**Please note that Cypress is an Infineon Technologies Company.**
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

**Continuity of document content**
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

**Continuity of ordering part numbers**
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.




"Spansion, Inc." and "Cypress Semiconductor Corp." have merged together to deliver high-performance, high-quality solutions at the heart of today's most advanced embedded systems, from automotive, industrial and networking platforms to highly interactive consumer and mobile devices. The new company "Cypress Semiconductor Corp." will continue to offer "Spansion, Inc." products to new and existing customers.

## CONTINUITY OF SPECIFICATIONS

There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

## CONTINUITY OF ORDERING PART NUMBERS

Cypress continues to support existing part numbers. To order these products, please use only the Ordering Part Numbers listed in this document.

## FOR MORE INFORMATION

Please visit our website at www.cypress.com or contact your local sales office for additional information about Cypress products and services.

## OUR CUSTOMERS

Cypress is for true innovators – in companies both large and small.

Our customers are smart, aggressive, out-of-the-box thinkers who design and develop game-changing products that revolutionize their industries or create new industries with products and solutions that nobody ever thought of before.

## ABOUT CYPRESS

Founded in 1982, Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, home automation and appliances, consumer electronics and medical products. Cypress's programmable systems-on-chip, general-purpose microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first.

Cypress is committed to providing customers with the best support and engineering resources on the planet enabling innovators and out-of-the-box thinkers to disrupt markets and create new product categories in record time. To learn more, go to www.cypress.com.

# S29VS064R, S29VS032R
# S29XS064R, S29XS032R

# 64/32 Mbit (4/2M x 16-bit), 1.8 V MirrorBit Flash

## Distinctive Characteristics

- Single 1.8 volt read, program and erase (1.7 to 1.95 volt)
- VersatileIO™ Feature
  - Device generates data output voltages and tolerates data input voltages as determined by the voltage on the $V_{CCQ}$ pin
  - 1.8 V compatible I/O signals
- Address and Data Interface Options
  - Address and Data Multiplexed for reduced I/O count (ADM) S29VS-R
  - Address-High, Address-Low, Data Multiplexed for minimum I/O count (AADM) S29XS-R
- Simultaneous Read/Write operation
  - Data can be continuously read from one bank while executing erase/program functions in other bank
  - Zero latency between read and write operations
- Burst length
  - Continuous linear burst
  - 8/16 word linear burst with wrap around
- Secured Silicon Sector region
  - 256 words accessible through a command sequence, 128 words for the Factory Secured Silicon Sector and 128 words for the Customer Secured Silicon Sector.
- Sector Architecture
  - Four 8 Kword sectors in upper-most address range
  - Sixty three 32 Kword sectors (VS/XS032R)
  - One hundred twenty-seven 32 Kword sectors (VS/XS064R)
  - Four banks

### Security Features

- Dynamic Protection Bit (DYB)
  - A command sector protection method to lock combinations of individual sectors to prevent program or erase operations within that sector
  - Sectors can be locked and unlocked in-system at $V_{CC}$ level
- Hardware Sector Protection
  - All sectors locked when $V_{PP} = V_{IL}$
- Handshaking feature
  - Provides host system with minimum possible latency by monitoring RDY
- Supports Common Flash Memory Interface (CFI)
- Manufactured on 65 nm MirrorBit® process technology
- Cycling endurance: 100,000 cycles per sector typical
- Data retention: 10 years typical
- Data# Polling and toggle bits
  - Provides a software method of detecting program and erase operation completion
- Erase Suspend/Resume
  - Suspends an erase operation to read data from, or program data to, a sector that is not being erased, then resumes the erase operation
- Program Suspend/Resume
  - Suspends a programming operation to read data from a sector other than the one being programmed, then resume the programming operation
- Packages
  - 44-ball Very Thin FBGA

## Performance Characteristics

| Read Access Times | |
|---|---|
| Speed Option (MHz) | 104 |
| Max. Synch. Latency, ns ($t_{IACC}$) | 80 |
| Max. Synch. Burst Access, ns ($t_{BACC}$) | 7.6 |
| Max. Asynch. Access Time, ns ($t_{ACC}$) | 80 |
| Max OE# Access Time, ns ($t_{OE}$) | 15 |

| Typical Program & Erase Times | |
|---|---|
| Single Word Programming | 130 µs |
| Effective Write Buffer Programming ($V_{CC}$) Per Word | 9.4 µs |
| Effective Write Buffer Programming ($V_{PP}$) Per Word | 6 µs |
| Sector Erase (8 Kword Sector) | 350 ms |
| Sector Erase (32 Kword Sector) | 800 ms |

| Current Consumption (typical values) | |
|---|---|
| Continuous Burst Read @ 104 MHz | 32 mA |
| Simultaneous Operation @ 104 MHz | 52 mA |
| Program/Erase | 19 mA |
| Standby Mode | 20 µA |

# 1. General Description

The S29V/XS064R and S29V/XS032R are 64 Mb/32 Mb, 1.8 Volt-only, Simultaneous Read/Write, Burst Mode Flash memory devices, organized as 4,194,304/2,097,152 words of 16 bits each. These devices use a single $V_{CC}$ of 1.70 to 1.95 V to read, program, and erase the memory array. A 9.0-volt $V_{PP}$, may be used for faster program performance if desired. These devices can also be programmed in standard EPROM programmers.

The devices operate within the temperature range of –25°C to +85°C, and are offered in Very Thin FBGA packages.

## 1.1 Simultaneous Read/Write Operations with Zero Latency

The Simultaneous Read/Write architecture provides **simultaneous operation** by dividing the memory space into four banks. The device allows a host system to program or erase in one bank, then immediately and simultaneously read from another bank, with zero latency. This releases the system from waiting for the completion of program or erase operations.

The VersatileIO™ ($V_{IO}$) control allows the host system to set the voltage levels that the device generates at its data outputs and the voltages tolerated at its data inputs to the same voltage level that is asserted on the $V_{CCQ}$ pin.

The devices use Chip Enable (CE#), Write Enable (WE#), Address Valid (AVD#) and Output Enable (OE#) to control asynchronous read and write operations. For burst operations, the devices additionally require Ready (RDY) and Clock (CLK). This implementation allows easy interface with minimal glue logic to microprocessors/microcontrollers for high performance read operations.

The devices offer complete compatibility with the **JEDEC 42.4 single-power-supply Flash command set standard**. Commands are written to the command register using standard microprocessor write timings. Reading data out of the device are similar to reading from other Flash or EPROM devices.

The host system can detect whether a program or erase operation is complete by using the device **status bit** DQ7 (Data# Polling) and DQ6/DQ2 (toggle bits). After a program or erase cycle has been completed, the device automatically returns to reading array data.

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

**Hardware data protection** measures include a low $V_{CC}$ detector that automatically inhibits write operations during power transitions. The devices also offers another type of data protection at the sector level. When $V_{PP}$ is at $V_{IL}$, all sectors are locked.

The devices offer two power-saving features. When addresses have been stable for a specified amount of time, the device enters the **automatic sleep mode**. The system can also place the device into the **standby mode**. Power consumption is greatly reduced in both modes.

Device programming occurs by executing the program command sequence. This initiates the **Embedded Program** algorithm - an internal algorithm that automatically times the program pulse widths and verifies proper cell margin. Additionally, **Write Buffer Programming** is available on this family of devices. This feature provides superior programming performance by grouping locations being programmed.

Device erasure occurs by executing the erase command sequence. This initiates the **Embedded Erase** algorithm - an internal algorithm that automatically preprograms the array (if it is not already fully programmed) before executing the erase operation. During erase, the device automatically times the erase pulse widths and verifies proper cell margin.

The **Program Suspend/Program Resume** feature enables the user to put program on hold to read data from any sector that is not selected for programming. If a read is needed from the Dynamic Protection area, or the CFI area, after an program suspend, then the user must use the proper command sequence to enter and exit this region. The program suspend/resume functionality is also available when programming in erase suspend (1 level depth only).

The **Erase Suspend/Erase Resume** feature enables the user to put erase on hold to read data from, or program data to, any sector that is not selected for erasure. True background erase can thus be achieved. If a read is needed from the Dynamic Protection area, or the CFI area, after an erase suspend, then the user must use the proper command sequence to enter and exit this region.

The **hardware RESET# pin** terminates any operation in progress and resets the internal state machine to reading array data. The RESET# pin may be tied to the system reset circuitry. A system reset would thus also reset the device, enabling the system microprocessor to read boot-up firmware from the Flash memory device.

The host system can detect whether a memory array program or erase operation is complete by using the device status bit DQ7 (Data# Polling), DQ6/DQ2 (toggle bits), DQ5 (exceeded timing limit), DQ3 (sector erase start timeout state indicator), and DQ1 (write to buffer abort). After a program or erase cycle has been completed, the device automatically returns to reading array data.

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**.

**Hardware data protection** measures include a low $V_{CC}$ detector that automatically inhibits write operations during power transitions. The device also offers another type of data protection at the sector level.

When the $V_{PP}$ pin = $V_{IL}$, the entire flash memory array is protected.

Spansion Inc. 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. The data is programmed using hot electron injection.

# Table of Contents

**Distinctive Characteristics** ..... 2

**Performance Characteristics** ..... 2

**1. General Description** ..... 2
1.1 Simultaneous Read/Write Operations with Zero Latency ..... 3

**2. Block Diagrams** ..... 6

**3. Connection Diagram** ..... 8

**4. Physical Dimensions** ..... 9

**5. Input/Output Descriptions** ..... 10

**6. Logic Symbol** ..... 10

**7. Ordering Information** ..... 11
7.1 Valid Combinations ..... 11

**8. Address/Data Configuration (Interface) Modes** ..... 12
8.1 ADM Interface Mode (S29VS064R, S29VS032R) ..... 12
8.2 AADM Interface Mode (S29XS064R, S29XS032R) ..... 12

**9. Device Bus Operations** ..... 13
9.1 VersatileIO™ ($V_{IO}$) Control ..... 13
9.2 Asynchronous Read ..... 14
9.3 Synchronous (Burst) Read Mode and Configuration Register ..... 14
9.4 Programmable Wait State ..... 18
9.5 Configuration Register ..... 18
9.6 Handshaking Feature ..... 18
9.7 Simultaneous Read/Write Operations with Zero Latency ..... 18
9.8 Writing Commands/Command Sequences ..... 18
9.9 Accelerated Program and Erase Operations ..... 18
9.10 Write Buffer Programming Operation ..... 19
9.11 Autoselect Mode ..... 20
9.12 Sector Protection ..... 20
9.13 Hardware Data Protection Mode ..... 20
9.14 Low VCC Write Inhibit ..... 20
9.15 Write Pulse “Glitch” Protection ..... 21
9.16 Logical Inhibit ..... 21
9.17 Lock Register ..... 21
9.18 Standby Mode ..... 21
9.19 Automatic Sleep Mode ..... 21
9.20 RESET#: Hardware Reset Input ..... 21
9.21 Output Disable Mode ..... 22
9.22 Secured Silicon Sector Flash Memory Region ..... 22

**10. Common Flash Memory Interface (CFI)** ..... 23

**11. Command Definitions** ..... 28
11.1 Reading Array Data ..... 28
11.2 Set Configuration Register Command Sequence ..... 28
11.3 Read Configuration Register Command Sequence ..... 28

**12. Configuration Register** ..... 31
12.1 Reset Command ..... 31
12.2 Autoselect Command Sequence ..... 32
12.3 Enter/Exit Secured Silicon Sector \ Command Sequence ..... 32
12.4 Program Command Sequence ..... 32
12.5 Accelerated Program ..... 32
12.6 Write Buffer Programming Command Sequence ..... 33
12.7 Chip Erase Command Sequence ..... 34
12.8 Sector Erase Command Sequence ..... 35
12.9 Erase Suspend/Erase Resume Commands ..... 36
12.10Program Suspend/Program Resume Commands ..... 37
12.11Volatile Sector Protection Command Set ..... 37

**13. Write Operation Status** ..... 40
13.1 DQ7: Data# Polling ..... 40
13.2 RDY: Ready ..... 41
13.3 DQ6: Toggle Bit I ..... 42
13.4 DQ2: Toggle Bit II ..... 42
13.5 Reading Toggle Bits DQ6/DQ2 ..... 44
13.6 DQ5: Exceeded Timing Limits ..... 44
13.7 DQ3: Sector Erase Start Timeout State Indicator ..... 44
13.8 DQ1: Write to Buffer Abort ..... 44

**14. Absolute Maximum Ratings** ..... 46

**15. Operating Ranges** ..... 46

**16. DC Characteristics** ..... 47

**17. Test Conditions** ..... 48

**18. Key to Switching Waveforms** ..... 48
18.1 Switching Waveforms ..... 48

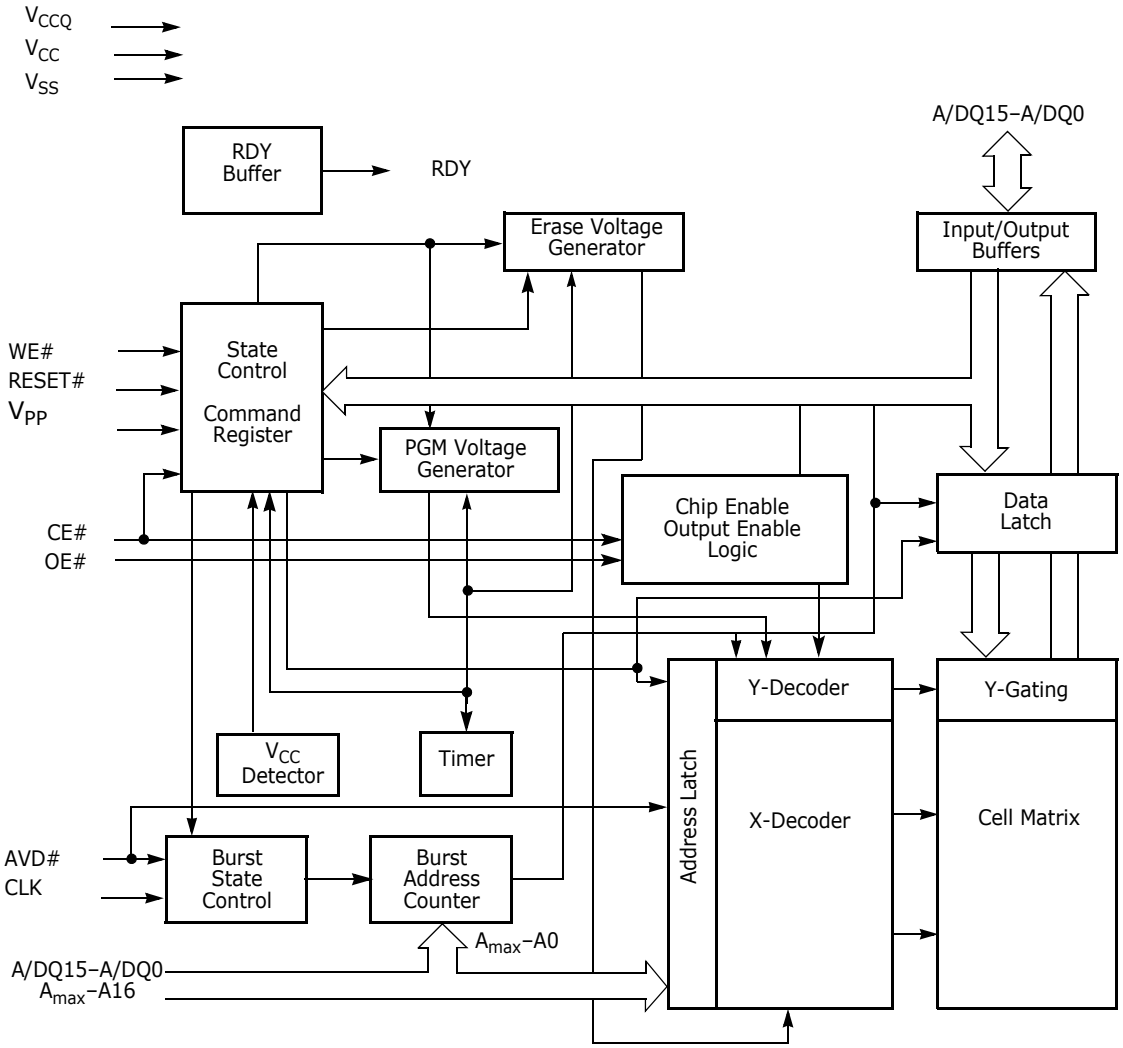
**19. AC Characteristics** ..... 49
19.1 $V_{CC}$ Power-up ..... 49
19.2 Synchronous/Burst Read ..... 50
19.3 Asynchronous Read ..... 52
19.4 Hardware Reset (RESET#) ..... 52
19.5 Erase/Program Operations ..... 53

**20. Erase and Programming Performance** ..... 59

**21. BGA Ball Capacitance** ..... 59

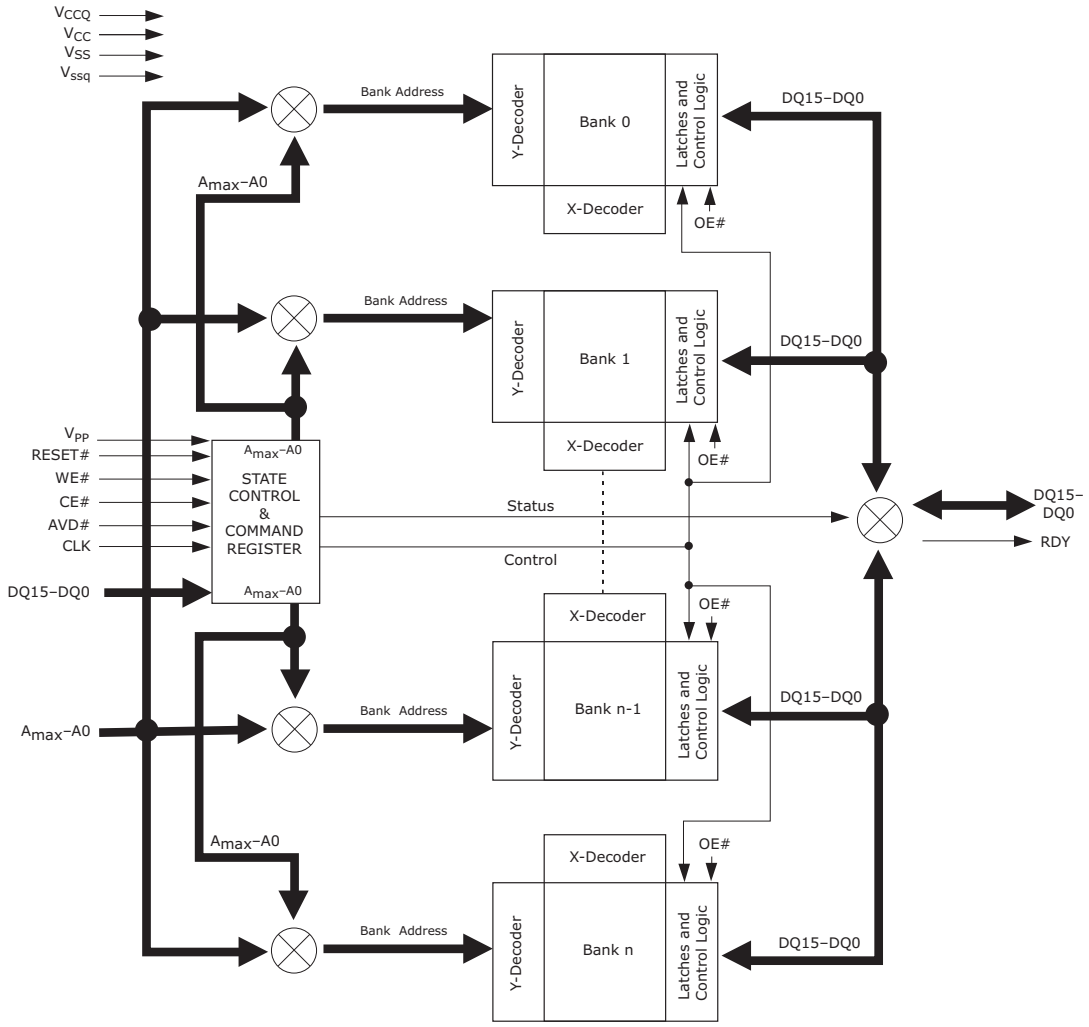
**22. Revision History** ..... 60

## 2. Block Diagrams




***Note***
*$A_{max}$ indicates the highest order address bit. $A_{max}$ equals A21 for S29V/XS064R, A20 for S29V/XS032R.*

