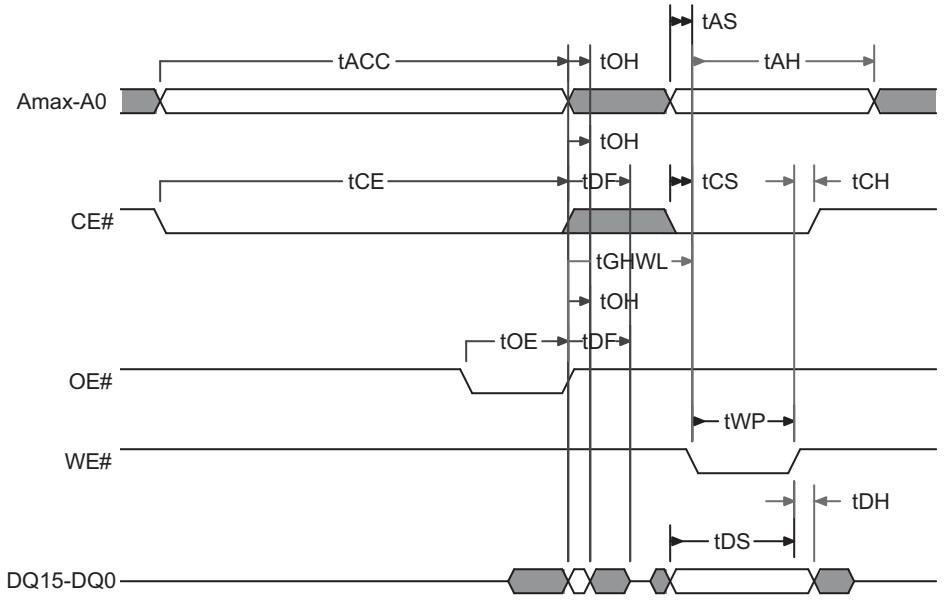
**Figure 30. Read to Write (CE# $V_{IL}$) Operation Timing Diagram**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

**Figure 31. Read to Write (CE# Toggle) Operation Timing Diagram**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*



Downloaded from Arrow.com.

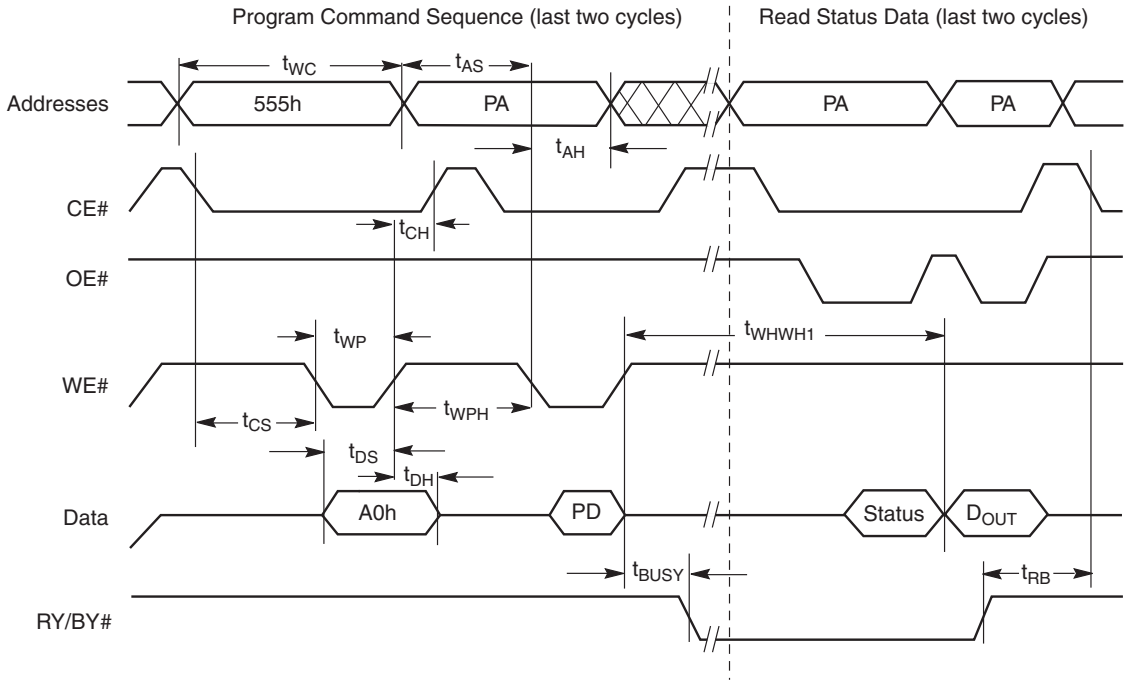
**Table 71. Erase / Program Operations**

| Parameter | | Description | | $V_{IO}$ = 2.7V to $V_{CC}$ | $V_{IO}$ = 1.65V to $V_{CC}$ | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | |
| $t_{WHWH1}$ | $t_{WHWH1}$ | Write Buffer Program Operation | Typ | (Note 3) | | µs |
| | | Effective Write Buffer Program Operation per Word | Typ | (Note 3) | | µs |
| | | Program Operation per Word or Page | Typ | (Note 3) | | µs |
| $t_{WHWH2}$ | $t_{WHWH2}$ | Sector Erase Operation (Note 1) | Typ | (Note 3) | | ms |
| | $t_{BUSY}$ | Erase / Program Valid to RY/BY# Delay | Max | 80 | | ns |
| | $t_{SR/W}$ | Latency between Read and Write operations (Note 2) | Min | 10 | | ns |
| | $t_{ESL}$ | Erase Suspend Latency | Max | (Note 3) | | µs |
| | $t_{PSL}$ | Program Suspend Latency | Max | (Note 3) | | µs |
| | $t_{RB}$ | RY/BY# Recovery Time | Min | 0 | | µs |
| | $t_{PPB}$ | PPB LOCK Unlock | Min | 80 | | µs |
| | | | Max | 120 | | |
| | $t_{DP}$ | Data Polling to Protected Sector (Program) | Min | 1 | | µs |
| | | Data Polling to Protected Sector (Erase) | Min | 100 | | µs |
| | $t_{TOR}$ | Exceeded Timing Cleared (DQ5) | Max | 2 | | µs |
| | $t_{VHH}$ | $V_{HH}$ Rise and Fall Time (Note 1) | Min | 250 | | ns |

***Notes:***

1. *Not 100% tested.*
2. *Upon the rising edge of WE#, must wait $t_{SR/W}$ before switching to another address.*
3. *See Table 73 on page 97 and Table 74 on page 98 for specific values.*

**Figure 32. Program Operation Timings**




***Notes:***

1. *PA = program address, PD = program data, $D_{OUT}$ is the true data at the program address.*
2. *Illustration shows device in word mode.*



Downloaded from Arrow.com.