## Block Diagram of Simultaneous Operation Circuit

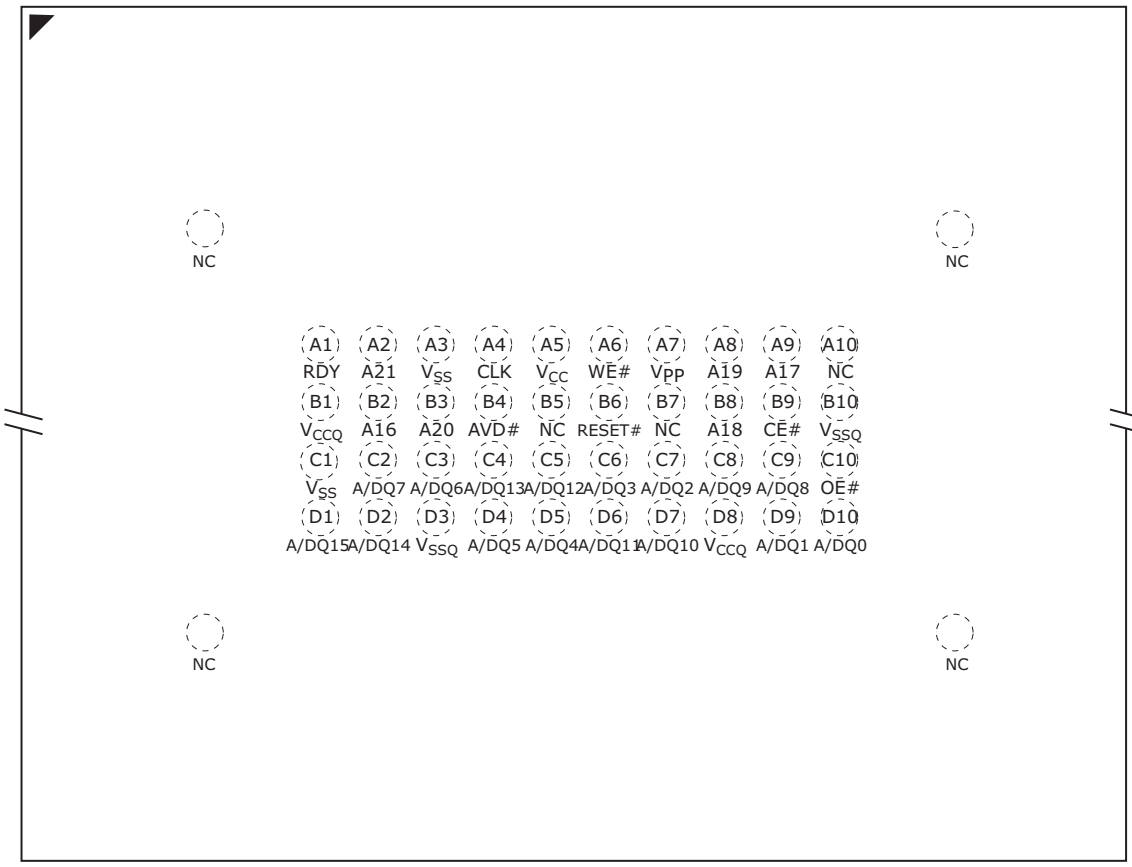



***Notes***

1. *A15–A0 are multiplexed with DQ15–DQ0.*
2. *Amax indicates the highest order address bit.*
3. *n = 3.*

# 3. Connection Diagram

**S29VS/XS-64R**
**44-Ball Very Thin FBGA**
**Top View, Balls Facing Down**

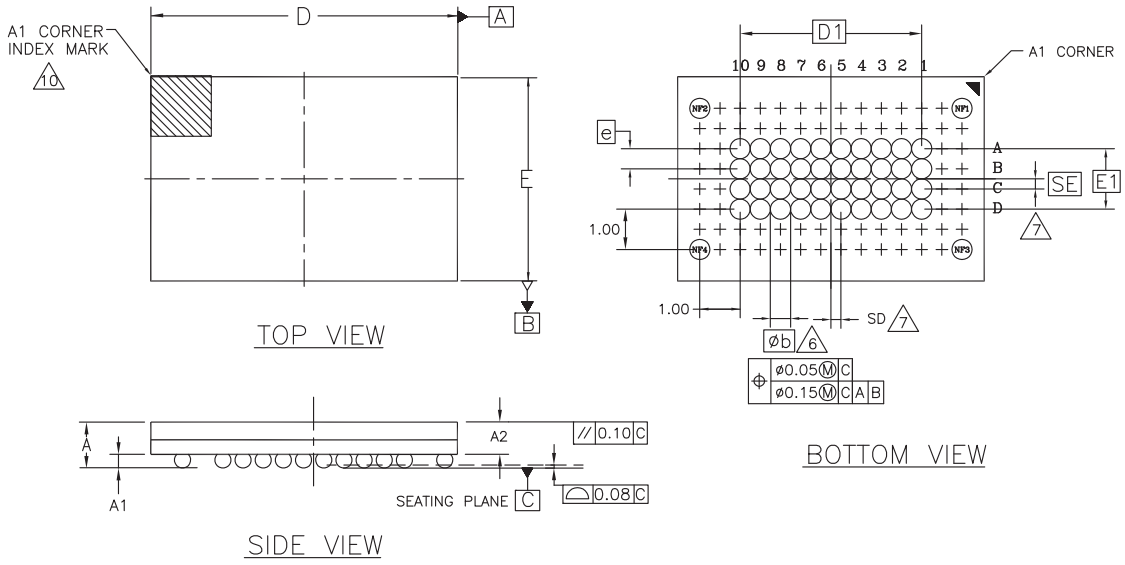



### Special Package Handling Instructions

Special handling is required for Flash Memory products in FBGA packages.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.

# 4. Physical Dimensions

## VDL044—44-Ball Very Thin Fine-Pitch Ball Grid Array (FBGA) 7.5 x5.0 mm Package




| PACKAGE | VDL 044 | | | NOTE |
|---|---|---|---|---|
| JEDEC | N/A | | | |
| D x E | 7.50 mm x 5.00 mm PACKAGE | | | |
| SYMBOL | MIN | NOM | MAX | |
| A | --- | --- | 1.00 | PROFILE |
| A1 | 0.18 | --- | --- | BALL HEIGHT |
| A2 | 0.62 | --- | 0.74 | BODY THICKNESS |
| D | 7.50 BSC | | | BODY SIZE |
| E | 5.00 BSC | | | BODY SIZE |
| D1 | 4.50 BSC | | | MATRIX FOOTPRINT |
| E1 | 1.50 BSC | | | MATRIX FOOTPRINT |
| MD | 10 | | | MATRIX SIZE D DIRECTION |
| ME | 4 | | | MATRIX SIZE E DIRECTION |
| n | 44 | | | BALL COUNT |
| Øb | 0.25 | 0.30 | 0.35 | BALL DIAMETER |
| e | 0.50 BSC | | | BALL PITCH |
| SE / SD | 0.25 BSC | | | BALL PITCH |
| | NONE | | | DEPOPULATED SOLDER BALLS |
| | | | | |

NOTES:

1. DIMENSIONING AND TOLERANCING METHODS 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 MATRIX SIZE IN THE "D" DIRECTION.

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

   n IS THE 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 SD OR SE = 0.000.

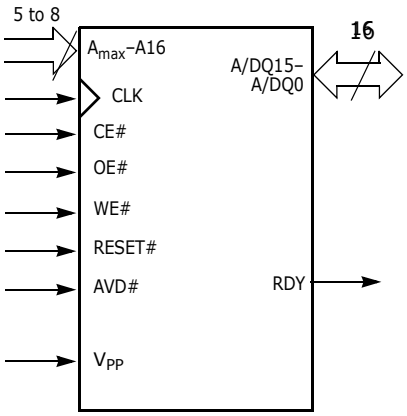
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = e/2
8. NOT USED.
9. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
10. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

3697 \ f16-038.27 \ 6.17.8

# 5. Input/Output Descriptions

| Signal | Description |
|---|---|
| A21-A16 | Address Inputs (A21 is for 64 Mb only) |
| A/DQ15–A/DQ0 | Multiplexed Address/Data input/output |
| CE# | Chip Enable Input. Asynchronous relative to CLK for the Burst mode. |
| OE# | Output Enable Input. Asynchronous relative to CLK for the Burst mode. |
| WE# | Write Enable Input. |
| $V_{CC}$ | Device Power Supply (1.70 V–1.95 V). |
| $V_{CCQ}$ | Input/Output Power Supply (1.70 V–1.95 V). |
| $V_{SS}$ | Ground |
| $V_{SSQ}$ | Input/Output Ground |
| NC | No Connect; not connected internally |
| RDY | Ready output; indicates the status of the Burst read. $V_{OL}$= data invalid. $V_{OH}$ = data valid. |
| CLK | The first rising edge of CLK in conjunction with AVD# low latches address input and activates burst mode operation. After the initial word is output, subsequent rising edges of CLK increment the internal address counter. CLK should remain low during asynchronous access. |
| AVD# | Address Valid input. Indicates to device that the valid address is present on the address inputs (address bits A15–A0 are multiplexed, address bits Amax–A16 are address only).<br>$V_{IL}$ = for asynchronous mode, indicates valid address; for burst mode, causes starting address to be latched on rising edge of CLK.<br>$V_{IH}$= device ignores address inputs |
| RESET# | Hardware reset input. $V_{IL}$= device resets and returns to reading array data |
| $V_{PP}$ | At 9 V, accelerates programming. At $V_{IL}$, disables program and erase functions. Should be at $V_{IH}$ for all other conditions. |

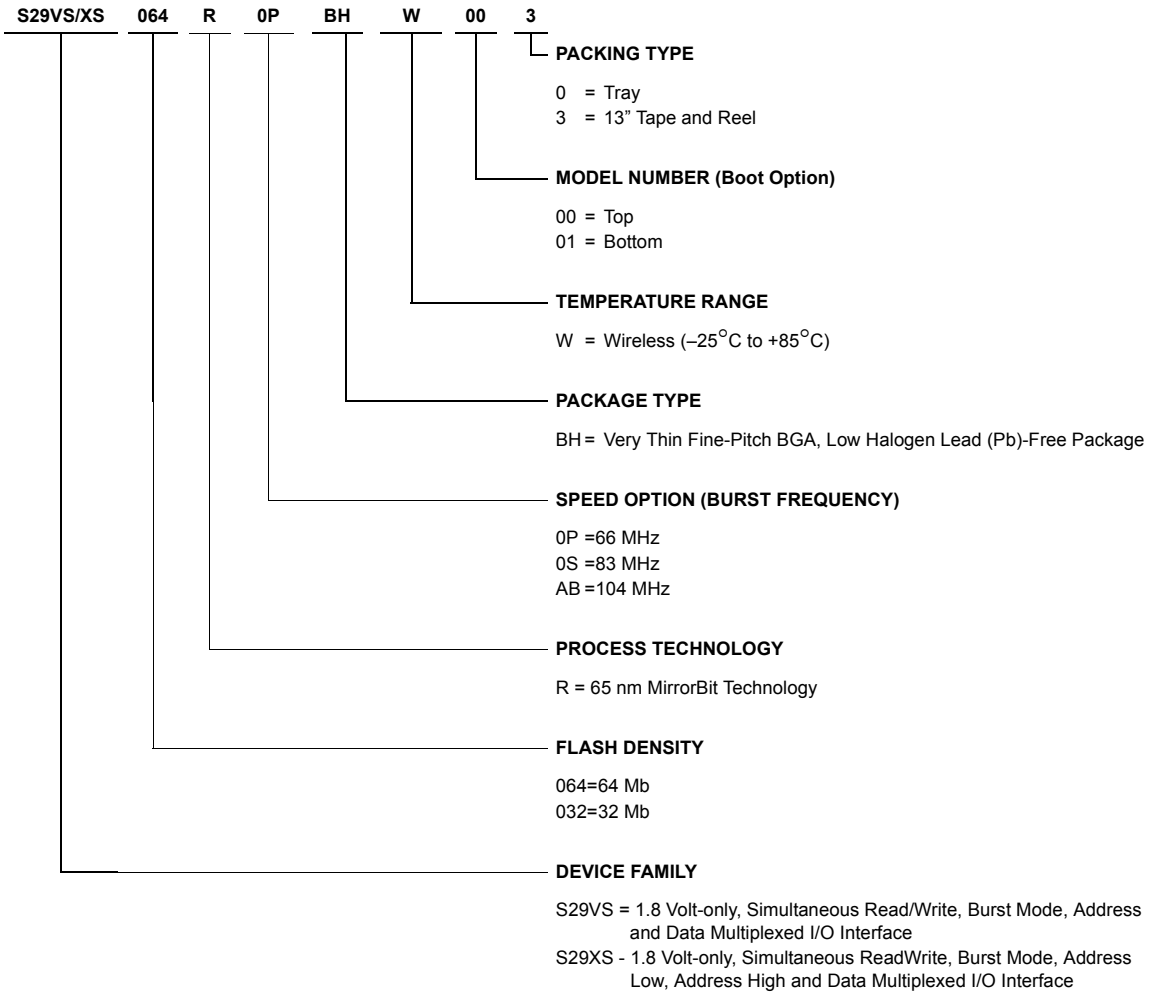
# 6. Logic Symbol




$A_{max}$ indicates the highest order address bit.

# 7. Ordering Information

The order number (Valid Combination) is formed by the following:




## 7.1 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.

| S29V/XS-R Valid Combinations (1)(2) | | | | | |
|---|---|---|---|---|---|
| **Base Ordering Part Number** | **Speed Option** | **Package Type, Material, and Temperature** | **Packing Type** | **Model Numbers** | **Package Type (2)** |
| S29VS064R | 0P, 0S, AB | BHW Low Halgen Lead (Pb)-Free | 0, 3 (1) | 00, 01 | 5.0 mm x 7.5 mm, 44-ball |
| S29VS032R | | | | | |
| S29XS064R | | | | | |
| S29XS032R | | | | | |

***Notes***

1. *Type 0 is standard. Specify other options as required.*
2. *BGA package marking omits leading “S” and packing type designator from ordering part number.*

# 8. Address/Data Configuration (Interface) Modes

There are two options for connection to the address and data buses.

- Address and Data Multiplexed (ADM) mode - On the S29VS-R devices upper address is supplied on separate signal inputs and the lower 16-bits of address are multiplexed with 16-bit data on the A/DQ15 to A/DQ0 I/Os.
- Address-high, Address-low, and Data Multiplexed (AADM) mode - On the S29XS-R devices upper and lower address are multiplexed with 16-bit data on the A/DQ15 to A/D0 signal I/Os.

The two options allow use with the traditional address/data multiplexed NOR interface (S29VS family), or an address multiplexed/ data multiplexed interface with the lowest signal count (S29XS family).

ADM or AADM mode can be selected via ordering part number only.

## 8.1 ADM Interface Mode (S29VS064R, S29VS032R)

In ADM mode, the AVD# signal is used to capture the entire address with a single toggle of AVD# in asynchronous mode or in a single clock cycle in synchronous mode.

## 8.2 AADM Interface Mode (S29XS064R, S29XS032R)

Signal input and output (I/O) connections on a high complexity component such as an Application Specific Integrated Circuit (ASIC) are a limited resource. Reducing signal count on any interface of the ASIC allows for either more features or lower package cost. The memory interface described in this section is intended to reduce the I/O signal count associated with the Flash memory interface with an ASIC.

The interface is called Address-High, Address-Low, and Data Multiplexed (AADM) because all address and data information is time multiplexed on a single 16-bit wide bus. This interface is electrically compatible with existing ADM 16-bit wide random access static memory interfaces but uses fewer address signals. In that sense AADM is a signal count subset of existing static memory interfaces. This interface can be implemented in existing memory controller designs, as an additional mode, with minimal changes. No new I/O technology is needed and existing memory interfaces can continue to be supported while the electronics industry adopts this new interface. ASIC designers can reuse the existing memory address signals above A15 for other functions when an AADM memory is in use.

By breaking up the memory address in to two time slots the address is naturally extended to be a 32-bit word address. But, using two bus cycles to transfer the address increases initial access latency by increasing the time address is using the bus. However, many memory accesses are to locations in memory nearby the previous access. Very often it is not necessary to provide both cycles of address. This interface stores the high half of address in the memory so that if the high half of address does not change from the previous access, only the low half of address needs to be sent on the bus. If a new upper address is not captured at the beginning of an access the last captured value of the upper address is used. This allows accesses within the same 128-KByte address range to provide only the lower address as part of each access.

In AADM mode two signal rising edges are needed to capture the upper and lower address portions in asynchronous mode or two signal combinations over two clocks is needed in synchronous mode. In asynchronous mode the upper address is captured by an AVD# rising edge when OE# is Low; the lower address is captured on the rising edge of AVD# with OE# High. In synchronous mode the upper address is captured at the rising clock edge when AVD# and OE# are Low; the lower address is captured at the rising edge of clock when AVD# is Low and OE# is High.

CE# going High at any time during the access or OE# returning High after RDY is first asserted High during an access, terminates the read access and causes the address/data bus direction to switch back to input mode. The address/data bus direction switches from input to output mode only after an Address-Low capture when AVD# is Low and OE# is High. This prevents the assertion of OE# during Address-High capture from causing a bus conflict between the host address and memory data signals. Note, in burst mode, this implies at least one cycle of CE# or OE# High before an Address-high for a new access may be placed on the bus so that there is time for the memory to recognize the end of the previous access, stop driving data outputs, and ignore OE# so that assertion of OE# with the new Address-high does not create a bus conflict with a new address being driven on the bus. At high bus frequencies more than one cycle may be need in order to allow time for data outputs to stop driving and new address to be driven (bus turn around time).

During a write access, the address/data bus direction is always in the input mode.

The upper address is set to Zero or all Ones, for bottom or top boot respectively, during a Hardware Reset, operate in ADM mode during the early phase of boot code execution where only a single address cycle would be issued with the lower 16 bit of the address reaching the memory in AADM mode. The default high order address bits will direct the early boot accesses to the 128 Kbytes at the boot end of the device. Note that in AADM interface mode this effectively requires that one of the boot sectors is selected for any address overlay mode because in the initial phase of AADM mode operation the host memory controller may only issue the low order address thus limiting the early boot time address space to the 128 Kbytes at the boot end of the device.

# 9. 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 composed of latches that 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 9.1 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 9.1** Device Bus Operations

| Operation | CE# | OE# | WE# | $A_{max}$–16 | A/DQ15–0 | RESET# | CLK | AVD# |
|---|---|---|---|---|---|---|---|---|
| Standby (CE#) | H | X | X | X | HIGH Z | H | H/L | X |
| Hardware Reset | X | X | X | X | HIGH Z | L | X | X |
| Asynchronous Address Latch<br>- ADM mode (29VS064R and S29VS032R only) | L | H | X | Addr In | Addr In | H | X | ↑ |
| Asynchronous Upper Address Latch<br>- AADM mode (29XS064R and S29XS032R only) | L | L | H | X | Addr In | H | X | ↑ |
| Asynchronous Lower Address Latch<br>- AADM mode (29XS064R and S29XS032R only) | L | H | X | X | Addr In | H | X | ↑ |
| Asynchronous Read | L | L | H | Addr In | I/O | H | L | ⊓ |
| Write | L | H | L | Addr In | I/O | H | H/L | ⊓ |
| **Burst Read Operations** | | | | | | | | |
| Latch Starting Burst Address by CLK<br>-ADM mode (29VS064R and S29VS032R only) | L | H | H | Addr In | Addr In | H | ↑ | L |
| Latch Upper Starting Burst Address by CLK -AADM mode (29XS064R and S29XS032R only) | L | L | H | X | Addr In | H | ↑ | L |
| Latch Lower Starting Burst Address by CLK -AADM mode (29XS064R and S29XS032R only) | L | H | H | X | Addr In | H | ↑ | L |
| Advance Burst to next address with appropriate Data presented on the Data Bus | L | L | H | X | Burst Data Out | H | ↑ | H |
| Terminate current Burst read cycle | H | X | H | X | HIGH Z | H | ↑ | X |
| Terminate current Burst read cycle via RESET# | X | X | H | X | HIGH Z | L | X | X |
| Terminate current Burst read cycle and start new Burst read cycle | L | H | H | X | I/O | H | ↑ | ⊓ |

***Legend***
*L = Logic 0, H = Logic 1, X = Don't Care.*

## 9.1 VersatileIO™ ($V_{IO}$) Control

The VersatileIO ($V_{IO}$) control allows the host system to set the voltage levels that the device generates at its data outputs and the voltages tolerated at its data inputs to the same voltage level that is asserted on the $V_{CCQ}$ pin.

## 9.2 Asynchronous Read

The device is in the Asynchronous mode when Bit 15 of the Configuration register is set to '1'. To read data from the memory array, the system must first assert a valid address.

### 9.2.1 S29VS-R ADM Access

With CE# LOW, WE# HIGH, and OE# HIGH, the system presents the address to the device and sets AVD# LOW. AVD# is kept LOW for at least $t_{AVDP}$ ns. The address is latched on the rising edge of AVD#.

### 9.2.2 S29XS-R AADM Access

With CE# LOW, WE# HIGH, and OE# HIGH, the system presents the upper address bits to DQ and sets AVD# LOW. The system then sets OE# LOW. The upper address bits are set when AVD# goes HIGH.

The system then sets AVD# LOW again, with OE# HIGH to capture the lower address bits. The lower address bits are latched on the next rising edge of AVD#.

Address access time ($t_{ACC}$) is equal to the delay from stable addresses to valid output data. The chip enable access time ($t_{CE}$) is the delay from stable CE# to valid data at the outputs. See *AC Characteristics* on page 49.

## 9.3 Synchronous (Burst) Read Mode and Configuration Register

The device is capable of continuous sequential burst operation and linear burst operation of a preset length.

When the device first powers up, it is enabled for asynchronous read operation. The device is automatically enabled for burst mode and addresses are latched on the first rising edge of CLK input, while AVD# is held low for one clock cycle. A trim bit option, when is set, requires configuration register bit 15 set to 1 in order to use Asynchronous Read Mode.

Prior to entering burst mode, the system should determine how many wait states are needed for the initial word of each burst access (see table below), what mode of burst operation is desired, how the RDY signal transitions with valid data, and output drive strength. The system would then write the configuration register command sequence. See *Configuration Register* on page 31 for further details.

When the appropriate number of Wait States have occurred, data is output after the rising edge of the CLK. Subsequent words are output $t_{BACC}$ after the rising edge of each successive clock cycle, which automatically increments the internal address counter. RDY indicates the initial latency and any subsequent waits.

### 9.3.1 S29VS-R ADM Access

To burst read data from the memory array in ADM mode, the system must assert CE# to $V_{IL}$, and provide a valid address while driving AVD# to $V_{IL}$ for one cycle. OE# must remain at $V_{IH}$ during the one cycle that AVD# is low. The data appears on A/DQ15 -A/DQ0 when CE# remains Low, after OE# is Low and the synchronous access times are satisfied. The next data in the burst sequence is read on each clock cycle that OE# and CE# remain Low.

OE# does not terminate a burst access if it rises to $V_{IH}$ during a burst access. The outputs will go to high impedance but the burst access will continue until terminated by CE# going to $V_{IH}$, or AVD# returns to $V_{IL}$ with a new address to initiate a another burst access.

### 9.3.2 S29XS-R AADM Access

To burst read data from the memory array in AADM mode, the system must assert CE# to $V_{IL}$, OE# must go low with AVD# for one cycle while the upper address is valid. The rising edge of CLK when OE# and AVD# are Low captures the upper 16 bits of address. The rising edge of CLK when OE# is High and AVD# is Low latches the lower 16 bits of address. The data appears on A/DQ15 -A/DQ0 when CE# remains Low, after OE# is Low and the synchronous access times are satisfied. The next data in the burst sequence is read on each clock cycle that OE# and CE# remain Low.

Once OE# returns to $V_{IH}$ during a burst read the OE# no longer enables the outputs until after AVD# is at $V_{IL}$ with OE# at $V_{IH}$ - which signals that address-low has been captured for the next burst access. This is so that OE# at $V_{IL}$ may be used in conjunction with AVD# at $V_{IL}$ to indicate address-high on the A/DQ signals without enabling the A/DQ outputs, thus avoiding data output contention with Address-high.

The device has a fixed internal address boundary that occurs every 128 words. A boundary crossing of one or two additional wait states is required. The time the device is outputting data with the starting burst address not divisible by eight, additional waits might be required.

The following Tables show the latency for variable wait state operation (note that ws = wait state).

**Table 9.2** Wait State vs. Frequency

| Wait State | Frequency (Maximum MHz) |
|---|---|
| 3 | 27 |
| 4 | 40 |
| 5 | 54 |
| 6 | 66 |
| 7 | 80 |
| 8 | 95 |
| 9 | 104 |
| 10 | 104 |

**Table 9.3** Address Latency for 10 -13 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10 -13 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | +2 ws | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | +2 ws | D8 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | +2 ws | D8 |
| 3 | | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | +2 ws | D8 |
| 4 | | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | +2 ws | D8 |
| 5 | | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | +2 ws | D8 |
| 6 | | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | +2 ws | D8 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | +2 ws | D8 |

**Table 9.4** Address Latency for 9 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 9 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | +1 ws | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | +1 ws | D8 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | +1 ws | D8 |
| 3 | | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | +1 ws | D8 |
| 4 | | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | +1 ws | D8 |
| 5 | | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | +1 ws | D8 |
| 6 | | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | +1 ws | D8 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | +1 ws | D8 |

**Table 9.5** Address Latency for 8 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 8 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | D8 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | D8 |
| 3 | | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | D8 |
| 4 | | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | D8 |
| 5 | | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | D8 |
| 6 | | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | D8 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | D8 |

**Table 9.6** Address Latency for 7 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | 1 ws | D8 | D9 |
| 3 | | D3 | D4 | D5 | D6 | D7 | 1 ws | 1 ws | D8 | D9 |
| 4 | | D4 | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | D8 | D9 |
| 5 | | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | D8 | D9 |
| 6 | | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | D8 | D9 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | D8 | D9 |

**Table 9.7** Address Latency for 6 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 |
| 3 | | D3 | D4 | D5 | D6 | D7 | 1 ws | D8 | D9 | D10 |
| 4 | | D4 | D5 | D6 | D7 | 1 ws | 1 ws | D8 | D9 | D10 |
| 5 | | D5 | D6 | D7 | 1 ws | 1 ws | 1 ws | D8 | D9 | D10 |
| 6 | | D6 | D7 | 1 ws | 1 ws | 1 ws | 1 ws | D8 | D9 | D10 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | 1 ws | 1 ws | D8 | D9 | D10 |

**Table 9.8** Address Latency for 5 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 5 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 |
| 3 | | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 |
| 4 | | D4 | D5 | D6 | D7 | 1 ws | D8 | D9 | D10 | D11 |
| 5 | | D5 | D6 | D7 | 1 ws | 1 ws | D8 | D9 | D10 | D11 |
| 6 | | D6 | D7 | 1 ws | 1 ws | 1 ws | D8 | D9 | D10 | D11 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | 1 ws | D8 | D9 | D10 | D11 |

**Table 9.9** Address Latency for 4 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 |
| 3 | | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 |
| 4 | | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
| 5 | | D5 | D6 | D7 | 1 ws | D8 | D9 | D10 | D11 | D12 |
| 6 | | D6 | D7 | 1 ws | 1 ws | D8 | D9 | D10 | D11 | D12 |
| 7 | | D7 | 1 ws | 1 ws | 1 ws | D8 | D9 | D10 | D11 | D12 |

**Table 9.10** Address Latency for 3 Wait States

| Word | Initial Wait | Subsequent Clock Cycles After Initial Wait States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3 wait states | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
| 1 | | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 |
| 2 | | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 |
| 3 | | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 |
| 4 | | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
| 5 | | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 |
| 6 | | D6 | D7 | 1 ws | D8 | D9 | D10 | D11 | D12 | D13 |
| 7 | | D7 | 1 ws | 1 ws | D8 | D9 | D10 | D11 | D12 | D13 |

The device will continue to output continuous, sequential burst data, wrapping around to address 000000h after it reaches the highest addressable memory location, until the system asserts CE# high, RESET# low, or AVD# low in conjunction with a new address. See Table 9.1 on page 13. The reset command does *not* terminate the burst read operation.

### 8- and 16-Word Linear Burst with Wrap Around

These two modes are of the linear wrap around design, in which a fixed number of words are read from consecutive addresses. In each of these modes, the burst addresses read are determined by the group within which the starting address falls. The groups are sized according to the number of words read in a single burst sequence for a given mode (see Table 9.11.)

**Table 9.11** Burst Address Groups

| Mode | Group Size | Group Address Ranges |
|---|---|---|
| 8-word | 8 words | 0-7h, 8-Fh, 10-17h, 18-1Fh... |
| 16-word | 16 words | 0-Fh, 10-1Fh, 20-2Fh, 30-3Fh... |

As an example: if the starting address in the 8-word mode is 3Ah, and the burst sequence would be 3A-3B-3C-3D-3E-3F-38-39h. The burst sequence begins with the starting address written to the device, but wraps back to the first address in the selected group. In a similar fashion, the 16-word Linear Wrap mode begins its burst sequence on the starting address written to the device, and then wraps back to the first address in the selected address group and terminates the burst read. **Note that in these two burst read modes the address pointer does not cross the boundary that occurs every 128 words; thus, no wait states are inserted (except during the initial access).**

## 9.4 Programmable Wait State

The programmable wait state feature indicates to the device the number of additional clock cycles that must elapse after AVD# is driven active before data will be available. Upon power up, the device defaults to the maximum of seven total cycles. The total number of wait states is programmable from three to nine cycles. For further details, see *Set Configuration Register Command Sequence* on page 28.

## 9.5 Configuration Register

The device uses a configuration register to set the various burst parameters: number of wait states, burst read mode, burst length, RDY configuration, and synchronous mode active.

## 9.6 Handshaking Feature

The handshaking feature allows the host system to simply monitor the RDY signal from the device to determine when the initial word of burst data is ready to be read. The host system should use the configuration register to set the number of wait states for optimal burst mode operation. The initial word of burst data is indicated by the rising edge of RDY after OE# goes low.

## 9.7 Simultaneous Read/Write Operations with Zero Latency

This device is capable of reading data from one bank of memory while programming or erasing in one of the other banks of memory. An erase operation may also be suspended to read from or program to another location within the same bank (except the sector being erased). Figure 19.12, on page 58 shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to the *DC Characteristics* on page 47 table for read-while-program and read-while-erase current specifications.

## 9.8 Writing Commands/Command Sequences

The device has inputs/outputs that accept both address and data information. To write a command or command sequence (which includes programming data to the device and erasing sectors of memory), the system must drive AVD# and CE# to $V_{IL}$, and OE# to $V_{IH}$ when providing an address to the device, and drive WE# and CE# to $V_{IL}$, and OE# to $V_{IH}$. when writing commands or data.

An erase operation can erase one sector, multiple sectors, or the entire device. Table 14-17 indicates the address space that each sector occupies. The device address space is divided into multiple banks. A “bank address” is the address bits required to uniquely select a bank. Similarly, a “sector address” is the address bits required to uniquely select a sector.

Refer to the DC Characteristics table for write mode current specifications. The *AC Characteristics* on page 49 section contains timing specification tables and timing diagrams for write operations.

## 9.9 Accelerated Program and Erase Operations

The device offers accelerated program and erase operation through the $V_{PP}$ function. $V_{PP}$ is primarily intended to allow faster manufacturing throughput at the factory and not to be used in system operations.

If the system asserts $V_{HH}$ on this input, the device uses the higher voltage on the input to reduce the time required for program and erase operations. Removing $V_{HH}$ from the $V_{PP}$ input, upon completion of the embedded program or erase operation, returns the device to normal operation. Note that sectors must be unlocked prior to raising $V_{PP}$ to $V_{HH}$. *Note that the $V_{PP}$ pin must not be at $V_{HH}$ for operations other than accelerated programming, or device damage may result. In addition, the $V_{PP}$ pin must not be left floating or unconnected; inconsistent behavior of the device may result*.

When at $V_{IL}$, $V_{PP}$ locks all sectors. $V_{PP}$ should be at $V_{IH}$ for all other conditions.

## 9.10 Write Buffer Programming Operation

**Write Buffer Programming** allows the system to write a maximum of **32** words in one programming operation. This results in a faster effective word programming time than the standard "word" 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 will occur. At this point, the system writes the number of "**word locations minus 1**" that will be loaded into the page buffer at the Sector Address in which programming will occur. This tells the device how many write buffer addresses will be loaded with data and therefore when to expect the "Program Buffer to Flash" confirm command. The number of locations to program cannot exceed the size of the write buffer or the operation will abort. (NOTE: The number loaded = the number of locations to program minus 1. For example, if the system will program 6 address locations, then 05h should be written to the device.)

The system then writes the starting address/data combination. This starting address is the first address/data pair to be programmed, and selects the "write-buffer-page" address. All subsequent address/data pairs **must** fall within the "selected-write-buffer-page", and be loaded in sequential order.

The "write-buffer-page" is selected by using the addresses $A_{MAX}$-A5 where $A_{MAX}$ is A21 for S29VS/XS064R.

The "write-buffer-page" addresses **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 will ABORT.)

After writing the Starting Address/Data pair, the system then writes the remaining address/data pairs into the write buffer. Write buffer locations must be loaded in sequential order.

Note that if a Write Buffer address location is loaded multiple times, the "address/data pair" counter **will be decremented for every data load operation**. Also, the **last data loaded** at a location before the "Program Buffer to Flash" confirm command will be programmed into the device. It is the software's responsibility to comprehend ramifications of loading a write-buffer location more than once. The counter decrements **for each data load operation**, **NOT for each unique write-buffer-address location**.

Once the specified number of write buffer locations have been loaded, the system must then write the "Program Buffer to Flash" command at the Sector Address. Any other address/data write combinations will abort the Write Buffer Programming operation. The device will then "go busy". The Data Bar polling techniques should be used while monitoring the **last address location loaded into the write buffer**. This eliminates the need to store an address in memory because the system can load the last address location, issue the program confirm command at the last loaded address location, and then data bar poll at that same address. DQ7, DQ6, DQ5, DQ2, and DQ1 should be monitored to determine the device status during Write Buffer Programming.

The write-buffer "embedded" programming operation can be suspended using the standard suspend/resume commands. Upon successful completion of the Write Buffer Programming operation, the device will return to READ mode.

The Write Buffer Programming Sequence can be ABORTED under any of the following conditions:

- 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, DQ5 = 0. This indicates that the Write Buffer Programming Operation was ABORTED. **Note: The Secured Silicon sector, autoselect, and CFI functions are unavailable when a program operation is in progress**.

**Use of the write buffer is strongly recommended for programming when multiple words are to be programmed**. Write buffer programming is allowed in any sequence of memory (or address) locations. These flash devices are capable of handling multiple write buffer programming operations on the same write buffer address range without intervening erases. However, programming the same word address multiple times without intervening erases requires a modified programming method. Please contact your local Spansion™ representative for details.

## 9.11 Autoselect Mode

The autoselect mode provides manufacturer and device identification, and sector protection verification, through identifier codes output from the internal register (which is separate from the memory array) on DQ15–DQ0. This mode is primarily intended for programming equipment to automatically match a device to be programmed with its corresponding programming algorithm. The autoselect codes can also be accessed in-system.

When verifying sector protection, the sector address must appear on the appropriate highest order address bits. The remaining address bits are don't care. When all necessary bits have been set as required, the programming equipment may then read the corresponding identifier code on DQ15–DQ0. The autoselect codes can also be accessed in-system through the command register. The command sequence is illustrated in Table 12.3 on page 37. *Note that if a Bank Address (BA) on address bits A21, A20, and A19 for the*
*VS/XS064R are asserted during the third write cycle of the autoselect command, the host system can read autoselect data that bank and then immediately read array data from the other bank, without exiting the autoselect mode.*

To access the autoselect codes, the host system must issue the autoselect command via the command register, as shown in Table 12.3 on page 37.

## 9.12 Sector Protection

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

### Dynamic Protection Bit (DYB)

DYB is a security feature used to protect individual sectors from being programmed or erased inadvertently. It is a volatile protection bit and is assigned to each sector. Upon power-up, the contents of all DYBs are set (programmed to "0"). Each DYB can be individually modified through the DYB Set Command or the DYB Clear Command.

The Protection Status for a particular sector is determined by the status of the DYB relative to that sector. By issuing the DYB Set or Clear command sequences, the DYBs will be set (programmed to "0") or cleared (erased to "1"), thus placing each sector in the protected or unprotected state respectively. These states are the so-called Dynamic Locked or Unlocked states due to the fact that they can switch back and forth between the protected and unprotected states. This feature allows software to easily protect sectors against inadvertent changes yet does not prevent the easy removal of protection when changes are needed. The DYBs maybe set (programmed to "0") or cleared (erased to "1") as often as needed.