**Figure 33. Accelerated Program Timing Diagram**

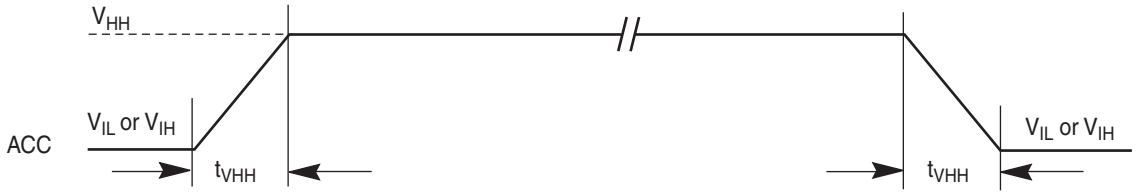
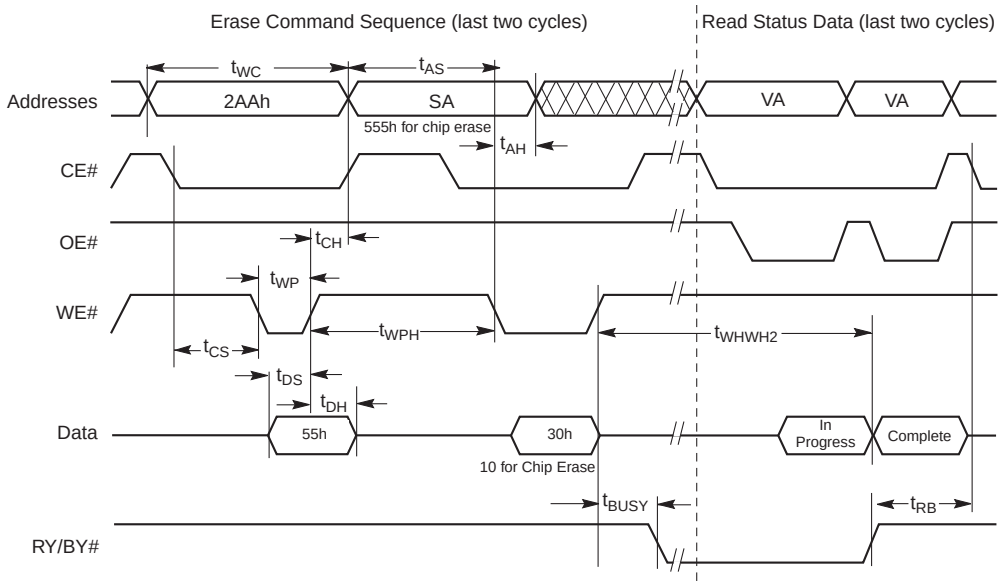


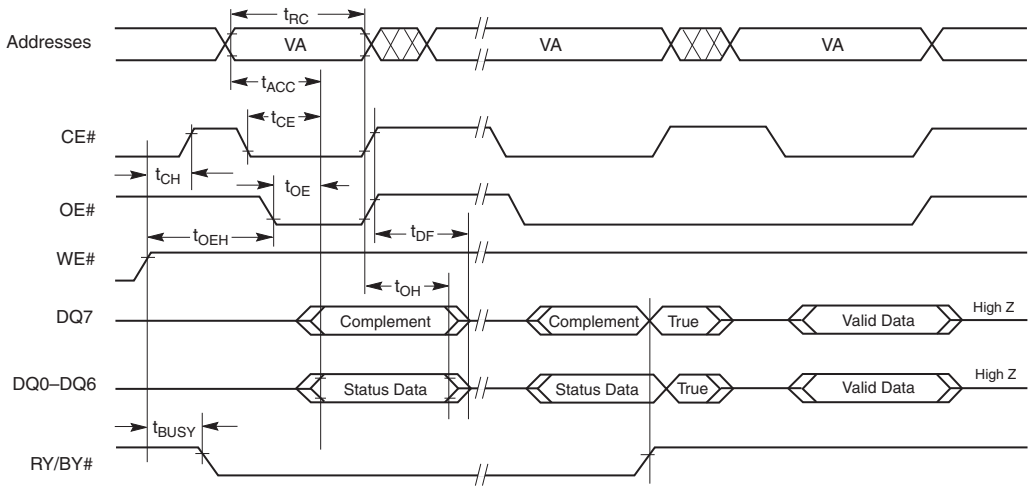

**Figure 34. Chip / Sector Erase Operation Timings**




***Notes:***

1. *SA = sector address (for Sector Erase), VA = Valid Address for reading status data (see Write Operation Status on page 51.)*
2. *Illustration shows device in word mode.*

**Figure 35. Data# Polling Timings (During Embedded Algorithms)**



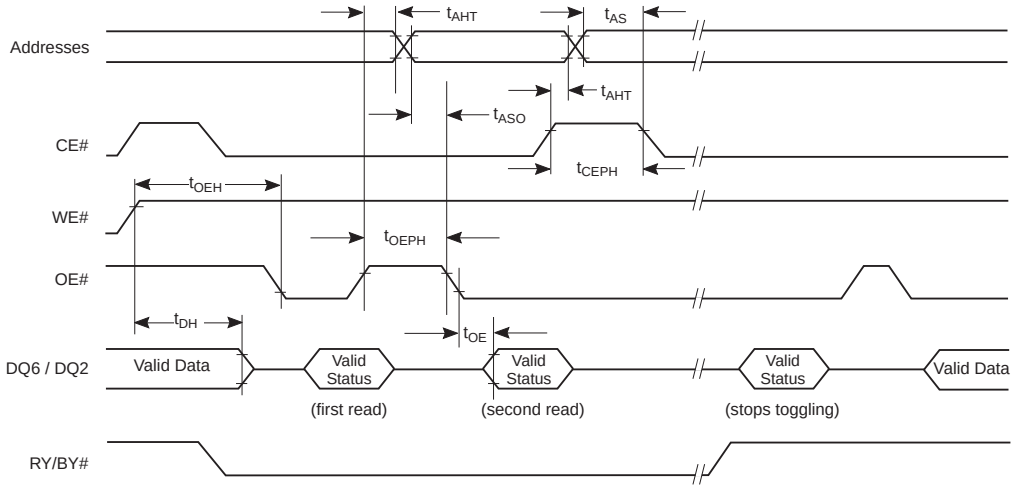

***Note:***

1. *VA = Valid address. Illustration shows first status cycle after command sequence, last status read cycle, and array data read cycle.*



Downloaded from Arrow.com.