When the parts are first shipped, upon power up or reset, the DYBs are set (programmed to "0").

Note:Dynamic protection bits revert back to their default values after programming device's "Lock Register."

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 or Clear command for the dynamic sectors signify protected or unprotected state of the sectors respectively.

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 and returns to read mode without having modified the contents of the protected sector.

The programming of the DYB for a given sector can be verified by writing individual status read commands DYB Status.

## 9.13 Hardware Data Protection Mode

The device offers one type of data protection at the sector level:

- When $V_{PP}$ is at $V_{IL}$, all sectors are locked

## 9.14 Low VCC 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 reading array data. Subsequent writes are ignored until $V_{CC}$ is greater than $V_{LKO}$. The system must provide the proper signals to the control inputs to prevent unintentional writes when VCC is greater than VLKO.

## 9.15 Write Pulse “Glitch” Protection

Noise pulses of less than $t_{WEP}$ on WE# do not initiate a write cycle.

## 9.16 Logical Inhibit

Write cycles are inhibited by holding any one of OE# = VIL, CE# = VIH or WE# = VIH. To initiate a write cycle, CE# and WE# must be a logical zero while OE# is a logical one.

### Power-Up Write Inhibit

If WE# = CE# = RESET# = $V_{IL}$ 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

## 9.17 Lock Register

The Lock Register consists of one bit. This bit is non-volatile and read-only. DQ15-DQ1 are reserved and are undefined.

**Table 9.12** Lock Register

| DQ15-1 | DQ0 |
|---|---|
| Undefined | Secured Silicon Sector Protection Bit |

***Note***
*When the device lock register is programmed (the Secured Silicon lock bit is programmed) all DYBs revert to the power-on default state.*

## 9.18 Standby Mode

When the system is not reading or writing to the device, it can place the device in the 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# inputs are both held at $V_{CC}$. The device requires standard access time ($t_{CE}$) 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.

$I_{CC3}$ in the *DC Characteristics* on page 47 table represents the standby current specification.

## 9.19 Automatic Sleep Mode

The automatic sleep mode minimizes Flash device energy consumption. The device automatically enters this mode when addresses and clock remain stable for $t_{ACC}$ + 20 ns. The automatic sleep mode is independent of the CE#, WE#, and OE# control signals. Standard address access timings provide new data when addresses are changed. While in sleep mode, output data is latched and always available to the system. $I_{CC4}$ in the *DC Characteristics* on page 47 table represents the automatic sleep mode current specification.

## 9.20 RESET#: Hardware Reset Input

The RESET# input provides a hardware method of resetting the device to reading array data. When RESET# is driven low for at least a period of $t_{RP}$, the device immediately terminates any operation in progress, tristates all outputs, and ignores all read/write commands for the duration of the RESET# pulse. The device also resets the internal state machine to reading array data. The operation that was 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 $V_{SS}$, the device draws CMOS standby current ($I_{CC4}$). If RESET# is held at $V_{IL}$ but not within $V_{SS}$, the standby current will be greater.

RESET# 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 49 tables for RESET# parameters and to Figure 19.5 on page 53 for the timing diagram.

### $V_{CC}$ Power-up and Power-down Sequencing

The device imposes no restrictions on $V_{CC}$ power-up or power-down sequencing. Asserting RESET# to $V_{IL}$ is required during the entire $V_{CC}$ power sequence until the respective supplies reach their operating voltages. Once $V_{CC}$ attains its operating voltage, de-assertion of RESET# to $V_{IH}$ is permitted.

## 9.21 Output Disable Mode

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

## 9.22 Secured Silicon Sector Flash Memory Region

The Secured Silicon Sector feature provides a Flash memory region that enables permanent part identification through an Electronic Serial Number (ESN). The Secured Silicon Sector is 256 words in length. All reads outside of the 256 word address range will return non-valid data. The Factory Indicator Bit (DQ7) is used to indicate whether or not the Factory Secured Silicon Sector is locked when shipped from the factory. The Customer Indicator Bit (DQ6) is used to indicate whether or not the Customer Secured Silicon Sector is locked when shipped from the factory. The Factory Secured Silicon bits are permanently set at the factory and cannot be changed, which prevents cloning of a factory locked part. This ensures the security of the ESN and customer code once the product is shipped to the field.

Spansion offers the device with a Factory Secured Silicon Sector that is locked and a Customer Secured Silicon Sector that is either locked or is lockable. The Factory Secured Silicon Sector is always protected when shipped from the factory, and has the Factory Indicator Bit (DQ7) permanently set to a “1”. The Customer Secured Silicon Sector is shipped unprotected, allowing customers to utilize that sector in any manner they choose. Once the Customer Secured Silicon Sector area is protected, the Customer Indicator Bit will be permanently set to “1.”

The system accesses the Secured Silicon Sector through a command sequence (see *Enter/Exit Secured Silicon Sector Command Sequence* on page 32). After the system has written the Enter Secured Silicon Sector command sequence, it may read the Secured Silicon Sector by using the addresses normally occupied by sector SA0 of the memory array. This mode of operation continues until the system issues the Exit Secured Silicon Sector command sequence, or until power is removed from the device. While Secured Silicon Sector access is enabled, Memory Array read access, program operations, and erase operations to all sectors other than SA0 are also available. On power-up, or following a hardware reset, the device reverts to sending commands to the normal address space.

### Factory Locked: Factor Secured Silicon Sector Programmed and Protected At the Factory

In a factory sector locked device, the Factory Secured Silicon Sector is protected when the device is shipped from the factory. The Factory Secured Silicon Sector cannot be modified in any way. The device is pre programmed with both a random number and a secure ESN. The Factory Secured Silicon Sector is located at addresses 000000h–00007Fh.

The device is available pre programmed with one of the following:

- A random, secure ESN only within the Factor Secured Silicon Sector
- Customer code within the Customer Secured Silicon Sector through the Spansion programming services
- Both a random, secure ESN and customer code through the Spansion programming services.

**Table 9.13** Secured Silicon Sector Addresses

| Sector | Sector Size | Address Range |
|---|---|---|
| Customer | 128 words | 000080h-0000FFh |
| Factory | 128 words | 000000h-00007Fh |

Customers may opt to have their code programmed by Spansion through the Spansion™ programming services. Spansion programs the customer’s code, with or without the random ESN. The devices are then shipped from Spansion’s factory with the Factory Secured Silicon Sector and Customer Secured Silicon Sector permanently locked. Contact an Spansion representative for details on using Spansion’s Spansion™ programming services.

### Customer Secured Silicon Sector

If the security feature is not required, the Customer Secured Silicon Sector can be treated as an additional Flash memory space. The Customer Secured Silicon Sector can be read any number of times, but can be programmed and locked only once. Note that the accelerated programming ($V_{PP}$) function is not available when programming the Customer Secured Silicon Sector, but reading the first Bank through the last bank is available. The Customer Secured Silicon Sector is located at addresses 000080h–0000FFh.

The Customer Secured Silicon Sector area can be protected by writing the Secured Silicon Sector Protection Bit Lock command sequence.

Once the Customer Secured Silicon Sector is locked and verified, the system must write the Exit Secured Silicon Sector Region command sequence to return to reading and writing SA0 in the memory array.

The Customer Secured Silicon Sector lock must be used with caution since, once locked, there is no procedure available for unlocking the Customer Secured Silicon Sector area and none of the bits in the Customer Secured Silicon Sector memory space can be modified in any way.

## 10. 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 Tables Table 10.2–Table 10.5. To terminate reading CFI data, the system must write the reset command.

For further information, please refer to the CFI Specification (see JEDEC publications JEP137-A and JESD68.01). Please contact your sales office for copies of these documents.

**Table 10.1** ID/CFI Data

| | Word Offset Address | DATA | | Description |
|---|---|---|---|---|
| | | VS/XS064R | VS/XS032R | |
| Device Identification | (SA) + 00h | 0001h | | Spansion Manufacturer ID |
| | (SA) + 01h | 007Eh(top/bottom) | 007Eh(top/bottom) | Device ID, Word 1 Extended ID address code. Indicates an extended two byte device ID is located at byte address 1Ch and 1Eh. |
| | (SA) + 02h | 0001h-Locked, 0000h-Unlocked | | |
| | (SA) + 03h | 0000h | | Revision ID |
| | (SA) + 04h | Reserved | | Reserved |
| | (SA) + 05h | Reserved | | Reserved |
| | (SA) + 06h | 0010h | | ID Version |
| | (SA) + 07h | 00BFh | | Indicator Bits:<br>DQ15 - DQ8 = Reserved<br>DQ7 - Factory Lock Bit: 1 = Locked; 0 = Not Locked<br>DQ6 - Customer Lock Bit: 1 = Locked; 0 = Not locked<br>DQ5 - DQ0 = Reserved |
| | (SA) + 08h | Reserved | | Reserved |
| | (SA) + 09h | Reserved | | Reserved |
| | (SA) + 0Ah | Reserved | | Reserved |
| | (SA) + 0Bh | Reserved | | Reserved |
| | (SA) + 0Ch | 0002h | | Lower Software Bits<br>Bit 0 - Status Register Support<br>1 = Status Register Supported<br>0 = Status register not Supported<br>Bit 1 - DQ Polling Support<br>1 = DQ bits polling supported<br>0 = DQ bits polling not supported<br>Bit 3-2 - Command Set Support<br>11 = Reserved<br>10 = Reserved<br>01 = Reduced Command Set<br>00 = Old Command Set<br>Bit 4-F - Reserved |
| | (SA) + 0Dh | Reserved | | Upper Software Bits<br>Reserved |
| | (SA) + 0Eh | 0061h(top/bottom) | 0061h(top/bottom) | High Order Device ID, Word 2 |
| | (SA) + 0Fh | 0001h(top)<br>0002h(bottom) | 0003h(top)<br>0004h(bottom) | Low Order Device ID, Word 3 |

**Table 10.2** CFI Query Identification String

| Addresses | Data | | Description |
|---|---|---|---|
| | VS/XS064R | VS/XS064R | |
| 10h<br>11h<br>12h | 0051h<br>0052h<br>0059h | | Query Unique ASCII string “QRY” |
| 13h<br>14h | 0002h<br>0000h | | Primary OEM Command Set |
| 15h<br>16h | 0040h<br>0000h | | Address for Primary Extended Table |
| 17h<br>18h | 0000h<br>0000h | | Alternate OEM Command Set (00h = none exists) |
| 19h<br>1Ah | 0000h<br>0000h | | Address for Alternate OEM Extended Table (00h = none exists) |

**Table 10.3** System Interface String

| Addresses | Data | | Description |
|---|---|---|---|
| | VS/XS064R | VS/XS032R | |
| 1Bh | 0017h | | $V_{CC}$ Min. (write/erase)<br>D7–D4: volt, D3–D0: 100 millivolt |
| 1Ch | 0019h | | $V_{CC}$ Max. (write/erase)<br>D7–D4: volt, D3–D0: 100 millivolt |
| 1Dh | 0000h | | $V_{PP}$ Min. voltage (00h = no $V_{PP}$ pin present) Refer to 4Dh |
| 1Eh | 0000h | | $V_{PP}$ Max. voltage (00h = no $V_{PP}$ pin present) Refer to 4Eh |
| 1Fh | 0008h | | Typical timeout per single byte/word write $2^N$ µs |
| 20h | 0009h | | Typical timeout for Min. size buffer write $2^N$ µs (00h = not supported) |
| 21h | 000Ah | | Typical timeout per individual block erase $2^N$ ms |
| 22h | 0011h | | Typical timeout for full chip erase $2^N$ ms (00h = not supported) |
| 23h | 0003h | | Max. timeout for byte/word write $2^N$ times typical |
| 24h | 0003h | | Max. timeout for buffer write $2^N$ times typical |
| 25h | 0003h | | Max. timeout per individual block erase $2^N$ times typical |
| 26h | 0003h | | Max. timeout for full chip erase $2^N$ times typical (00h = not supported) |

**Table 10.4** Device Geometry Definition

| Addresses | Data | | Description |
|---|---|---|---|
| | VS/XS064R | VS/XS032R | |
| 27h | 0017h | | Device Size = $2^N$ byte |
| 28h | 0001h | | Flash Device Interface description (refer to CFI publication 100) |
| 29h | 0000h | | |
| 2Ah | 0006h | | Max. number of bytes in multi-byte write = $2^N$ (00h = not supported) |
| 2Bh | 0000h | | |
| 2Ch | 0002h | | Number of Erase Block Regions within device |
| 2Dh | 007Eh (top boot) | 003Eh (top boot) | Erase Block Region 1 Information (refer to the CFI specification or CFI publication 100) |
| | 0003h (bottom boot) | 0003h (bottom boot) | |
| 2Eh | 0070h (top boot) | 0070h (top boot) | |
| | 0000h (bottom boot) | 0000h (bottom boot) | |
| 2Fh | 0000h (top boot) | 0000h (top boot) | |
| | 0040h (bottom boot) | 0040h (bottom boot) | |
| 30h | 0001h (top boot) | 0001h (top boot) | |
| | 0000h (bottom boot) | 0000h (bottom boot) | |
| 31h | 0003h (top boot) | 0003h (top boot) | Erase Block Region 2 Information |
| | 007Eh (bottom boot) | 003Eh (bottom boot) | |
| 32h | 0000h (top boot) | 0000h (top boot) | |
| | 0070h (bottom boot) | 0070h (bottom boot) | |
| 33h | 0040h (top boot) | 0040h (top boot) | |
| | 0000h (bottom boot) | 0000h (bottom boot) | |
| 34h | 0000h (top boot) | 0000h (top boot) | |
| | 0001h (bottom boot) | 0001h (bottom boot) | |
| 35h | 00FFh | | Erase Block Region 3 Information |
| 36h | 00FFh | | |
| 37h | 00FFh | | |
| 38h | 00FFh | | |
| 39h | 00FFh | | Erase Block Region 4 Information |
| 3Ah | 00FFh | | |
| 3Bh | 00FFh | | |
| 3Ch | 00FFh | | |

**Table 10.5** Primary Vendor-Specific Extended Query

| Addresses | Data | | Description |
|---|---|---|---|
| | VS/XS064R | VS/XS032R | |
| 40h | 0050h | | Query-unique ASCII string "PRI" |
| 41h | 0052h | | |
| 42h | 0049h | | |
| 43h | 0031h | | Major version number, ASCII |
| 44h | 0034h | | Minor version number, ASCII |
| 45h | 0020h | | Address Sensitive Unlock (Bits 1-0) 0 = Required, 1 = Not Required Silicon Revision Number (Bits 7-2) |
| 46h | 0002h | | Erase Suspend 0 = Not Supported, 1 = To Rea Only, 2= To Read & Write |
| 47h | 0001h | | Sector Protect 0 = Not Supported, X = Number of sectors in per group |
| 48h | 0000h | | Sector Tempoary Unprotect 00 = Not Supported, 01 = Supported |
| 49h | 0008h | | Sector Protect/Unprotect scheme 08 = Advanced Sector Protection |
| 4Ah | 0020h | | Simultaneous Operation Number of Sectors in all banks except boot bank |
| 4Bh | 0001h | | Burst Mode Type 00 = Not Supported, 01 = Supported |
| 4Ch | 0000h | | Page Mode 00 = Not Supported, 01 = Supported |
| 4Dh | 0085h | | Vpp (Acceleration) Supply Minimum 00h = Not Supported, D7-D4: Volt, D3-D0: 100 mv |
| 4Eh | 0095h | | Vpp (Acceleration) Supply Maximum 00h = Not Supported, D7-D4: Volt, D3-D0: 100 mv |
| 4Fh | 0003h (top boot) | | Top/Bottom Boot Sector Flag 0001h = Top/Middle Boot Device, 0002h = Bottom Boot Device, 03h = Top Boot Device |
| | 0002h (bottom boot) | | |
| 50h | 0001h | | Program Supend. 00h = not supported |
| 51h | 0000h | | Unlock Bypass 00 = Not Supported, 01 = Supported |
| 52h | 0008h | | Secured Silicon Sector (Customer OTP Area) Size $2^N$ bytes |
| 53h | 000Eh | | Hardware Reset Low Time-out during an embedded algorithm to read more mode Maximum $2^N$ ns |
| 54h | 000Eh | | Hardware Reset Low Time-out during an embedded algorithm to read more mode Maximum $2^N$ ns |
| 55h | 0005h | | Erase Suspend Time-out Maximum $2^N$ ns |
| 56h | 0005h | | Program Supend Time-out Maximum $2^N$ ns |
| 57h | 0004h | | Bank Organzation: X = Number of banks |
| 58h | 0020h (top boot) | 0010h (top boot) | Bank 0 Region Information. X = Number of sectors in banks |
| | 0023h (bottom boot) | 0013h (bottom boot) | |
| 59h | 0020h | | Bank 1 Region Information. X = Number of sectors in banks |
| 5Ah | 0020h | | Bank 2 Region Information. X = Number of sectors in banks |
| 5Bh | 0023h (top boot) | 0013h (top boot) | Bank 3 Region Information. X = Number of sectors in banks |
| | 0020h (bottom boot) | 0010h (bottom boot) | |

# 11. Command Definitions

Writing specific address and data commands or sequences into the command register initiates device operations. Table 12.3 on page 37 defines the valid register command sequences. Writing **incorrect address and data values** or writing them in the **improper sequence** resets the device to reading array data.

All addresses are latched on the rising edge of AVD#. All data is latched on the rising edge of WE#. Refer to the *AC Characteristics* on page 49 section for timing diagrams.

## 11.1 Reading Array Data

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

After the device accepts an Erase Suspend command, the corresponding bank 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 the *Erase Suspend/Erase Resume Commands* on page 36 section for more information.

After the device accepts a Program Suspend command, the corresponding bank enters the program-suspend-read mode, after which the system can read data from any non-program-suspended sector within the same bank.

The system *must* issue the reset command to return a bank to the read (or erase-suspend-read) mode if DQ5 goes high during an active program or erase operation, or if the bank is in the autoselect mode.

See also *VersatileIO™ ($V_{IO}$) Control* on page 13 and *Synchronous (Burst) Read Mode and Configuration Register* on page 14 in the *Device Bus Operations* section for more information. The *Asynchronous Read* and *Synchronous/Burst Read* tables provide the read parameters, and Figure 19.3, on page 51 and Figure 19.4, on page 52 show the timings.

## 11.2 Set Configuration Register Command Sequence

The device uses a configuration register to set the various burst parameters: number of wait states, burst read mode, RDY configuration, and synchronous mode active. The configuration register must be set before the device will enter burst mode.

The configuration register is loaded with a four-cycle command sequence. The first two cycles are standard unlock sequences. On the third cycle, the data should be D0h and address bits should be 555h. During the fourth cycle, the configuration code should be entered onto the data bus with the address bus set to address 000h. Once the data has been programmed into the configuration register, a software reset command is required to set the device into the correct state. The device will power up or after a hardware reset with the default setting, which is in asynchronous mode. The register must be set before the device can enter synchronous mode. The configuration register can not be changed during device operations (program, erase, or sector lock).

## 11.3 Read Configuration Register Command Sequence

The configuration register can be read with a four-cycle command sequence. The first two cycles are standard unlock sequences. On the third cycle, the data should be C6h and address bits should be 555h. During the fourth cycle, the configuration code should be read out of the data bus with the address bus set to address 000h. Once the data has been read from the configuration register, a software reset command is required to set the device into the correct set mode.