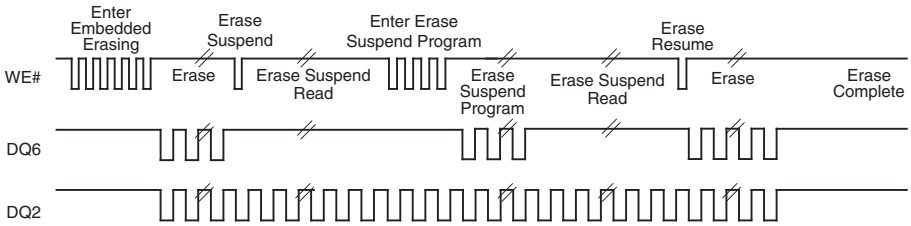
**Figure 36. Toggle Bit Timings (During Embedded Algorithms)**




***Notes:***

1. *VA = Valid address; not required for DQ6. Illustration shows first two status cycle after command sequence, last status read cycle, and array data read cycle.*
2. *CE# does not need to go high between status bit reads.*

**Figure 37. DQ2 vs. DQ6**




***Note:***

1. *DQ2 toggles only when read at an address within an erase-suspended sector. The system may use OE# or CE# to toggle DQ2 and DQ6.*

## 17.3 Alternative CE# Controlled Write Operations

**Table 72. Alternate CE# Controlled Erase and Program Operations**

| Parameter | | Description | | $V_{IO}$ = 2.7V to $V_{CC}$ | $V_{IO}$ = 1.65V to $V_{CC}$ | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Std. | | | | | |
| $t_{AVAV}$ | $t_{WC}$ | Write Cycle Time (Note 1) | Min | 60 | | ns |
| $t_{AVWL}$ | $t_{AS}$ | Address Setup Time | Min | 0 | | ns |
| $t_{ELAX}$ | $t_{AH}$ | Address Hold Time | Min | 45 | | ns |
| $t_{DVEH}$ | $t_{DS}$ | Data Setup Time | Min | 30 | | ns |
| $t_{EHDX}$ | $t_{DH}$ | Data Hold Time | Min | 0 | | ns |
| $t_{GHEL}$ | $t_{GHEL}$ | Read Recovery Time Before Write (OE# High to CE# Low) | Min | 0 | | ns |
| $t_{WLEL}$ | $t_{WS}$ | WE# Setup Time | Min | 0 | | ns |
| $t_{EHWH}$ | $t_{WH}$ | WE# Hold Time | Min | 0 | | ns |
| $t_{ELEH}$ | $t_{CP}$ | CE# Pulse Width | Min | 25 | | ns |
| $t_{EHEL}$ | $t_{CPH}$ | CE# Pulse Width High | Min | 20 | | ns |
| | $t_{SEA}$ | Sector Erase Time-Out | Min | 50 | | µs |

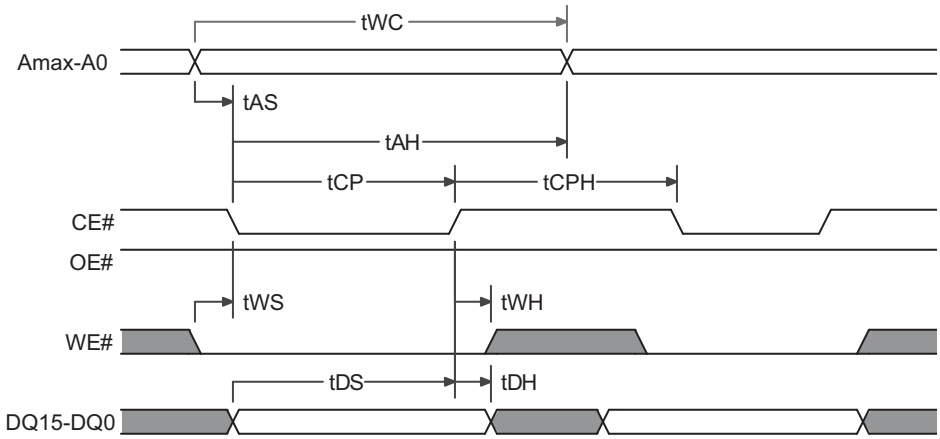
***Notes:***

1. *Not 100% tested.*
2. *See the Erase and Programming Performance on page 97 for more information.*



Downloaded from Arrow.com.

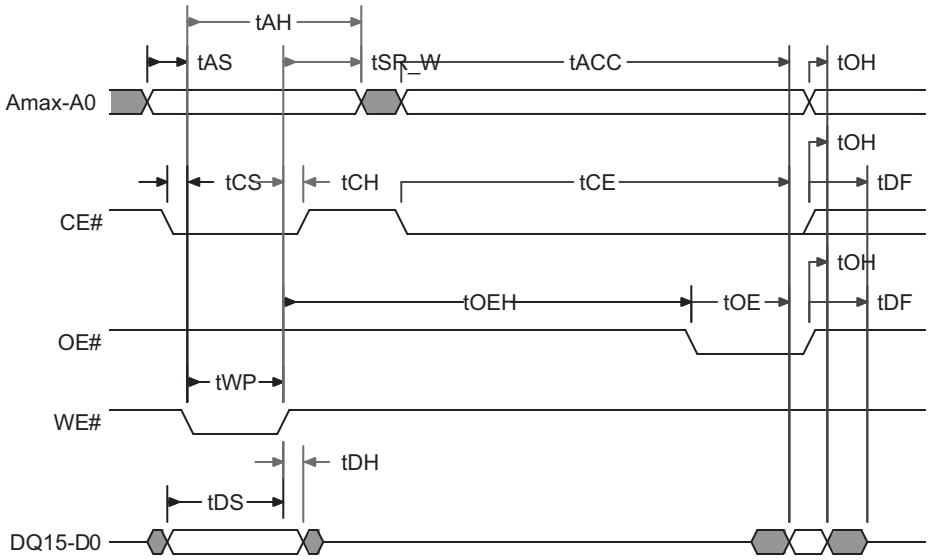
**Figure 38. Back to Back (CE#) Write Operation Timing Diagram**