### Read Mode Setting

On power-up or hardware reset, the device is set to be in asynchronous read mode. This setting allows the system to enable or disable burst mode during system operations.

### Programmable Wait State Configuration

The programmable wait state feature informs the device of the number of clock cycles that must elapse after AVD# is driven active before data will be available. This value is determined by the input frequency of the device. **Configuration Bit CR13–CR11** determine the setting (see Table 11.1).

The wait state command sequence instructs the device to set a particular number of clock cycles for the initial access in burst mode. The number of wait states that should be programmed into the device is directly related to the clock frequency.

**Table 11.1** Programmable Wait State Settings

| CR13 | CR12 | CR11 | Total Initial Access Cycles |
|---|---|---|---|
| 0 | 0 | 0 | Reserved |
| 0 | 0 | 1 | 3 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 5 |
| 1 | 0 | 0 | 6 |
| 1 | 0 | 1 | 7 (default) |
| 1 | 1 | 0 | 8 |
| 1 | 1 | 1 | 9 |

***Notes***

1. *Upon power-up or hardware reset, the default setting is seven wait states.*
2. *RDY will default to being active with data when the Wait State Setting is set to a total initial access cycle of 2.*

It is recommended that the wait state command sequence be written, even if the default wait state value is desired, to ensure the device is set as expected. A hardware reset will set the wait state to the default setting.

### Programmable Wait State

The host system should set **CR13-CR11** to 100 for a clock frequency of 66 MHz for the system/device to execute at maximum speed.

Table 11.2 describes the typical number of clock cycles (wait states) for various conditions.

**Table 11.2** Wait States for Handshaking

| Conditions at Address | Typical No. of Clock Cycles after AVD# Low |
|---|---|
| | 66 MHz |
| Initial address ($V_{CCQ}$ = 1.8 V) | 6 |

### Handshaking

For optimal burst mode performance, the host system must set the appropriate number of wait states in the flash device depending on the clock frequency.

The autoselect function allows the host system to determine whether the flash device is enabled for handshaking.

### Burst Length Configuration

The device supports four different read modes: continuous mode, and 8 and 16 word linear with wrap around modes. A continuous sequence (default) begins at the starting address and advances the address pointer until the burst operation is complete. If the highest address in the device is reached during the continuous burst read mode, the address pointer wraps around to the lowest address.

For example, an eight-word linear read with wrap around begins on the starting address written to the device and then advances to the next 8 word boundary. The address pointer then returns to the 1st word after the previous eight word boundary, wrapping through the starting location. The sixteen linear wrap around mode operates in a fashion similar to the eight-word mode.

Table 11.3 shows the **CR2-CR0** and settings for the four read modes.

**Table 11.3** Burst Length Configuration

| Burst Modes | Address Bits | | |
|---|---|---|---|
| | CR2 | CR1 | CR0 |
| Continuous | 0 | 0 | 0 |
| 8-word linear | 0 | 1 | 0 |
| 16-word linear | 0 | 1 | 1 |

***Notes***

1. *Upon power-up or hardware reset the default setting is continuous.*
2. *All other conditions are reserved.*

## Burst Wrap Around

By default, the device will perform burst wrap around with **CR3** set to a '1'. Changing the **CR3** to a '0' disables burst wrap around.

## RDY Configuration

By default, the device is set so that the RDY pin will output $V_{OH}$ whenever there is valid data on the outputs. The device can be set so that RDY goes active one data cycle before active data. **CR8** determines this setting; "1" for RDY active (default) with data, "0" for RDY active one clock cycle before valid data.

## RDY Polarity

By default, the RDY pin will always indicate that the device is ready to handle a new transaction with **CR10** set to a '1'. In this case, the RDY pin is active high. Changing the **CR10** to a '0' sets the RDY pin to be active low. In this case, the RDY pin will always indicate that the device is ready to handle a new transaction when low.

# 12. Configuration Register

Table 12.1 shows the address bits that determine the configuration register settings for various device functions.

**Table 12.1** Configuration Register

| CR Bit | Function | Settings (Binary) |
|---|---|---|
| CR15 | Reserved | 0 = Default (Note 3) |
| CR14 | Reserved | 0 = Default |
| CR13<br>CR12<br>CR11 | Programmable Wait State | 000 = Reserved<br>001 = Data is valid on the 3rd active CLK edge after AVD# transition to $V_{IH}$<br>010 = Data is valid on the 4th active CLK edge after AVD# transition to $V_{IH}$<br>011 = Data is valid on the 5th active CLK edge after AVD# transition to $V_{IH}$<br>100 = Data is valid on the 6th active CLK edge after AVD# transition to $V_{IH}$<br>101 = Data is valid on the 7th active CLK edge after AVD# transition to $V_{IH}$ (default)<br>110 = Data is valid on the 8th active CLK edge after AVD# transition to $V_{IH}$<br>111 = Data is valid on the 9th active CLK edge after AVD# transition to $V_{IH}$ |
| CR10 | RDY Polarity | 0 = RDY signal is active low<br>1 = RDY signal is active high (default) |
| CR9 | Reserved | 1 = Default |
| CR8 | RDY | 0 = RDY active one clock cycle before data<br>1 = RDY active with data (default) |
| CR7 | Reserved | 1 = default |
| CR6 | Reserved | 1 = default |
| CR5 | Reserved | 0 = default |
| CR4 | Reserved | 0 = default |
| CR3 | Burst Wrap Around | 0 = Reserved<br>1 = Wrap Around Burst (default) |
| CR2<br>CR1<br>CR0 | Burst Length | 000 = Continuous (default)<br>010 = 8-Word Linear Burst<br>011 = 16-Word Linear Burst<br>100 = Reserved<br>(All other bit settings are reserved) |

***Notes***

1. *Device will be in the default state upon power-up or hardware reset.*
2. *CR3 will always equal to 1 (Wrap around mode) when CR0,CR1,CR2 = 000 (continuous Burst mode).*
3. *Trim bit option:*
   *0 = Synchronous Read Mode*
   *1= Asynchronous Read Mode (Default)*

## 12.1 Reset Command

Writing the reset command resets the banks 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 bank to which the system was writing 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 bank to which the system was writing to the read mode. If the program command sequence is written to a bank that is in the Erase Suspend mode, writing the reset command returns that bank 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 a bank entered the autoselect mode while in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.

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

Note: 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 to reading array data. The standard RESET command will not work. See Table 10.5 on page 27 for details on this command sequence.

## 12.2 Autoselect Command Sequence

The autoselect command sequence allows the host system to access the manufacturer and device codes, and determine whether or not a sector is protected. The autoselect command sequence may be written to an address within a bank that is either in the read or erase-suspend-read mode. The autoselect command may not be written while the device is actively programming or erasing in the other bank. Autoselect does not support simultaneous operations or burst mode.

The autoselect command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle that contains the bank address and the autoselect command. The bank then enters the autoselect mode. The system may read at any address within the same bank any number of times without initiating another autoselect command sequence. The following table describes the address requirements for the various autoselect functions, and the resulting data. BA represents the bank address. The device ID is read in three cycles. During this time, other banks are still available to read the data from the memory.

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

## 12.3 Enter/Exit Secured Silicon Sector Command Sequence

The Secured Silicon Sector region provides a secured data area containing a random, eight word electronic serial number (ESN). The system can access the Secured Silicon Sector region by issuing the three-cycle Enter Secured Silicon Sector command sequence. The device continues to access the Secured Silicon Sector region until the system issues the four-cycle Exit Secured Silicon Sector command sequence. The Exit Secured Silicon Sector command sequence returns the device to normal operation. The Secured Silicon Sector is not accessible when the device is executing an Embedded Program or embedded Erase algorithm. Table 12.3 on page 37 shows the address and data requirements for both command sequences.

## 12.4 Program Command Sequence

### 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 12.3 on page 37 shows the address and data requirements for the program command sequence.

When the Embedded Program algorithm is complete, that bank then returns to the read mode and addresses are no longer latched. The system can determine the status of the program operation by monitoring DQ7 or DQ6/DQ2. Refer to the *Write Operation Status* section for information on these status bits.

Any commands written to the device during the Embedded Program Algorithm are ignored. Note that a **hardware reset** immediately terminates the program operation. The program command sequence should be reinitiated once that bank has returned to the read mode, to ensure data integrity.

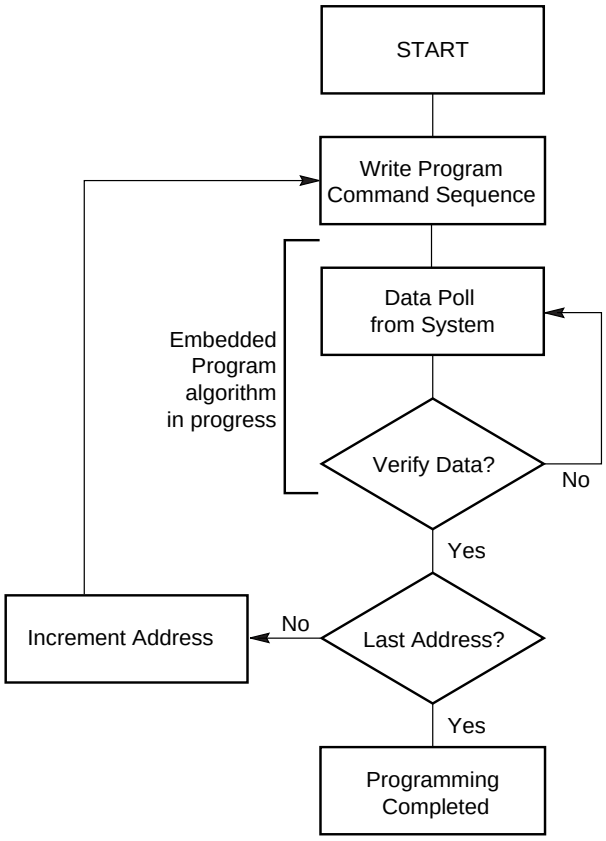
Programming is allowed in any sequence and across sector boundaries. **A bit cannot be programmed from “0” back to a “1.”** Attempting to do so may causes that bank to set DQ5 = 1 (change-up condition). However, a succeeding read will show that the data is still “0.” Only erase operations can convert a “0” to a “1.”

## 12.5 Accelerated Program

The device offers accelerated program operations through the $V_{PP}$ input. The device uses the higher voltage on the $V_{PP}$ input to accelerate the operation.

Figure 12.1 illustrates the algorithm for the program operation. Refer to the *Erase/Program Operations* table in the AC Characteristics section for parameters, and Figure 19.6, on page 54 for timing diagrams.

**Figure 12.1** Program Operation




***Note***
*1. See Table 12.3 on page 37 for program command sequence.*

## 12.6 Write Buffer Programming Command Sequence

Write Buffer Programming Sequence allows for faster programming as compared to the standard Program Command Sequence. See Table 12.2 for the program command sequence.

**Table 12.2** Write Buffer Command Sequence

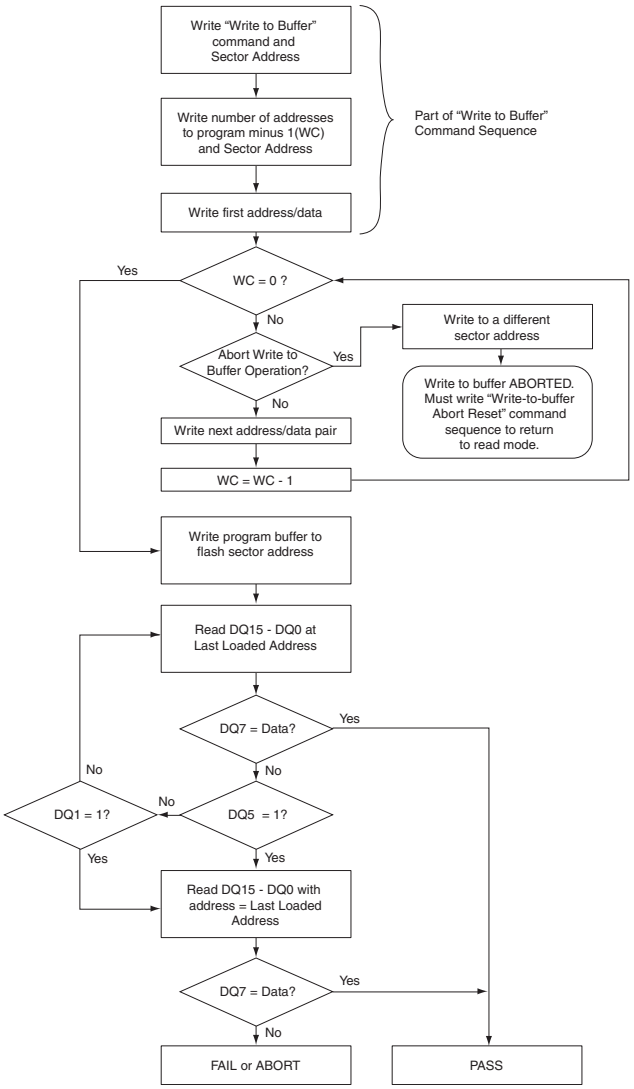
| Sequence | Address | Data | Comment |
|---|---|---|---|
| Unlock Command 1 | 555 | 00AA | Not required in the Unlock Bypass mode |
| Unlock Command 2 | 2AA | 0055 | Same as above |
| Write Buffer Load | Starting Address | 0025h | |
| Specify the Number of Program Locations | Starting Address | Word Count | Number of locations to program minus 1 |
| Load 1st data word | Starting Address | Program Data | All addresses must be within write-buffer-page boundaries, but do not have to be loaded in any order |
| Load next data word | Write Buffer Location | Program Data | Same as above |
| ... | ... | ... | Same as above |
| Load last data word | Write Buffer Location | Program Data | Same as above |

**Table 12.2** Write Buffer Command Sequence

| | | | |
|---|---|---|---|
| Write Buffer Program Confirm | Sector Address | 0029h | This command must follow the last write buffer location loaded, or the operation will ABORT |
| Device goes busy | | | |
| Status monitoring through DQ pins (Perform Data Bar Polling on the **Last Loaded Address**) | | | |

***Notes***
1. *Write buffer addresses must be loaded in sequential order.*

**Figure 12.2** Write Buffer Programming Operation




## 12.7 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 preprogram 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 12.3 on page 37 shows the address and data requirements for the chip erase command sequence.

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

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

## 12.8 Sector Erase Command Sequence

Sector erase in normal mode 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 12.3 on page 37 shows the address and data requirements for the sector erase command sequence.

The device does *not* require the system to preprogram 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.

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

When the Embedded Erase algorithm is complete, the bank returns to reading array data and addresses are no longer latched. Note that while the Embedded Erase operation is in progress, the system can read data from the non-erasing banks. The system can determine the status of the erase operation by reading DQ7 or DQ6/ DQ2 in the erasing bank. Refer to the *Write Operation Status* section for information on these status bits.

Once the sector erase operation has begun, 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 that bank has returned to reading array data, to ensure data integrity.

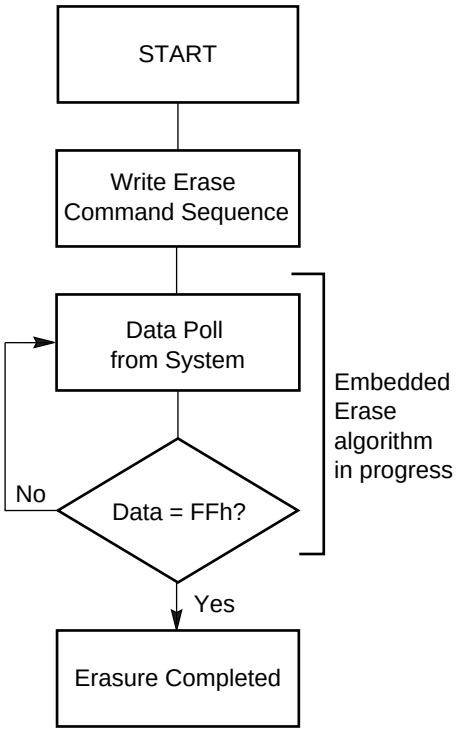
### Accelerated Sector Erase

The device offers accelerated sector erase operation through the $V_{PP}$ function. This method of erasing sectors is faster than the standard sector erase command sequence. **The accelerated sector erase function must not be used more than 100 times per sector.** In addition, accelerated sector erase should be performed at room temperature (30°C ± 10°C).

The following procedure is used to perform accelerated sector erase:

1. Sectors to be erased must be DYB cleared. All sectors that remain locked will not be erased.
2. Apply 9V to the $V_{PP}$ input. This voltage must be applied at least 1 µs before executing Step 3.
3. Issue the standard chip erase command.
4. Monitor status bits DQ2/DQ6 or DQ7 to determine when erasure is complete, just as in the standard erase operation. See *Write Operation Status* for further details.
5. Lower $V_{PP}$ from 9V to $V_{CC}$.

**Figure 12.3** Erase Operation




***Note***
*See the section on DQ3 for information on the sector erase start timeout state indicator.*

## 12.9 Erase Suspend/Erase Resume Commands

The Erase Suspend command, B0h, allows the system to interrupt a sector erase operation and then read data from, program data to, any sector not selected for erasure. The system may also lock or unlock any sector while the erase operation is suspended. **The system must not write the sector lock/unlock command to sectors selected for erasure.** The bank address is required when writing this command. This command is valid only during the sector erase operation. 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 a maximum of $t_{ESL}$, 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 has been suspended, the bank 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.) The system may also lock or unlock any sector while in the erase-suspend-read mode. 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 the *Write Operation Status* section for information on these status bits.

After an erase-suspended program operation is complete, the bank 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 program operation. Refer to the *Write Operation Status* section for more information.

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

To resume the sector erase operation, the system must write the Erase Resume command. The bank address of the erase-suspended bank is required when writing this command. Further writes of the Resume command are ignored. Another Erase Suspend command can be written after the chip has resumed erasing.

## 12.10 Program Suspend/Program Resume Commands

The Program Suspend command allows the system to interrupt a embedded programming operation or a “Write to Buffer” programming operation so that data can read from any non-suspended sector. When the Program Suspend command is written during a programming process, the device halts the programming operation within $t_{PSL}$, program suspend latency, and updates the status bits. Addresses are defined when writing the Program Suspend command.

After the programming operation has been 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 Secured Silicon Sector area (One Time Program area), then user must use the proper command sequences to enter and exit this region.

The system may also write the autoselect command sequence when the device is in Program Suspend mode. The device allows reading autoselect codes in the suspended sectors, since the codes are not stored in the memory array. When the device exits the autoselect mode, the device reverts to Program Suspend mode, and is ready for another valid operation. See “Autoselect Command Sequence” 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” for more information.

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 Program Resume command are ignored. Another Program Suspend command can be written after the device has resume programming.

## 12.11 Volatile Sector Protection Command Set

The Volatile Sector Protection Command Set permits the user to set the Dynamic Protection Bit (DYB), clear the Dynamic Protection Bit (DYB), and read the logic state of the Dynamic Protection Bit (DYB).

The **Volatile Sector Protection Command Set Entry** command sequence must be issued prior to any of the commands listed following to enable proper command execution.