***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*

**Figure 39. (CE#) Write to Read Operation Timing Diagram**



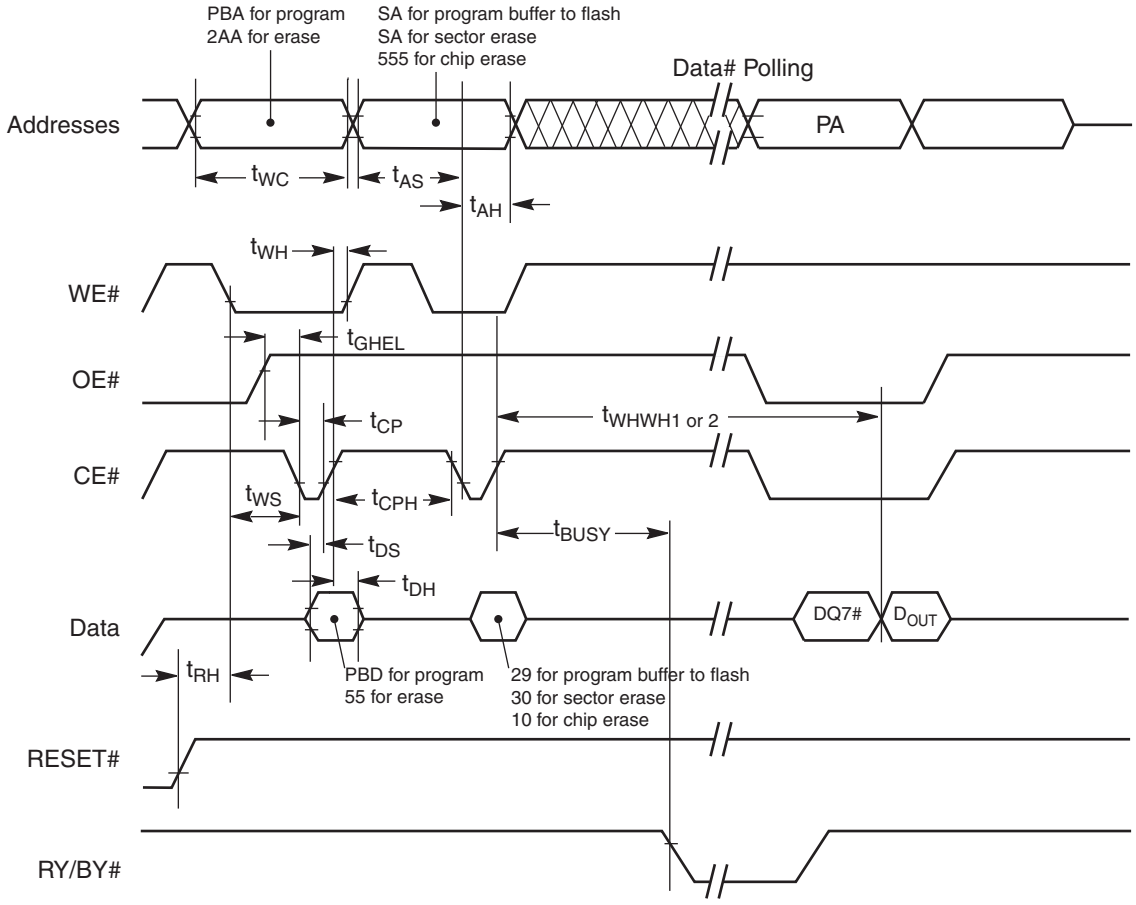

***Notes:***

1. *Word Configuration: Toggle A0, A1, and A2.*
2. *Byte Configuration: Toggle A-1, A0, A2, and A3.*



Downloaded from Arrow.com.

**Figure 17.1** Alternate CE# Controlled Write (Erase / Program) Operation Timings




***Notes:***

1. *Figure indicates last two bus cycles of a program or erase operation.*
2. *PA = program address, SA = sector address, PD = program data.*
3. *DQ7# is the complement of the data written to the device. $D_{OUT}$ is the data written to the device.*
4. *Illustration shows device in word mode.*



Downloaded from Arrow.com.

# 18. Erase and Programming Performance

The Joint Electron Device Engineering Council (JEDEC) standard JESD22-A117 defines the procedural requirements for performing valid endurance and retention tests based on a qualification specification. This methodology is intended to determine the ability of a flash device to sustain repeated data changes without failure (program / erase endurance) and to retain data for the expected life (data retention). Endurance and retention qualification specifications are specified in JESD47 or may be developed using knowledge-based methods as in JESD94.

**Table 73. Erase and Programming Performance for Industrial Temperature (-40°C to +85°C)**

| Parameter | | Min | Typ (Note 2) | Max (Note 3) | Unit | Comments |
|---|---|---|---|---|---|---|
| Sector Erase Time | 8 kB | | 235 | 1000 | ms | Includes 00h programming prior to erasure (Note 5) |
| | 64 kB | | 300 | 1000 | | |
| Chip Erase Time (Note 1) | | | 38.4 | 65.4 | sec | |
| Single Word Programming Time (Note 1) | | | 150 | 1200 | µs | |
| Buffer Program Time (Note 4) | 2 byte (Note 1) | | 150 | 1200 | µs | Excludes system level overhead (Note 6) |
| | 32 byte (Note 1) | | 200 | 1200 | | |
| | 64 byte (Note 1) | | 220 | 1200 | | |
| | 128 byte (Note 1) | | 300 | 1200 | | |
| | 256 byte | | 400 | 1200 | | |
| Effective Write Buffer Program Operation per Word | 256 byte | | 3.125 | | µs | |
| Total Accelerated Effective Write Buffer Program Time | 32 byte (Note 1) | | 200 | 1200 | µs | |
| | 64 byte | | 220 | 1200 | | |
| Effective Accelerated Write Buffer Program Operation per Word | 64 byte | | 6.9 | | µs | |
| Chip Program Time for a 128-word / 256-byte Write Buffer Operation (Note 1) | | | 13.11 | | sec | |
| Erase Suspend / Erase Resume ($t_{ESL}$) | | | | 30 | µs | |
| Program Suspend / Program Resume ($t_{PSL}$) | | | | 23.5 | µs | |
| Erase Resume to next Erase Suspend ($t_{ERS}$) | | | 100 | | µs | Minimum of 60 ns but ≥ typical periods are needed for Erase to progress to completion. |
| Program Resume to next Program Suspend ($t_{PRS}$) | | | 100 | | µs | Minimum of 60 ns but ≥ typical periods are needed for Program to progress to completion. |
| Evaluate Erase Status ($t_{EES}$) | | | 25 | 30 | µs | |

***Notes:***

1. *Not 100% tested.*
2. *Typical program and erase times assume the following conditions: 25°C, $V_{CC}$ = 3.0V, 10,000 cycles; random data pattern.*
3. *Under worst case conditions of 90°C; Worst case $V_{CC}$, 100,000 cycles, random pattern.*
4. *Write buffer Programming time is calculated on a per-word / per-byte basis for a 128-word / 256-byte write buffer operation.*
5. *In the pre-programming step of the Embedded Erase algorithm, all bits are programmed to 00h before erasure.*
6. *System-level overhead is the time required to execute the command sequence(s) for the program command. See Table 21 on page 42 and Table 23 on page 45 for further information on command definitions.*



Downloaded from Arrow.com.

**Table 74. Erase and Programming Performance for Industrial Plus Temperature (-40°C to +105°C)**

| Parameter | | Min | Typ (Note 2) | Max (Note 3) | Unit | Comments |
|---|---|---|---|---|---|---|
| Sector Erase Time | 8 kB | | 235 | 1000 | ms | Includes 00h programming prior to erasure (Note 5) |
| | 64 kB | | 300 | 1000 | | |
| Chip Erase Time (Note 1) | | | 38.4 | 65.4 | sec | |
| Single Word Programming Time (Note 1) | | | 150 | 1200 | µs | |
| Buffer Program Time (Note 4) | 2byte (Note 1) | | 150 | 1200 | µs | Excludes system level overhead (Note 6) |
| | 32 byte (Note 1) | | 200 | 1200 | | |
| | 64 byte (Note 1) | | 220 | 1200 | | |
| | 128 byte (Note 1) | | 300 | 1200 | | |
| | 256 byte (Note 1) | | 400 | 1200 | | |
| Effective Write Buffer Program Operation per Word | 256 byte | | 3.125 | | µs | |
| Total Accelerated Effective Write Buffer Program Time | 32 byte (Note 1) | | 200 | 1200 | µs | |
| | 64 byte | | 220 | 1200 | | |
| Effective Accelerated Write Buffer Program Operation per Word | 64 byte | | 6.9 | | µs | |
| Chip Program Time for a 128-word / 256-byte Write Buffer Operation (Note 1) | | | 13.11 | | sec | |
| Erase Suspend / Erase Resume ($t_{ESL}$) | | | | 30 | µs | |
| Program Suspend / Program Resume ($t_{PSL}$) | | | | 23.5 | µs | |
| Erase Resume to next Erase Suspend ($t_{ERS}$) | | | 100 | | µs | Minimum of 60 ns but ≥ typical periods are needed for Erase to progress to completion. |
| Program Resume to next Program Suspend ($t_{PRS}$) | | | 100 | | µs | Minimum of 60 ns but ≥ typical periods are needed for Program to progress to completion. |
| Evaluate Erase Status ($t_{EES}$) | | | 25 | 30 | µs | |

***Notes:***

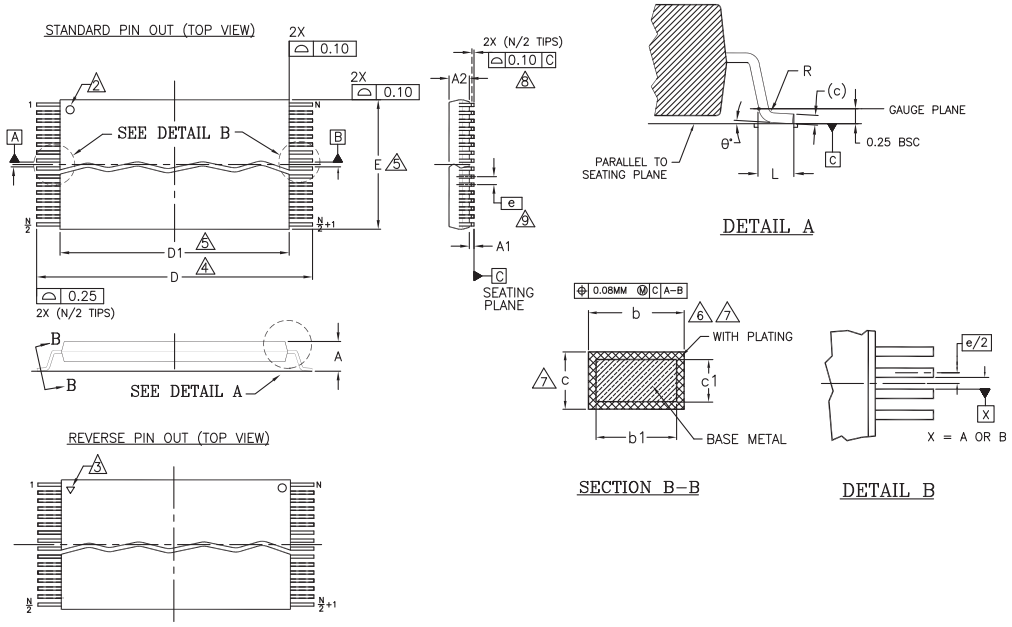
1. *Not 100% tested.*
2. *Typical program and erase times assume the following conditions: 25°C, $V_{CC}$ = 3.0V, 10,000 cycles; random data pattern.*
3. *Under worst case conditions of 110°C; Worst case $V_{CC}$, 100,000 cycles, random pattern.*
4. *Write buffer Programming time is calculated on a per-word / per-byte basis for a 128-word / 256-byte write buffer operation.*
5. *In the pre-programming step of the Embedded Erase algorithm, all bits are programmed to 00h before erasure.*
6. *System-level overhead is the time required to execute the command sequence(s) for the program command. See Table 21 on page 42 and Table 23 on page 45 for further information on command definitions.*



Downloaded from Arrow.com.

# 19. Physical Dimensions

## 19.1 TS048—48-Pin Standard Thin Small Outline Package (TSOP)




| PACKAGE | TS/TSR 48 | | |
|---|---|---|---|
| JEDEC | MO-142 (D) DD | | |
| SYMBOL | MIN | NOM | MAX |
| A | --- | --- | 1.20 |
| A1 | 0.05 | --- | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| b1 | 0.17 | 0.20 | 0.23 |
| b | 0.17 | 0.22 | 0.27 |
| c1 | 0.10 | --- | 0.16 |
| c | 0.10 | --- | 0.21 |
| D | 19.80 | 20.00 | 20.20 |
| D1 | 18.30 | 18.40 | 18.50 |
| E | 11.90 | 12.00 | 12.10 |
| e | 0.50 BASIC | | |
| L | 0.50 | 0.60 | 0.70 |
| θ | 0° | --- | 8 |
| R | 0.08 | --- | 0.20 |
| N | 48 | | |

NOTES:

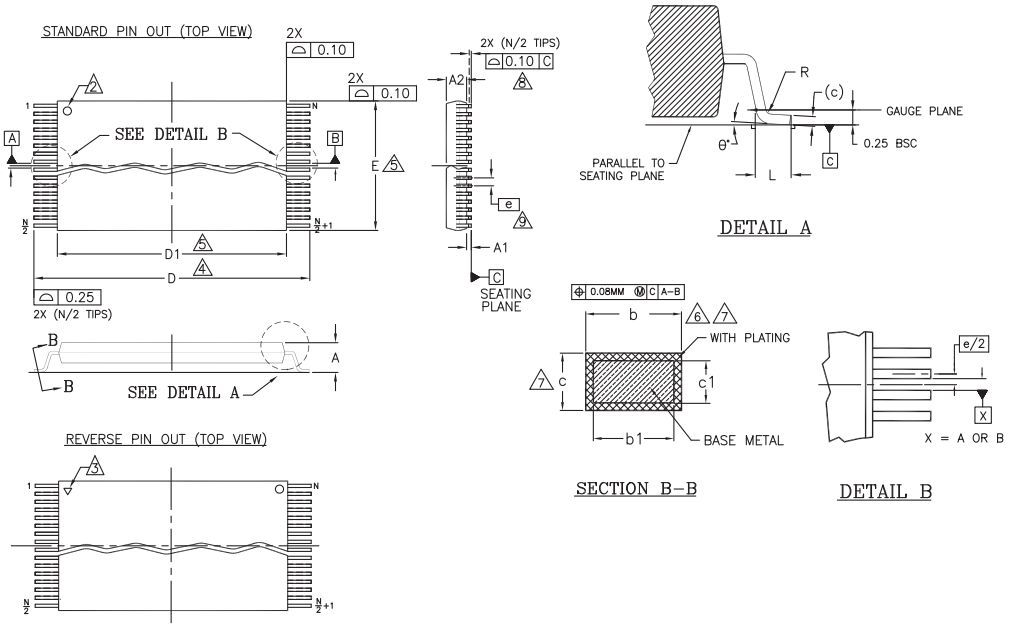
1. DIMENSIONS ARE IN MILLIMETERS (mm). (DIMENSIONING AND TOLERANCING CONFORM TO ANSI Y14.5M-1994).
2. PIN 1 IDENTIFIER FOR STANDARD PIN OUT (DIE UP).
3. PIN 1 IDENTIFIER FOR REVERSE PIN OUT (DIE DOWN): INK OR LASER MARK.
4. TO BE DETERMINED AT THE SEATING PLANE -C-. THE SEATING PLANE IS DEFINED AS THE PLANE OF CONTACT THAT IS MADE WHEN THE PACKAGE LEADS ARE ALLOWED TO REST FREELY ON A FLAT HORIZONTAL SURFACE.
5. DIMENSIONS D1 AND E DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION ON E IS 0.15mm PER SIDE AND ON D1 IS 0.25mm PER SIDE.
6. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08mm TOTAL IN EXCESS OF b DIMENSION AT MAX. MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND AN ADJACENT LEAD TO BE 0.07mm.
7. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10mm AND 0.25mm FROM THE LEAD TIP.
8. LEAD COPLANARITY SHALL BE WITHIN 0.10mm AS MEASURED FROM THE SEATING PLANE.
9. DIMENSION "e" IS MEASURED AT THE CENTERLINE OF THE LEADS.