Note that issuing the **Volatile Sector Protection Command Set Entry** command disables reads and writes for the bank selected with the command. Reads for other banks excluding the selected bank are allowed.

- DYB Set Command
- DYB Clear Command
- DYB Status Read Command

The DYB Set/Clear command is used to set or clear a DYB for a given sector. The high order address bits (Amax-A13 for VS/XS064R) are issued at the same time as the code 00h or 01h on DQ7-DQ0. All other DQ data bus pins are ignored during the data write cycle. The DYBs are modifiable at any time. The DYBs are set at power-up or hardware reset.

The programming state of the DYB for a given sector can be verified by writing a DYB Status Read Command to the device.

**Note:** The bank entered during entry is the active bank. Take for example the active bank is BA0. Any reads in BA0 will result in status reads of the DYB bit. If the user wants to set (programmed to “0”) in a different bank other than the active bank, say for example BA5, then the active bank switches from BA0 to BA5. Reading in BA5 will result in status read of the bit whereas reading in BA0 will result in true data.

The **Volatile Sector Protection Command Set Exit** command must be issued after the execution of the commands listed previously to reset the device to read mode.

Note that issuing the **Volatile Sector Protection Command Set Exit** command re-enables reads and writes for the bank selected.

**Table 12.3** Command Definitions (Sheet 1 of 3)

| Command Sequence (Notes) | Cycles | Bus Cycles (Notes 1–6) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Asynchronous Read ((7)) | 1 | RA | RD | | | | | | | | | | | | |
| Reset (8) | 1 | XXX | F0 | | | | | | | | | | | | |

**Table 12.3** Command Definitions (Sheet 2 of 3)

| Command Sequence (Notes) | | Cycles | Bus Cycles (Notes 1–6) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Autoselect (9) | Manufacturer ID | 4 | 555 | AA | 2AA | 55 | (BA) 555 | 90 | (BA) X00 | 0001 | | | | | | |
| | Device ID | 6 | 555 | AA | 2AA | 55 | (BA) 555 | 90 | (BA) X01 | (10) | (BA) X0E | (10) | (BA) X0F | (10) | | |
| | Indicator Bits (11) | 4 | 555 | AA | 2AA | 55 | (BA) 555 | 90 | (BA) X0D | (11) | | | | | | |
| | Revision ID | 4 | 555 | AA | 2AA | 55 | (BA) 555 | 90 | (BA) X03 | | | | | | | |
| CFI | | 1 | 55 | 98 | | | | | | | | | | | | |
| Program | | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | | | |
| Write to Buffer (15) | | 6 | 555 | AA | 2AA | 55 | SA | 25 | SA | WC | PA | PD | WBL | PD | | |
| Program Buffer to Flash | | 1 | SA | 29 | | | | | | | | | | | | |
| Write to Buffer Abort Reset (16) | | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | | | |
| 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 (12) | | 1 | BA | B0 | | | | | | | | | | | | |
| Erase Resume / Program Resume (13) | | 1 | BA | 30 | | | | | | | | | | | | |
| Set Config. Register | | 4 | 555 | AA | 2AA | 55 | 555 | D0 | X00 | CR | | | | | | |
| Read Configuration Register | | 4 | 555 | AA | 2AA | 55 | 555 | C6 | X00 | CR | | | | | | |
| **Lock Register Command Set Definitions** | | | | | | | | | | | | | | | | |
| Lock | Lock Register Command Set Entry | 3 | 555 | AA | 2AA | 55 | 555 | 40 | | | | | | | | |
| | Lock Register Bits Program | 2 | XX | A0 | 00 | data | | | | | | | | | | |
| | Lock Register Bits Read | 1 | (BA0) 00 | data | | | | | | | | | | | | |
| | Lock Register Command Set Exit (18) | 2 | XX | 90 | XX | 00 | | | | | | | | | | |

**Table 12.3** Command Definitions (Sheet 3 of 3)

| Command Sequence (Notes) | | Cycles | Bus Cycles (Notes 1–6) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| **Secured Silicon Sector Command Definitions** | | | | | | | | | | | | | | | | |
| Secured Silicon Sector | Secured Silicon Sector Entry (17) | 3 | 555 | AA | 2AA | 55 | 555 | 88 | | | | | | | | |
| | Secured Silicon Sector Program | 2 | XX | A0 | 00 | data | | | | | | | | | | |
| | Secured Silicon Sector Read | 1 | 00 | data | | | | | | | | | | | | |
| | Secured Silicon Sector Exit | 4 | 555 | AA | 2AA | 55 | 555 | 90 | XX | 00 | | | | | | |
| **Volatile Sector Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| DYB | Volatile Sector Protection Command Set Entry (17) | 3 | 555 | AA | 2AA | 55 | (BA) 555 | E0 | | | | | | | | |
| | DYB Set | 2 | XX | A0 | (BA) SA | 00 | | | | | | | | | | |
| | DYB Clear | 2 | XX | A0 | (BA) SA | 01 | | | | | | | | | | |
| | DYB Status Read | 1 | (BA) SA | RD(0) | | | | | | | | | | | | |
| | Volatile Sector Protection Command Set Exit | 2 | XX | 90 | XX | 00 | | | | | | | | | | |

***Legend***

*X = Don't care*

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

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

*PA = Address of the memory location to be programmed. Addresses latch on the falling edge of the WE# or CE# pulse, whichever happens later.*

*PD = Data to be programmed at location PA. Data latches on the rising edge of WE# or CE# pulse, whichever happens first.*

*PD(0) = Secured Silicon Sector Lock Bit. PD(0), or bit[0].*

*PD(3) = Protection Mode OTP Bit. PD(3) or bit[3].*

*SA = Address of the sector to be verified (in autoselect mode) or erased. SA includes BA. Address bits $A_{max}$ - A13 uniquely select any sector.*

*BA = Address of the bank A21-A19.*

*CR = Configuration Register set by data bits D15-D0.*

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

*RD(1) = DQ1 protection indicator bit. If protected, DQ1 = 0, if unprotected, DQ1 = 1.*

*RD(2) = DQ2 protection indicator bit. If protected, DQ2 = 0, if unprotected, DQ2 = 1.*

*RD(4) = DQ4 protection indicator bit. If protected, DQ4 = 0, if unprotected, DQ4 = 1.*

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

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

***Notes***

1. *See Table 9.1 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Except for the read cycle and the fourth cycle of the autoselect command sequence, all bus cycles are write cycles.*
4. *Data bits DQ15–DQ8 are don't care in command sequences, except for RD and PD.*
5. *Unless otherwise noted, address bits $A_{max}$–A12 are don't cares.*
6. *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.*
7. *No unlock or command cycles required when bank is reading array data.*
8. *The Reset command is required to return to reading array data (or to the erase-suspend-read mode if previously in Erase Suspend) when a bank is in the autoselect mode, or if DQ5 goes high (while the bank is providing status information).*
9. *The fourth cycle of the autoselect command sequence is a read cycle. The system must read device IDs across the 4th, 5th, and 6th cycles, The system must provide the bank address. See the Autoselect Command Sequence section for more information.*
10. *See Table 9.1 for description of bus operations.*
11. *See the Autoselect Command Sequence on page 32.*

12. *The system may read and program in non-erasing sectors, or enter the autoselect mode, when in the Erase Suspend mode. The Erase Suspend command is valid only during a sector erase operation, and requires the bank address.*
13. *The Erase Resume command is valid only during the Erase Suspend mode, and requires the bank address.*
14. *Command is valid when device is ready to read array data or when device is in autoselect mode.*
15. *The total number of cycles in the command sequence is determined by the number of words written to the write buffer. The maximum number of cycles in the command sequence is 37.*
16. *Command sequence resets device for next command after write-to-buffer operation.*
17. *Entry commands are needed to enter a specific mode to enable instructions only available within that mode.*
18. *The Exit command must be issued to reset the device into read mode. Otherwise the device will hang.*
19. *Note: Autoselect, CFI, OTP, and all ASP modes cannot be nested with each other.*
20. *Only A7 - A0 (lower address bits) are used*
21. *$A_{max}$–A0 (all address bits) are used.*
22. *Requires the RESET# command to configure the configuration register.*

# 13. 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 13.2 on page 45 and the following subsections describe the function of these bits. DQ7 and DQ6 each offers a method for determining whether a program or erase operation is complete or in progress.

## 13.1 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 a bank is in Erase Suspend. Data# Polling is valid after the rising edge of the final WE# pulse in the command sequence. **Note that the Data# Polling is valid only for the last word being programmed in the write-buffer-page during Write Buffer Programming. Reading Data# Polling status on any word other than the last word to be programmed in the write-buffer-page will return false status information.**

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 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 $t_{PSP}$, then that bank returns to the read mode.

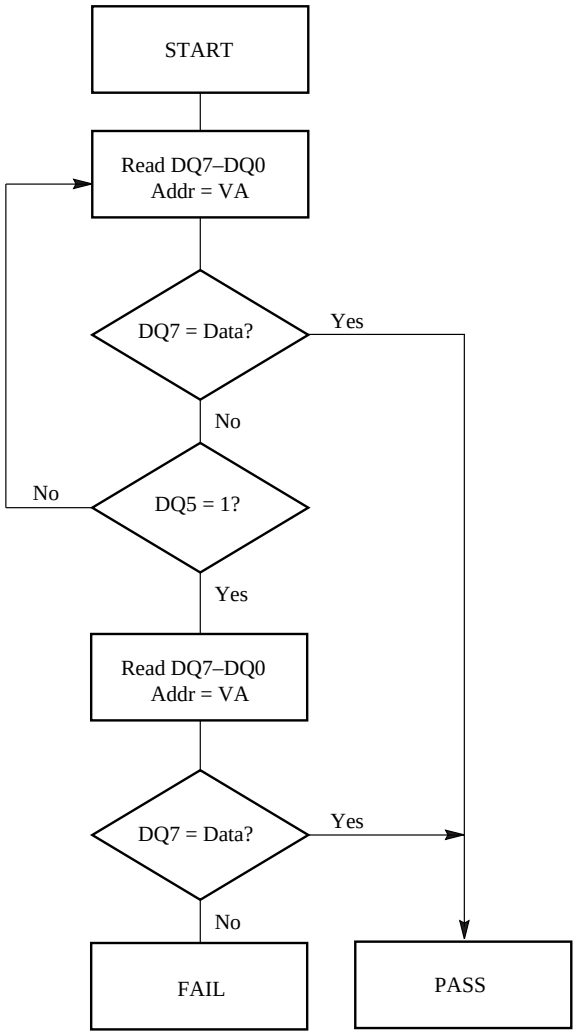
During the Embedded Erase algorithm, Data# Polling produces a “0” on DQ7. When the Embedded Erase algorithm is complete, or if the bank enters the Erase Suspend mode, Data# Polling produces a “1” on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status 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 $t_{ASP}$, then the bank 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 DQ6–DQ0 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 has completed the program or erase operation and DQ7 has valid data, the data outputs on DQ6–DQ0 may be still invalid. Valid data on DQ7–DQ0 will appear on successive read cycles.

Table 13.2 on page 45 shows the outputs for Data# Polling on DQ7. Figure 13.1 on page 41 shows the Data# Polling algorithm. Figure 19.8 on page 55 in the *AC Characteristics* section shows the Data# Polling timing diagram.

**Figure 13.1** 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.*

## 13.2 RDY: Ready

The RDY pin is a dedicated status output that indicates valid output data on A/DQ15–A/DQ0 during burst (synchronous) reads. When RDY is asserted (RDY = $V_{OH}$), the output data is valid and can be read. When RDY is de-asserted (RDY = $V_{OL}$), the system should wait until RDY is re-asserted before expecting the next word of data.

In synchronous (burst) mode with CE# = OE# = $V_{IL}$, RDY is de-asserted under the following conditions: during the initial access; after crossing the internal boundary between addresses 7Eh and 7Fh (and addresses offset from these by a multiple of 64). The RDY pin will also switch during status reads when a clock signal drives the CLK input. In addition, RDY = $V_{OH}$ when CE# = $V_{IL}$ and OE# = $V_{IH}$, and RDY is Hi-Z when CE# = $V_{IH}$.

In asynchronous (non-burst) mode, the RDY pin does not indicate valid or invalid output data. Instead, RDY = $V_{OH}$ when CE# = $V_{IL}$, and RDY is Hi-Z when CE# = $V_{IH}$.

## 13.3 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 has entered the Erase Suspend mode. Toggle Bit I may be read at any address in the same bank, 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. Note that OE# must be low during toggle bit status reads. 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 $t_{ASP}$, all sectors protected toggle time, 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 the subsection on *DQ7: Data# Polling*).

If a program address falls within a protected sector, DQ6 toggles for approximately $t_{PSP}$ 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.

See the following for additional information: (toggle bit flowchart), *DQ6: Toggle Bit I* (description), Figure 19.9 (toggle bit timing diagram), and Table 13.1 on page 43 (compares DQ2 and DQ6).

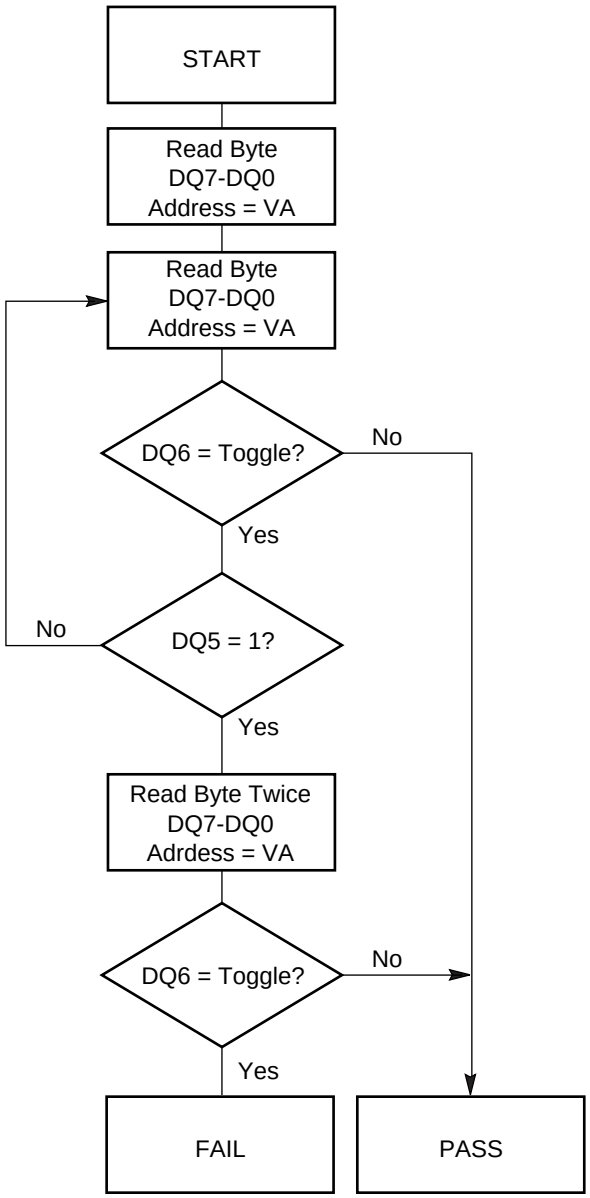
## 13.4 DQ2: Toggle Bit II

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. 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 have been selected for erasure. Note that OE# must be low during toggle bit status reads. 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 13.2 on page 45 to compare outputs for DQ2 and DQ6.

See the following for additional information: (toggle bit flowchart), *DQ6: Toggle Bit I* (description), Figure 19.9 on page 56 (toggle bit timing diagram), and Table 13.1 on page 43 (compares DQ2 and DQ6).

**Figure 13.2** 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 the subsections on DQ6 and DQ2 for more information.*

**Table 13.1** DQ6 and DQ2 Indications

| If device is | and the system reads | then DQ6 | and DQ2 |
|---|---|---|---|
| programming, | at any address, | toggles, | does not toggle. |
| actively erasing, | at an address within a sector selected for erasure, | toggles, | also toggles. |
| | at an address within sectors *not* selected for erasure, | toggles, | does not toggle. |

**Table 13.1** DQ6 and DQ2 Indications

| If device is | and the system reads | then DQ6 | and DQ2 |
|---|---|---|---|
| erase suspended, | at an address within a sector selected for erasure, | does not toggle, | toggles. |
| | at an address within sectors *not* selected for erasure, | returns array data, | returns array data. The system can read from any sector not selected for erasure. |
| programming in erase suspend | at any address, | toggles, | is not applicable. |

## 13.5 Reading Toggle Bits DQ6/DQ2

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 has 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 the section on DQ5). 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 has 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.

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.

## 13.6 DQ5: Exceeded Timing Limits

DQ5 indicates whether the program or erase time has 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.

The device may output a “1” on DQ5 if the system tries to program a “1” to a location that was previously programmed to “0.” **Only an erase operation can change a “0” back to a “1.”** Under this condition, the device halts the operation, and when the timing limit has been exceeded, DQ5 produces a “1.”

Under both these conditions, the system must write the reset command to return to the read mode (or to the erase-suspend-read mode if a bank was previously in the erase-suspend-program mode).

## 13.7 DQ3: Sector Erase Start Timeout State Indicator

After writing a sector erase command sequence, the system may read DQ3 to determine whether or not erasure has begun. (The sector erase timer does not apply to the chip 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.”

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 has 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 will accept additional sector erase commands. To ensure the command has been 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 13.2 shows the status of DQ3 relative to the other status bits.

## 13.8 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 Programming Operation* on page 19 for more details.

**Table 13.2** Write Operation Status

| Status | | | DQ7 (2) | DQ6 | DQ5 (1) | DQ3 | DQ2 (2) | DQ1 (4) |
|---|---|---|---|---|---|---|---|---|
| Standard Mode | Embedded Program Algorithm | | DQ7# | Toggle | 0 | N/A | No toggle | 0 |
| | Embedded Erase Algorithm | | 0 | Toggle | 0 | 1 | Toggle | N/A |
| Program Suspend Mode (3) | Reading within Program Suspended Sector | | Valid data for all address except the address being programed, which will return invalid data | | | | | |
| | Reading within Non-Program Suspended Sector | | Data | | | | | |
| Erase Suspend Mode | Erase-Suspend-Read | Erase Suspended Sector | 1 | No toggle | 0 | N/A | Toggle | N/A |
| | | Non-Erase Suspended Sector | Data | Data | Data | Data | Data | Data |
| | Erase-Suspend-Program | | DQ7# | Toggle | 0 | N/A | N/A | N/A |
| Write to Buffer (5) | BUSY State | | DQ7# | Toggle | 0 | N/A | N/A | 0 |
| | Exceeded Timing Limits | | DQ7# | Toggle | 1 | N/A | N/A | 0 |
| | ABORT State | | DQ7# | Toggle | 0 | N/A | N/A | 1 |

***Notes***

1. *DQ5 switches to '1' when an Embedded Program or Embedded Erase operation has exceeded the maximum timing limits. Refer to the section on DQ5 for more information.*
2. *DQ7 and DQ2 require a valid address when reading status information. Refer to the appropriate subsection for further details.*
3. *Data are invalid for addresses in a Program Suspended sector.*
4. *DQ1 indicates the Write to Buffer ABORT status during Write Buffer Programming operations.*
5. *The data-bar polling algorithm should be used for Write Buffer Programming operations. Note that DQ7# during Write Buffer Programming indicates the data-bar for DQ7 data* ***for the LAST LOADED WRITE-BUFFER ADDRESS location****.*

# 14. Absolute Maximum Ratings

| | |
|---|---|
| Storage Temperature | –65°C to +150°C |
| Ambient Temperature with Power Applied | –65°C to +125°C |
| Voltage with Respect to Ground, All Inputs and I/Os | |
| except $V_{PP}$ (1) | –0.5 V to $V_{CC}$ + 0.5 V |
| $V_{CC}$ (1) | –0.5 V to +2.5 V |
| $V_{PP}$ (2) | –0.5 V to + 9.5 V |
| Output Short Circuit Current (3) | 100 mA |

***Notes***

1. *Minimum DC voltage on input or I/Os is –0.5 V. During voltage transitions, input at I/Os may undershoot $V_{SS}$ to –2.0 V for periods of up to 20 ns during voltage transitions inputs might overshooot to $V_{CC}$ +0.5 V for periods up to 20 ns. See Figure 14.1. Maximum DC voltage on output and I/Os is $V_{CC}$ + 0.5 V. During voltage transitions outputs may overshoot to $V_{CC}$ + 2.0 V for periods up to 20 ns. See Figure 14.2.*
2. *Minimum DC input voltage on $V_{PP}$ is –0.5 V. During voltage transitions, $V_{PP}$ may undershoot $V_{SS}$ to –2.0 V for periods of up to 20 ns. See Figure 14.1. Maximum DC input voltage on $V_{PP}$ is +9.5 V which may overshoot to +10.5 V 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.*

**Figure 14.1** Maximum Negative Overshoot Waveform

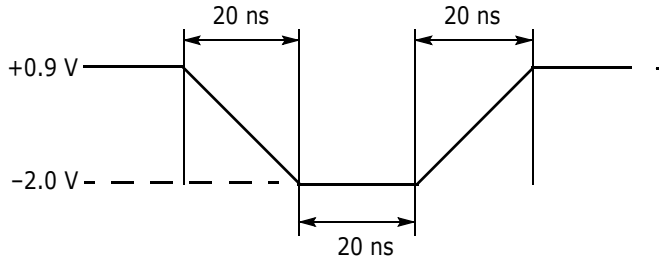
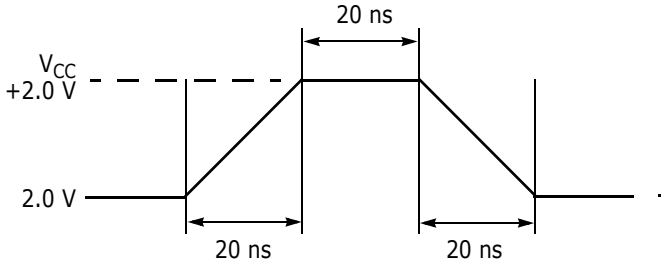



**Figure 14.2** Maximum Positive Overshoot Waveform




# 15. Operating Ranges

| | |
|---|---|
| Ambient Temperature ($T_A$) | –25°C to +85°C |
| Ambient Temperature ($T_A$) during Accelerated Sector Erase | +20°C to +40°C |
| $V_{CC}$ Supply Voltages | |
| $V_{CC}$ min | +1.70 V |
| $V_{CC}$ max | +1.95 V |

***Notes***

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

# 16. DC Characteristics

**Table 16.1** CMOS Compatible

| Parameter | Description | Test Conditions (1) | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Load Current | $V_{IN} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC\ max}$ | | | | ±1 | µA |
| $I_{LO}$ | Output Leakage Current | $V_{OUT} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC\ max}$ | | | | ±1 | µA |
| $I_{CCB}$ | $V_{CC}$ Active Burst Read Current (5) | CE# = $V_{IL}$, OE# = $V_{IL}$, burst length = 8 | 66 MHz | | 24 | 33 | mA |
| | | | 83 MHz | | 26 | 36 | |
| | | | 104 MHz | | 33 | 44 | |
| | | CE# = $V_{IL}$, OE# = $V_{IL}$, burst length = 16 | 66 MHz | | 24 | 35 | |
| | | | 83 MHz | | 26 | 38 | |
| | | | 104 MHz | | 30 | 40 | |
| | | CE# = $V_{IL}$, OE# = $V_{IL}$, burst length = continuous | 66 MHz | | 28 | 39 | |
| | | | 83 MHz | | 30 | 42 | |
| | | | 104 MHz | | 32 | 44 | |
| $I_{CC1}$ | $V_{CC}$ Active Asynchronous Read Current (2) | CE# = $V_{IL}$, OE# = $V_{IH}$ | 5 MHz | | 20 | 40 | mA |
| | | | 1 MHz | | 10 | 20 | mA |
| $I_{CC2}$ | $V_{CC}$ Active Write Current (3) | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{PP} = V_{IH}$ | | | 19 | 52.5 | mA |
| $I_{CC3}$ | $V_{CC}$ Standby Current (4) | CE# = $V_{IH}$, RESET# = $V_{IH}$ (Note 8) | | | 20 | 70 | µA |
| $I_{CC4}$ | $V_{CC}$ Reset Current | RESET# = $V_{IL}$, CLK = $V_{IL}$ (Note 8) | | | 150 | 250 | µA |
| $I_{CC5}$ | $V_{CC}$ Active Current (Read While Write) | CE# = $V_{IL}$, OE# = $V_{IL}$ (Note 8) (Note 9) | 66 MHz | | 48 | 60 | mA |
| | | | 83 MHz | | 50 | 60 | |
| | | | 104 MHz | | 52 | 60 | |
| $I_{CC6}$ | $V_{CC}$ Sleep Current (10) | CE# = $V_{IL}$, OE# = $V_{IH}$ | | | 20 | 70 | µA |
| $I_{PPW}$ | Accelerated Program Current (6) | $V_{PP}$ = 9 V | | | 20 | 30 | mA |
| $I_{PPE}$ | Accelerated Erase Current (6) | $V_{PP}$ = 9 V | | | 20 | 30 | mA |
| $V_{IL}$ | Input Low Voltage | | | –0.5 | | 0.4 | V |
| $V_{IH}$ | Input High Voltage | | | $V_{IO}$ – 0.4 | | $V_{IO}$ + 0.2 | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 100 µA, $V_{CC} = V_{CC\ min}$ | | | | 0.1 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = –100 µA, $V_{CC} = V_{CC\ min}$ | | $V_{IO}$ – 0.1 | | | V |
| $V_{ID}$ | Voltage for Accelerated Program | | | 8.5 | | 9.5 | V |
| $V_{LKO}$ | Low $V_{CC}$ Lock-out Voltage | | | 1.0 | | 1.1 | V |

***Notes***

1. *Maximum $I_{CC}$ specifications are tested with $V_{CC} = V_{CC}$max.*
2. *The $I_{CC}$ current listed is typically less than 2 mA/MHz, with OE# at $V_{IH}$.*
3. *$I_{CC}$ active while Embedded Erase or Embedded Program is in progress.*
4. *Device enters automatic sleep mode when addresses are stable for $t_{ACC}$ + 20 ns. Typical sleep mode current is equal to $I_{CC3}$.*
5. *Specifications assume 8 I/Os switching.*
6. *Not 100% tested. $V_{PP}$ is not a power supply pin.*
7. *While measuring Output Leakage Current, CE# should be at $V_{IH}$.*
8. *$V_{IH} = V_{CC}$ ± -0.2 V and $V_{IL}$ > -0.1V.*
9. *Clock Frequency 66 MHz and in Continuous Mode..*
10. *If the trim bit is set, $I_{CC6}$ may increase due to disabling auto-power down in the auto-detect mode for burst function.*

# 17. Test Conditions

**Figure 17.1** Test Setup

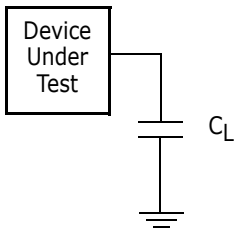



**Table 17.1** Test Specifications

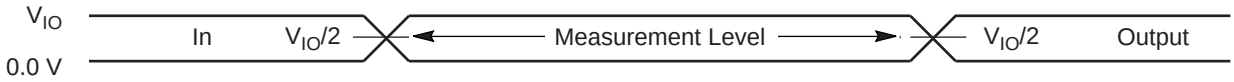
| Test Condition | All Speeds | Unit |
|---|---|---|
| Output Load Capacitance, $C_L$ (including jig capacitance) | 30 | pF |
| Input Rise and Fall Times | 3 @ 66 MHz | ns |
| Input Pulse Levels | 0.0–$V_{CC}$ | V |
| Input timing measurement reference levels | $V_{IO}/2$ | V |
| Output timing measurement reference levels | $V_{IO}/2$ | V |

# 18. 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) |

## 18.1 Switching Waveforms

**Figure 18.1** Input Waveforms and Measurement Levels



# 19. AC Characteristics

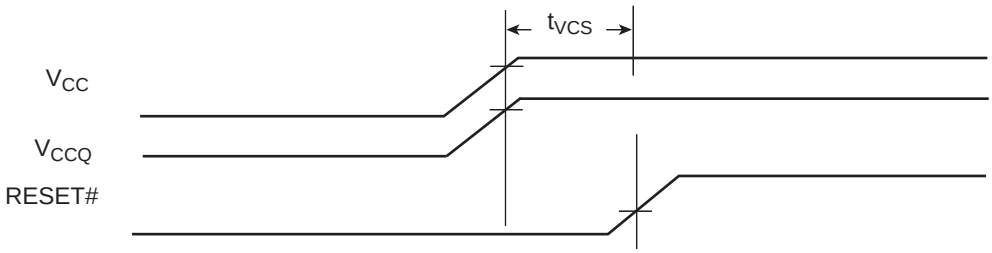
## 19.1 $V_{CC}$ Power-up

| Parameter | Description | Test Setup | Speed | Unit |
|---|---|---|---|---|
| $t_{VCS}$ | $V_{CC}$ Setup Time | Min | 1 | ms |

***Notes***

1. *$V_{CC}$ >+ $V_{CCQ}$ - 100 mV*
2. *$V_{CC}$ ramp rate is >1 V/100 µs.*

**Figure 19.1** $V_{CC}$ Power-up Diagram CLK Characterization

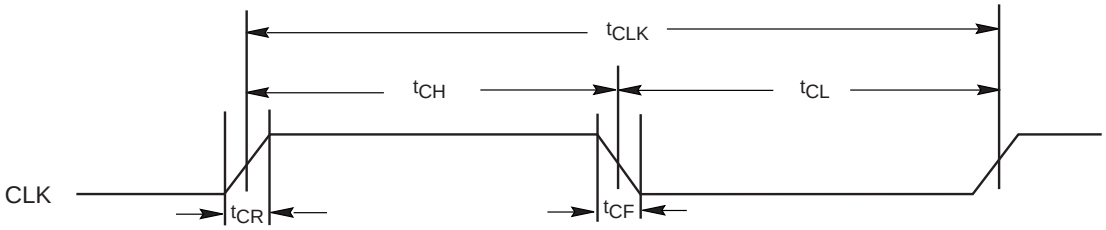



| Parameter | Description | | (66 MHz) | Unit |
|---|---|---|---|---|
| $f_{CLK}$ | CLK Frequency | Max | 66 | MHz |
| $t_{CLK}$ | CLK Period | Min | 15.0 | ns |
| $t_{CH}$ | CLK High Time | Min | 6.1 | ns |
| $t_{CL}$ | CLK Low Time | | | |
| $t_{CR}$ | CLK Rise Time | Max | 3 | ns |
| $t_{CF}$ | CLK Fall Time | | | |

***Notes***

1. *Clock jitter of +/- 5% permitted.*
2. *Not 100% tested.*

**Figure 19.2** CLK Characterization



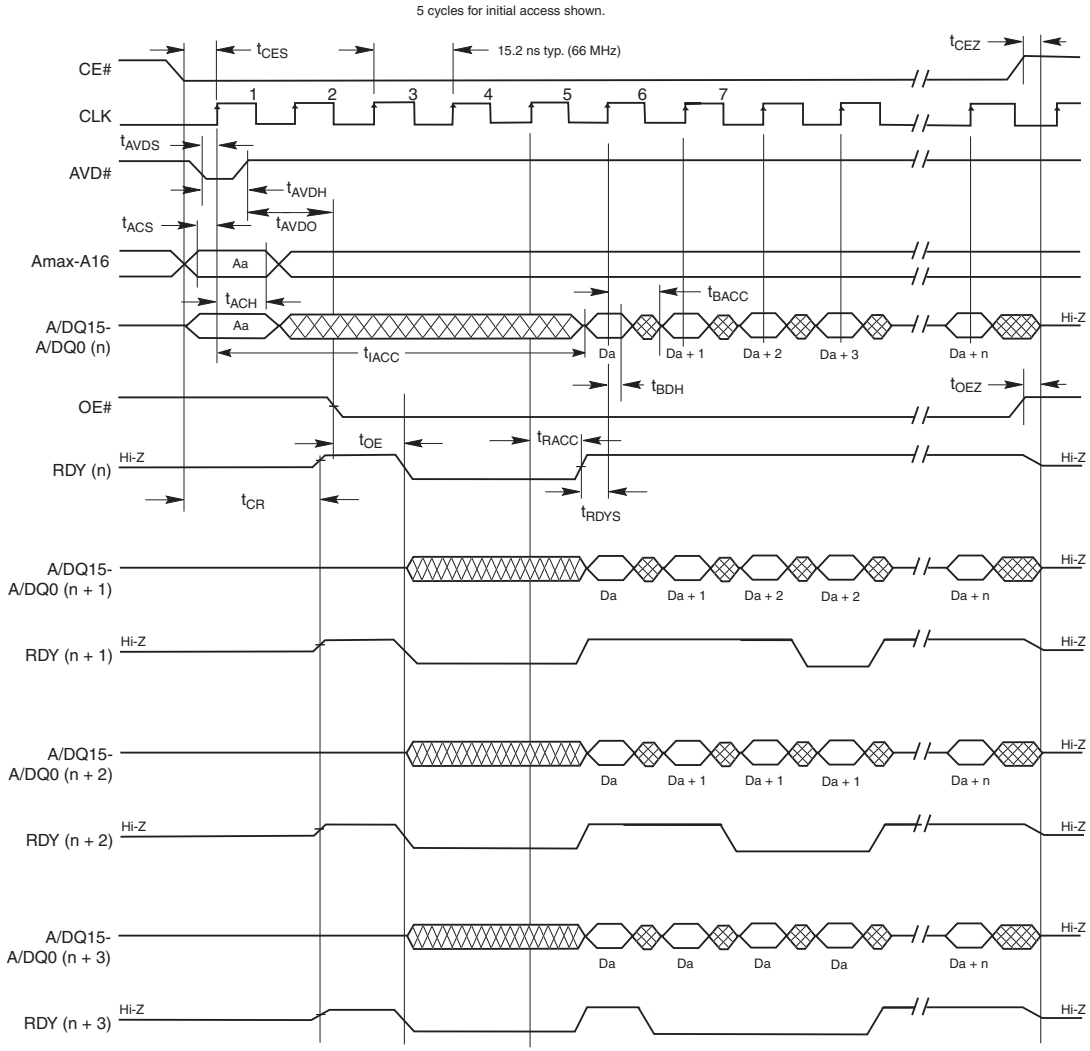
## 19.2 Synchronous/Burst Read

| Parameter | | Description | | (66 MHz) | (83 MHz) | (104 MHz) | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Standard | | | | | | |
| | $t_{IACC}$ | Initial Access Time | Max | 80 | | | ns |
| | $t_{BACC}$ | Burst Access Time Valid Clock to Output Delay | Max | 11.2 | 9 | 7.6 | ns |
| | $t_{AVDS}$ | AVD# Setup Time to CLK | Min | 4 | | | ns |
| | $t_{AVDH}$ | AVD# Hold Time from CLK | Min | 3 | | | ns |
| | $t_{AVDO}$ | AVD# High to OE# Low | Min | 0 | | | ns |
| | $t_{ACS}$ | Address Setup Time to CLK | Min | 4 | | | ns |
| | $t_{ACH}$ | Address Hold Time from CLK | Min | 6 | | | ns |
| | $t_{BDH}$ | Data Hold Time from Next Clock Cycle | Min | 3 | 3 | 2 | ns |
| | $t_{OE}$ | Output Enable to Data, or RDY Valid | Max | 15 | | | ns |
| | $t_{CEZ}$ | Chip Enable to High Z (1) | Max | 10 | | | ns |
| | $t_{OEZ}$ | Output Enable to High Z (1) | Max | 10 | | | ns |
| | $t_{CES}$ | CE# Setup Time to CLK | Min | 4 | | | ns |
| | $t_{RDYS}$ | RDY Setup Time to CLK | Min | 3.9 | 3 | 2 | ns |
| | $t_{RACC}$ | Ready access time from CLK | Max | 11.2 | 9 | 7.6 | ns |

***Note***

*1. Not 100% tested.*

**Figure 19.3** Burst Mode Read




***Notes***

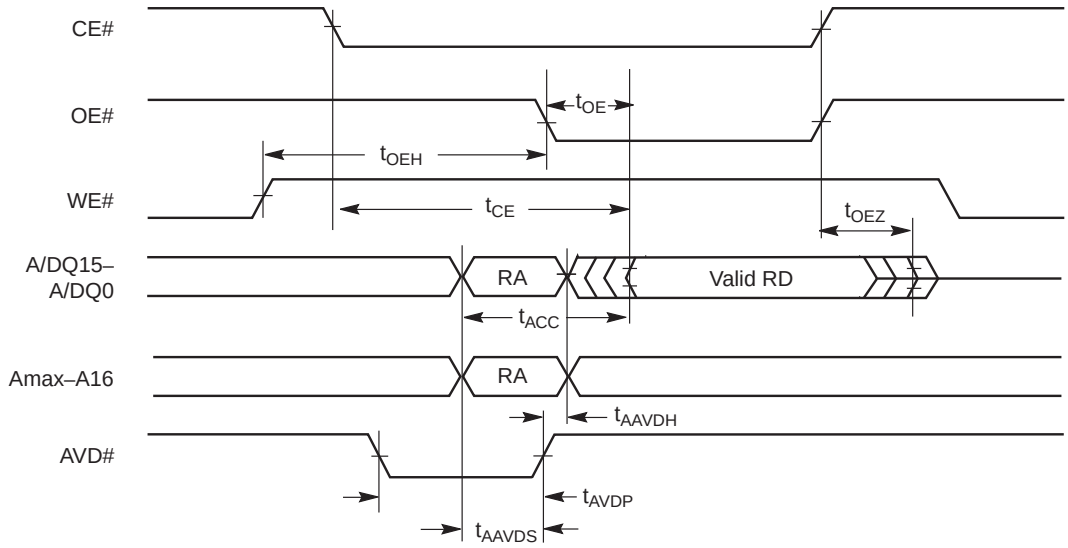
1. *Figure shows total number of clock set to five.*
2. *If any burst address occurs at “address + 1”, “address + 2”, or “address +3”, additional clock delays are inserted, and are indicated by RDY.*