5006 \ f16-038 \ 6.5.13



Downloaded from Arrow.com.

## 19.2 TS056—56-Pin Standard Thin Small Outline Package (TSOP)




| PACKAGE | TS/TSR 56 | | |
|---|---|---|---|
| JEDEC | MO-142 (D) EC | | |
| SYMBOL | MIN | NOM | MAX |
| A | --- | --- | 1.20 |
| A1 | 0.05 | --- | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| b1 | 0.17 | 0.20 | 0.23 |
| b | 0.17 | 0.22 | 0.27 |
| c1 | 0.10 | --- | 0.16 |
| c | 0.10 | --- | 0.21 |
| D | 19.80 | 20.00 | 20.20 |
| D1 | 18.30 | 18.40 | 18.50 |
| E | 13.90 | 14.00 | 14.10 |
| e | 0.50 BASIC | | |
| L | 0.50 | 0.60 | 0.70 |
| θ | 0° | --- | 8 |
| R | 0.08 | --- | 0.20 |
| N | 56 | | |

NOTES:

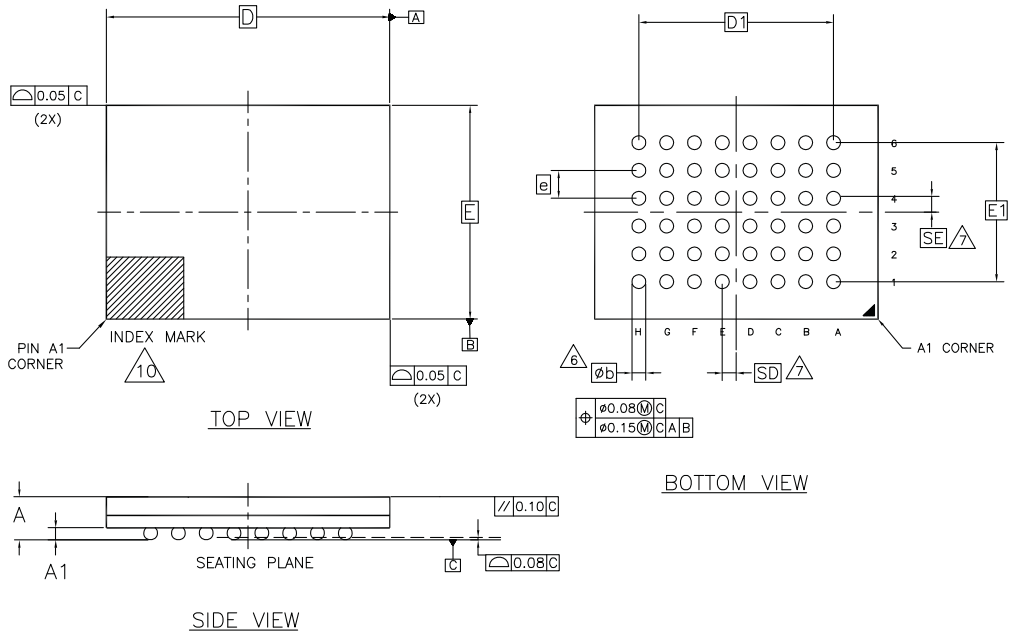
1. DIMENSIONS ARE IN MILLIMETERS (mm). (DIMENSIONING AND TOLERANCING CONFORM TO ANSI Y14.5M-1994).
2. PIN 1 IDENTIFIER FOR STANDARD PIN OUT (DIE UP).
3. PIN 1 IDENTIFIER FOR REVERSE PIN OUT (DIE DOWN): INK OR LASER MARK.
4. TO BE DETERMINED AT THE SEATING PLANE -C-. THE SEATING PLANE IS DEFINED AS THE PLANE OF CONTACT THAT IS MADE WHEN THE PACKAGE LEADS ARE ALLOWED TO REST FREELY ON A FLAT HORIZONTAL SURFACE.
5. DIMENSIONS D1 AND E DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION ON E IS 0.15mm PER SIDE AND ON D1 IS 0.25mm PER SIDE.
6. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08mm TOTAL IN EXCESS OF b DIMENSION AT MAX. MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND AN ADJACENT LEAD TO BE 0.07mm.
7. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10mm AND 0.25mm FROM THE LEAD TIP.
8. LEAD COPLANARITY SHALL BE WITHIN 0.10mm AS MEASURED FROM THE SEATING PLANE.
9. DIMENSION "e" IS MEASURED AT THE CENTERLINE OF THE LEADS.

5009 \ f16-038 \ 6.5.13



Downloaded from Arrow.com.

## 19.3 VBK048—Ball Fine-pitch Ball Grid Array (BGA) 8.15 x 6.15 mm Package




| PACKAGE | VBK 048 | | | |
|---|---|---|---|---|
| JEDEC | N/A | | | |
| | 8.15 mm x 6.15 mm NOM PACKAGE | | | |
| SYMBOL | MIN | NOM | MAX | NOTE |
| A | --- | --- | 1.00 | OVERALL THICKNESS |
| A1 | 0.18 | --- | --- | BALL HEIGHT |
| D | 8.15 BSC. | | | BODY SIZE |
| E | 6.15 BSC. | | | BODY SIZE |
| D1 | 5.60 BSC. | | | BALL FOOTPRINT |
| E1 | 4.00 BSC. | | | BALL FOOTPRINT |
| MD | 8 | | | ROW MATRIX SIZE D DIRECTION |
| ME | 6 | | | ROW MATRIX SIZE E DIRECTION |
| N | 48 | | | TOTAL BALL COUNT |
| ϕb | 0.33 | --- | 0.43 | BALL DIAMETER |
| e | 0.80 BSC. | | | BALL PITCH |
| SD / SE | 0.40 BSC. | | | SOLDER BALL PLACEMENT |
| | --- | | | DEPOPULATED SOLDER BALLS |

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. BALL POSITION DESIGNATION PER JESD 95-1, SPP-010 (EXCEPT AS NOTED).
4. e REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.

   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.

   N IS THE TOTAL NUMBER OF SOLDER BALLS.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.

   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.

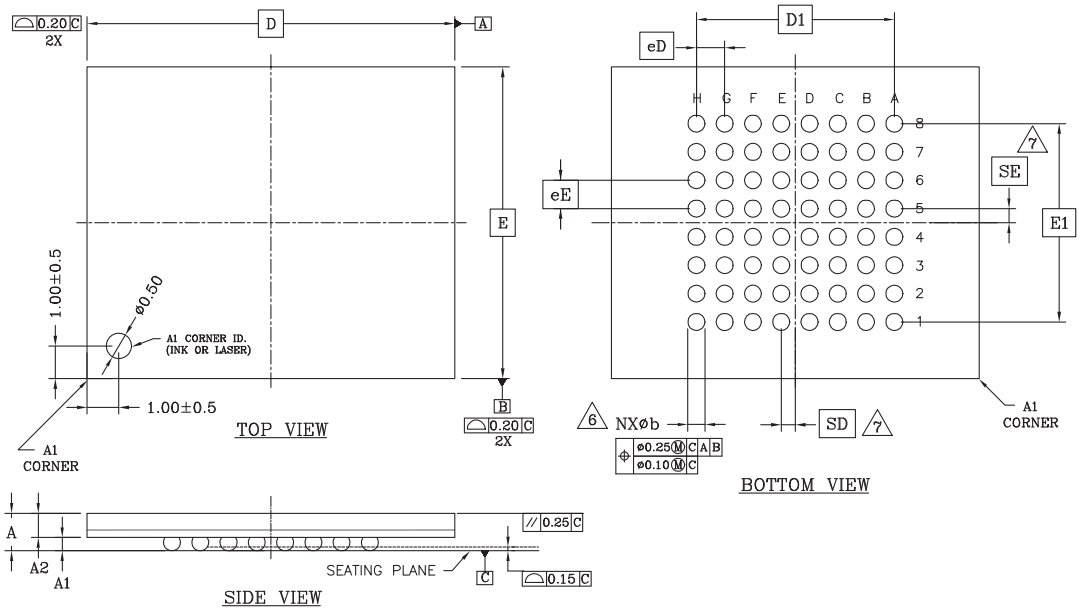
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = e/2
8. NOT USED.
9. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
10. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

g1001.2 \ f16-038.25 \ 07.13.10



Downloaded from Arrow.com.

## 19.4 LAA064—64-Ball Fortified Ball Grid Array (BGA) 13 x 11 mm Package




| PACKAGE | LAA 064 | | | |
|---|---|---|---|---|
| JEDEC | N/A | | | |
| | 13.00 mm x 11.00 mm PACKAGE | | | |
| SYMBOL | MIN | NOM | MAX | NOTE |
| A | --- | --- | 1.40 | PROFILE HEIGHT |
| A1 | 0.40 | --- | --- | STANDOFF |
| A2 | 0.60 | --- | --- | BODY THICKNESS |
| D | 13.00 BSC. | | | BODY SIZE |
| E | 11.00 BSC. | | | BODY SIZE |
| D1 | 7.00 BSC. | | | MATRIX FOOTPRINT |
| E1 | 7.00 BSC. | | | MATRIX FOOTPRINT |
| MD | 8 | | | MATRIX SIZE D DIRECTION |
| ME | 8 | | | MATRIX SIZE E DIRECTION |
| N | 64 | | | BALL COUNT |
| ϕb | 0.50 | 0.60 | 0.70 | BALL DIAMETER |
| eD | 1.00 BSC. | | | BALL PITCH - D DIRECTION |
| eE | 1.00 BSC. | | | BALL PITCH - E DIRECTION |
| SD / SE | 0.50 BSC. | | | SOLDER BALL PLACEMENT |
| | NONE | | | DEPOPULATED SOLDER BALLS |

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. BALL POSITION DESIGNATION PER JESD 95-1, SPP-010 (EXCEPT AS NOTED).
4. e REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.

   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.

   N IS THE TOTAL NUMBER OF SOLDER BALLS.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.

   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.

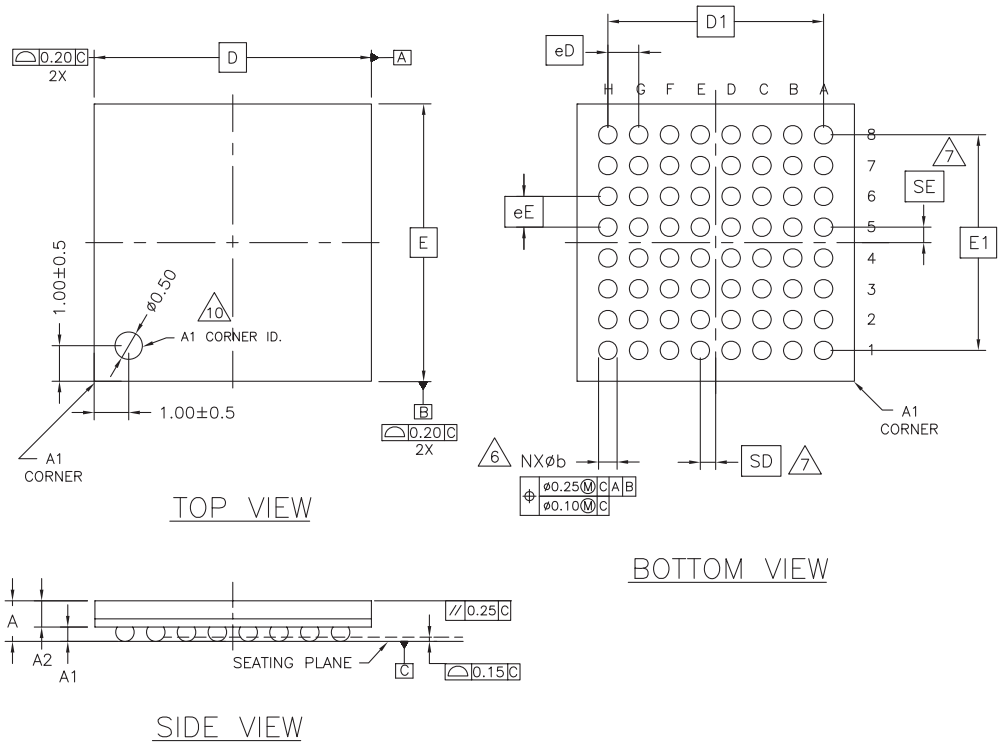
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = e/2
8. NOT USED.
9. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.

3354 \ 16-038.12d



Downloaded from Arrow.com.

## 19.5 LAE064—64-Ball Fortified Ball Grid Array (BGA) 9 x 9 mm Package




| PACKAGE | LAE 064 | | | |
|---|---|---|---|---|
| JEDEC | N/A | | | |
| | 9.00 mm x 9.00 mm PACKAGE | | | |
| SYMBOL | MIN | NOM | MAX | NOTE |
| A | --- | --- | 1.40 | PROFILE HEIGHT |
| A1 | 0.40 | --- | --- | STANDOFF |
| A2 | 0.60 | --- | --- | BODY THICKNESS |
| D | 9.00 BSC. | | | BODY SIZE |
| E | 9.00 BSC. | | | BODY SIZE |
| D1 | 7.00 BSC. | | | MATRIX FOOTPRINT |
| E1 | 7.00 BSC. | | | MATRIX FOOTPRINT |
| MD | 8 | | | MATRIX SIZE D DIRECTION |
| ME | 8 | | | MATRIX SIZE E DIRECTION |
| N | 64 | | | BALL COUNT |
| øb | 0.50 | 0.60 | 0.70 | BALL DIAMETER |
| eD | 1.00 BSC. | | | BALL PITCH - D DIRECTION |
| eE | 1.00 BSC. | | | BALL PITCH - E DIRECTION |
| SD / SE | 0.50 BSC. | | | SOLDER BALL PLACEMENT |
| | NONE | | | DEPOPULATED SOLDER BALLS |

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. BALL POSITION DESIGNATION PER JESD 95-1, SPP-010 EXCEPT AS NOTED).
4. e REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.

   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.

   N IS THE TOTAL NUMBER OF SOLDER BALLS.

6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.

   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.

   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = e/2

8. NOT USED.
9. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.

3623 \ 16-038.12 \ 1.16.07



Downloaded from Arrow.com.

# 20. Revision History

## Document History Page

| Document Title: S29GL064S, 64-Mbit (8 Mbyte), 3.0 V, Flash Memory<br>Document Number: 001-98286 | | | | |
|---|---|---|---|---|
| **Rev.** | **ECN No.** | **Orig. of Change** | **Submission Date** | **Description of Change** |
| ** | – | BWHA | 12/11/2013 | Initial release |
| *A | – | BWHA | 03/11/2014 | **Global:**<br>Changed data sheet designation from *Advance Information* to *Preliminary* |
| *B | – | BWHA | 04/16/2014 | **Common Flash Memory Interface (CFI):**<br>Corrected values for Addresses (x16) 22h, 26h, 2Ah, 45h<br>Clarified values for Addresses (x16) 4Eh and 2Eh |
| *C | – | BWHA | 11/26/2014 | **Global:**<br>Changed 'Automotive In-Cabin' Temperature Range to 'Industrial Plus' Temperature Range.<br>**Common Flash Memory Interface (CFI):**<br>System Interface String table: updated values for Addresses (x16) 21h, 22h, 25h, 26h<br>**Command Definitions:**<br>Sector Protection Commands (x16) table: corrected 'Password Protection' Addr value to '00'<br>**Power-On Reset (POR) and Warm Reset:**<br>Power-On and Reset Parameters table: corrected Value for $T_{RPH}$<br>**Erase and Programming Performance:**<br>Erase and Programming Performance for Industrial Temperature (-40°C to +85°C) table: updated Erase Times<br>Erase and Programming Performance for In Cabin Temperature (-40°C to +105°C) table: updated Erase Times<br>**AC Characteristics:**<br>Added Notes to Figures 15.1 - 15.3, 15.5 - 15.10, 15.17 - 15.19 |
| *D | 4871480 | BWHA | 08/13/2015 | Updated to Cypress template. |
| *E | 5560152 | BWHA | 01/04/2017 | Changed status from Preliminary to Final.<br>Added ECC related information in all instances across the document.<br>Added "Automotive AEC-Q100 Grade 3" and "Automotive AEC-Q100 Grade 2" Temperature Range related information in all instances across the document.<br>Updated Ordering Information:<br>Added Automotive AEC-Q100 Grade 2 and Automotive AEC-Q100 Grade 3 Temperature Range details.<br>Updated Valid Combinations:<br>Added Automotive AEC-Q100 Grade 2 and Automotive AEC-Q100 Grade 3 Temperature Range details.<br>Added Other Resources.<br>Updated Device Bus Operations:<br>Added Automatic ECC.<br>Updated Command Definitions:<br>Added ECC Status ASO.<br>Added Data Integrity.<br>Updated Electrical Specifications:<br>Added Thermal Resistance.<br>Updated Erase and Programming Performance:<br>Updated Table 73.<br>Updated Table 74.<br>Completing Sunset Review. |
| *F | 5767666 | AESATMP8 | 06/08/2017 | Updated logo and Copyright. |



Downloaded from Arrow.com.

## Document History Page (Continued)

| Document Title: S29GL064S, 64-Mbit (8 Mbyte), 3.0 V, Flash Memory<br>Document Number: 001-98286 | | | | |
|---|---|---|---|---|
| **Rev.** | **ECN No.** | **Orig. of Change** | **Submission Date** | **Description of Change** |
| *G | 5951514 | PRIT | 10/30/2017 | Updated Device Bus Operations:<br>Updated Automatic ECC:<br>Updated Write Buffer Programming:<br>Updated description.<br>Updated to new template.<br>Completing Sunset Review. |
| *H | 6214237 | PRIT | 08/03/2018 | Updated Ordering Information:<br>Updated details corresponding to “F” and “H” under “Package Materials Set” in the diagram.<br>Added a note “Halogen free definition is in accordance with IEC 61249-2-21 specification” and referred the same note in “F” and “H”.<br>Updated Command Definitions:<br>Updated Command Definitions:<br>Updated Table 23:<br>Updated Note 12.<br>Updated to new template. |



Downloaded from Arrow.com.

## Sales, Solutions, and Legal Information

### Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

### Products

| | |
|---|---|
| Arm® Cortex® Microcontrollers | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Internet of Things | cypress.com/iot |
| Memory | cypress.com/memory |
| Microcontrollers | cypress.com/mcu |
| PSoC | cypress.com/psoc |
| Power Management ICs | cypress.com/pmic |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless Connectivity | cypress.com/wireless |

### PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

### Cypress Developer Community

Community | Projects | Video | Blogs | Training | Components

### Technical Support

cypress.com/support

© Cypress Semiconductor Corporation, 2013-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.



Downloaded from Arrow.com.