## 19.3 Asynchronous Read

| Parameter | | Description | | | | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Standard | | | | | |
| | $t_{CE}$ | Access Time from CE# Low | | Max | 80 | ns |
| | $t_{ACC}$ | Asynchronous Access Time | | Max | 80 | ns |
| | $t_{AVDP}$ | AVD# Low Time | | Min | 6 | ns |
| | $t_{AAVDS}$ | Address Setup Time to Rising Edge of AVD | | Min | 4 | ns |
| | $t_{AAVDH}$ | Address Hold Time from Rising Edge of AVD | | Min | 3.5 | ns |
| | $t_{OE}$ | Output Enable to Output Valid | | Max | 15 | ns |
| | $t_{OEH}$ | Output Enable Hold Time | Read | Min | 0 | ns |
| | | | Toggle and Data# Polling | Min | 10 | ns |
| | $t_{OEZ}$ | Output Enable to High Z (1) | | Max | 10 | ns |

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

**Figure 19.4** Asynchronous Mode Read



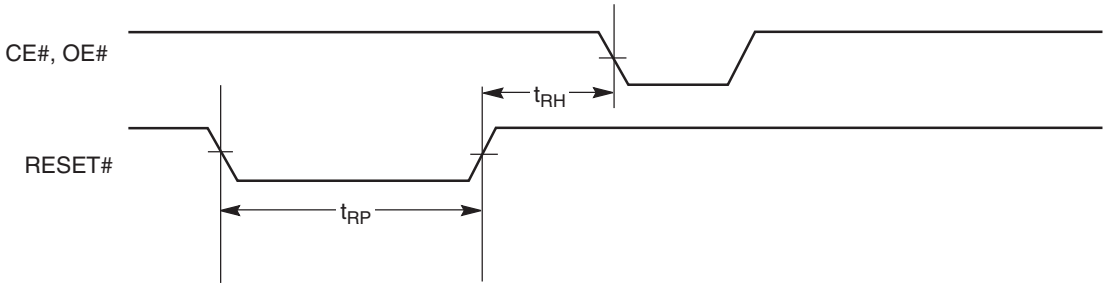

***Note***
*RA = Read Address, RD = Read Data.*

## 19.4 Hardware Reset (RESET#)

| Parameter | | Description | | All Speed Options | Unit |
|---|---|---|---|---|---|
| JEDEC | Std | | | | |
| | $t_{RP}$ | RESET# Pulse Width | Min | 200 | ns |
| | $t_{RH}$ | Reset High Time Before Read | Min | 10 | µs |

***Note***
*Not 100% tested*

**Figure 19.5 Reset Timings**



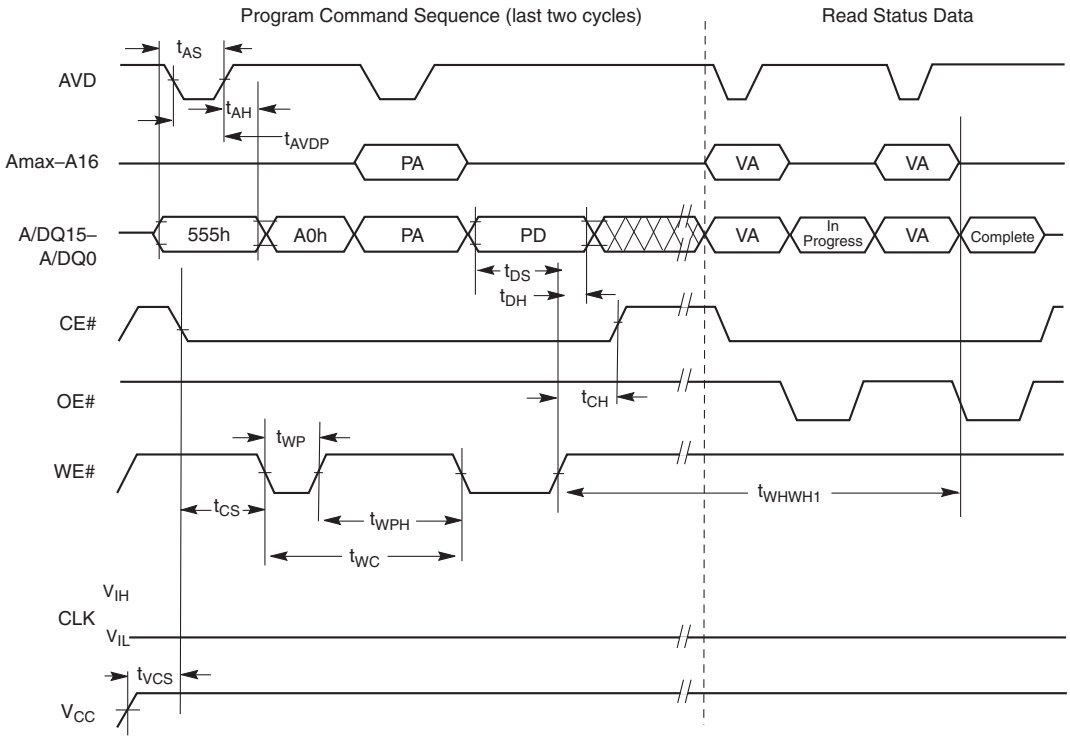

## 19.5 Erase/Program Operations

| Parameter | | Description | | (66 MHz) | Unit |
|---|---|---|---|---|---|
| JEDEC | Standard | | | | |
| $t_{AVAV}$ | $t_{WC}$ | Write Cycle Time (1) | Min | 60 | ns |
| $t_{AVWL}$ | $t_{AS}$ | Address Setup Time | Min | 4 | ns |
| $t_{WLAX}$ | $t_{AH}$ | Address Hold Time | Min | 3.5 | ns |
| | $t_{AVDP}$ | AVD# Low Time | Min | 6 | ns |
| | $t_{AAVDS}$ | Address Setup to Rising of AVD# | Min | 4 | ns |
| | $t_{AAVDH}$ | Address Hold from Rising of AVD# | Min | 3.5 | ns |
| $t_{DVWH}$ | $t_{DS}$ | Data Setup Time | Min | 20 | ns |
| $t_{WHDX}$ | $t_{DH}$ | Data Hold Time | Min | 0 | ns |
| $t_{GHWL}$ | $t_{GHWL}$ | Read Recovery Time Before Write | Typ | 0 | ns |
| $t_{ELWL}$ | $t_{CS}$ | CE# Setup Time to WE# | Typ | 8 | ns |
| $t_{WHEH}$ | $t_{CH}$ | CE# Hold Time | Typ | 0 | ns |
| $t_{WLWH}$ | $t_{WP}$/$t_{WRL}$ | Write Pulse Width | Typ | 30 | ns |
| | $t_{VLWH}$ | AVD# Disable to WE# Disable | Min | 23.5 | ns |
| | $t_{WEA}$ | WE# Disable to AVD# Enable | Min | 9.6 | ns |
| | $t_{CR}$ | CE# Low to RDY Valid | Min | 10 | ns |
| | $t_{CEZ}$ | CE# Disable to Output High Z | Min | 10 | ns |
| | $t_{WEH}$ | OE# Disable to WE# Enable | Min | 4 | ns |
| $t_{WHWL}$ | $t_{WPH}$ | Write Pulse Width High | Typ | 20 | ns |
| | $t_{SR/W}$ | Latency Between Read and Write Operations | Min | 0 | ns |
| | $t_{VPP}$ | $V_{PP}$ Rise and Fall Time | Min | 500 | ns |
| | $t_{VPS}$ | $V_{PP}$ Setup Time (During Accelerated Programming) | Min | 1 | µs |
| | $t_{VCS}$ | $V_{CC}$ Setup Time | Min | 50 | µs |
| | $t_{ESL}$ | Erase Suspend Latency | Max | 30 | µs |
| | $t_{PSL}$ | Program Suspend Latency | Max | 30 | µs |
| | $t_{PSP}$ | Toggle Time During Programming Within a Protected Sector | Typ | 20 | µs |
| | $t_{ASP}$ | Toggle Time During Sector Protection | Typ | 20 | µs |
| | $t_{WEP}$ | Noise Pulse Margin on WE# | Max | 3 | ns |
| | $t_{ERS}$ | ER to ES | Min | 30 | µs |
| | $t_{PRS}$ | PR to PS | Min | 30 | µs |

***Notes***

1. *Not 100% tested.*
2. *See the Erase and Programming Performance section for more information.*
3. *Does not include the preprogramming time.*

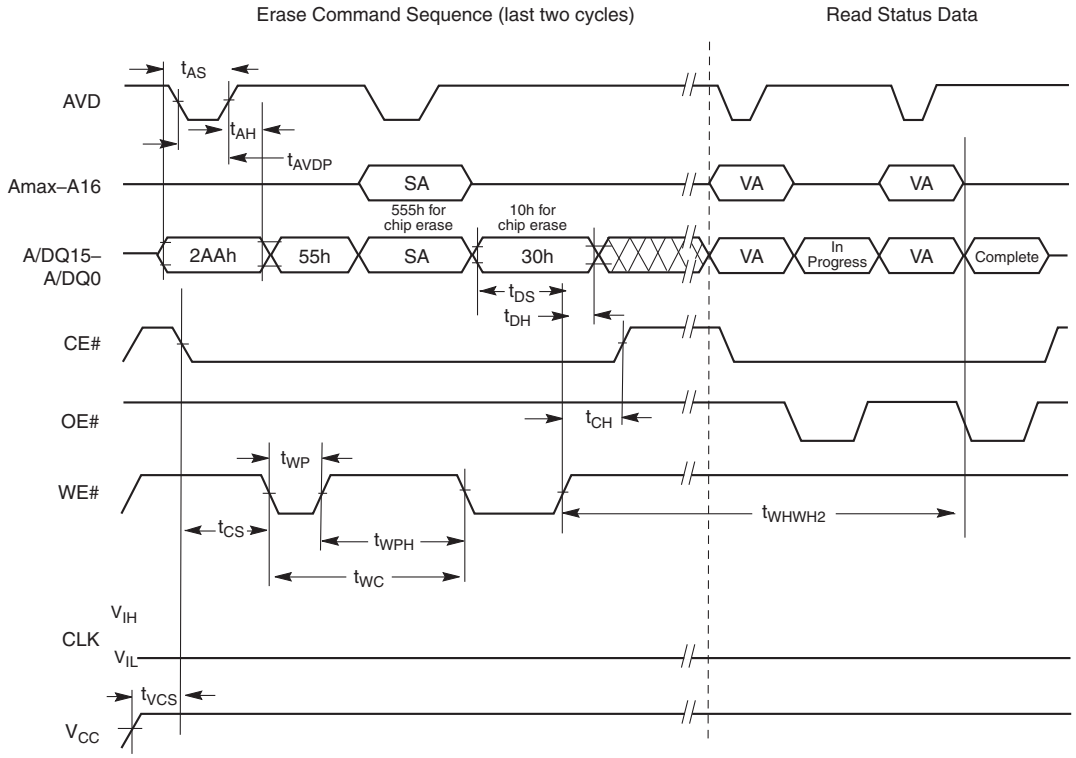
**Figure 19.6** Program Operation Timings




***Notes***

1. *PA = Program Address, PD = Program Data, VA = Valid Address for reading status bits.*
2. *"In progress" and "complete" refer to status of program operation.*
3. *$A_{max}$–A16 are don't care during command sequence unlock cycles.*

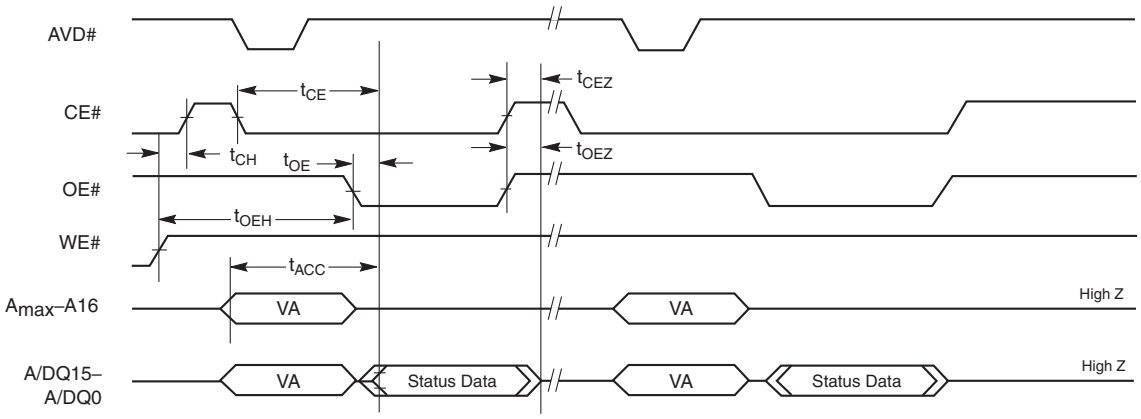
**Figure 19.7** Chip/Sector Erase Operations




***Notes***

1. *SA is the sector address for Sector Erase.*
2. *Address bits $A_{max}$–A16 are don't cares during unlock cycles in the command sequence.*

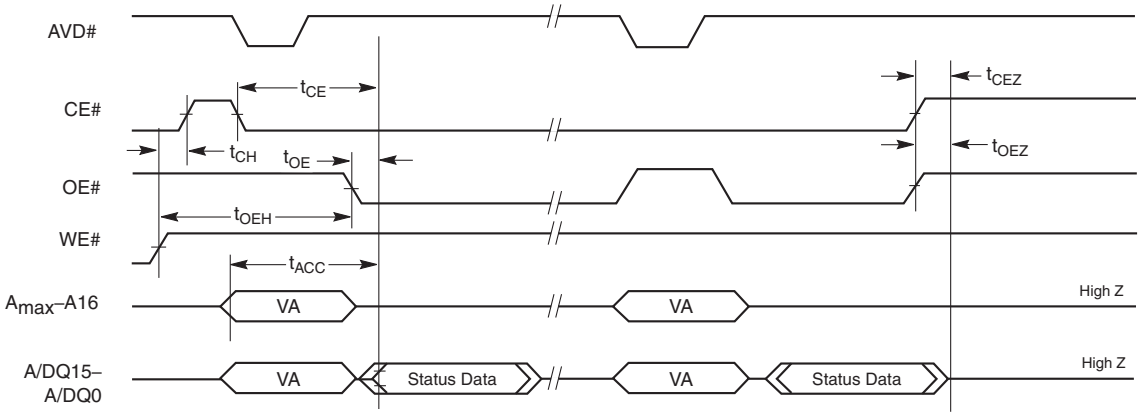
**Figure 19.8** Data# Polling Timings (During Embedded Algorithm)




***Notes***

1. *All status reads are asynchronous.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, and Data# Polling will output true data.*

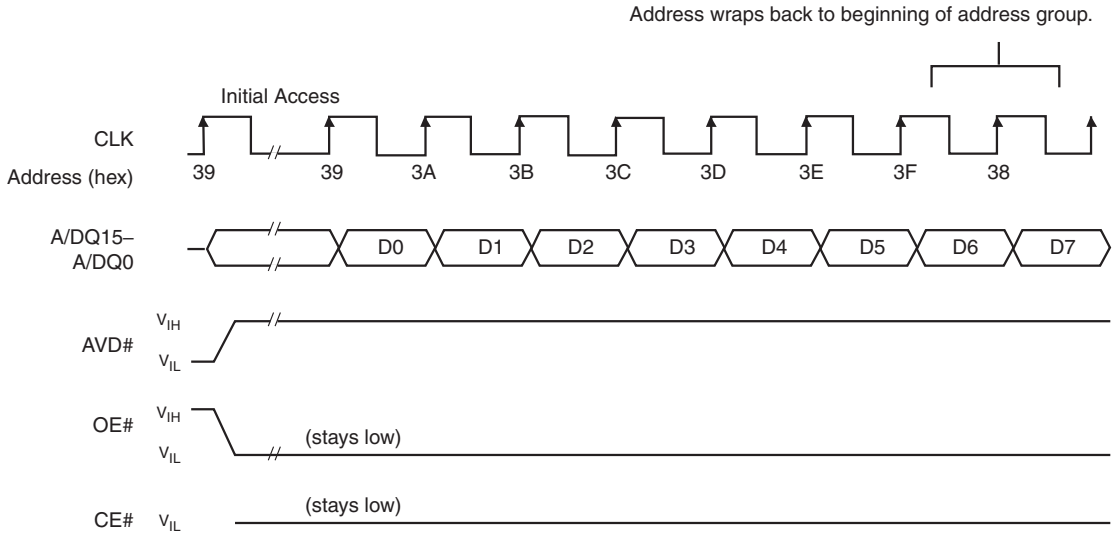
**Figure 19.9** Toggle Bit Timings (During Embedded Algorithm)




***Notes***

1. *All status reads are asynchronous.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, the toggle bits will stop toggling.*

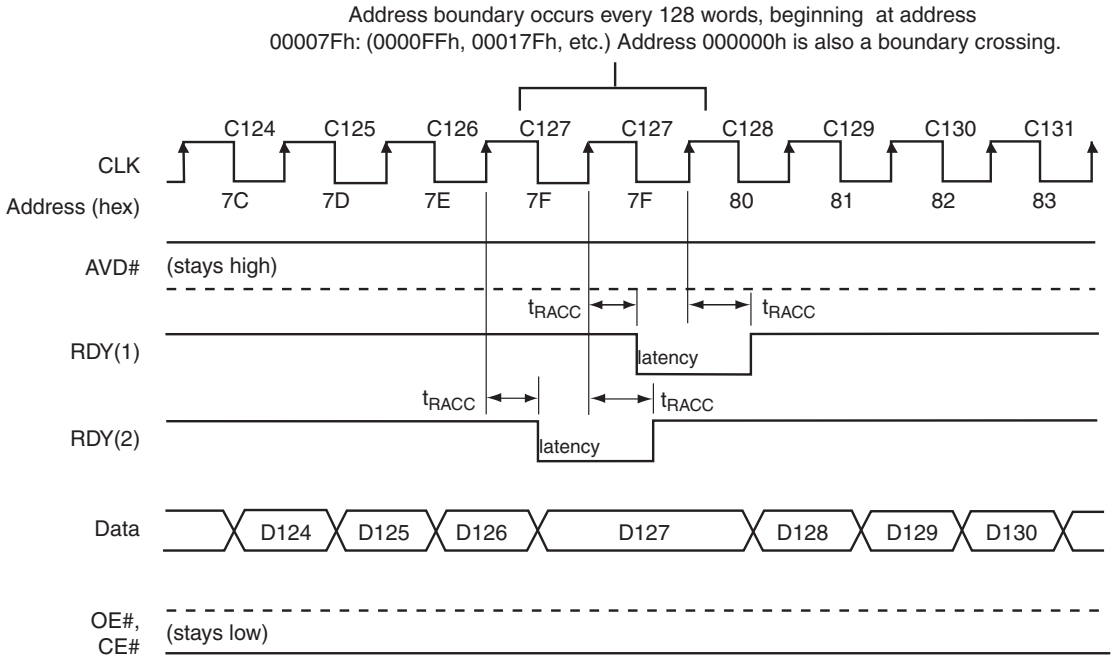
**Figure 19.10** 8- and 16-Word Linear Burst Address Wrap Around




***Note***

*8-word linear burst mode shown. 16-word linear burst read mode behaves similarly. D0 represents the first word of the linear burst.*

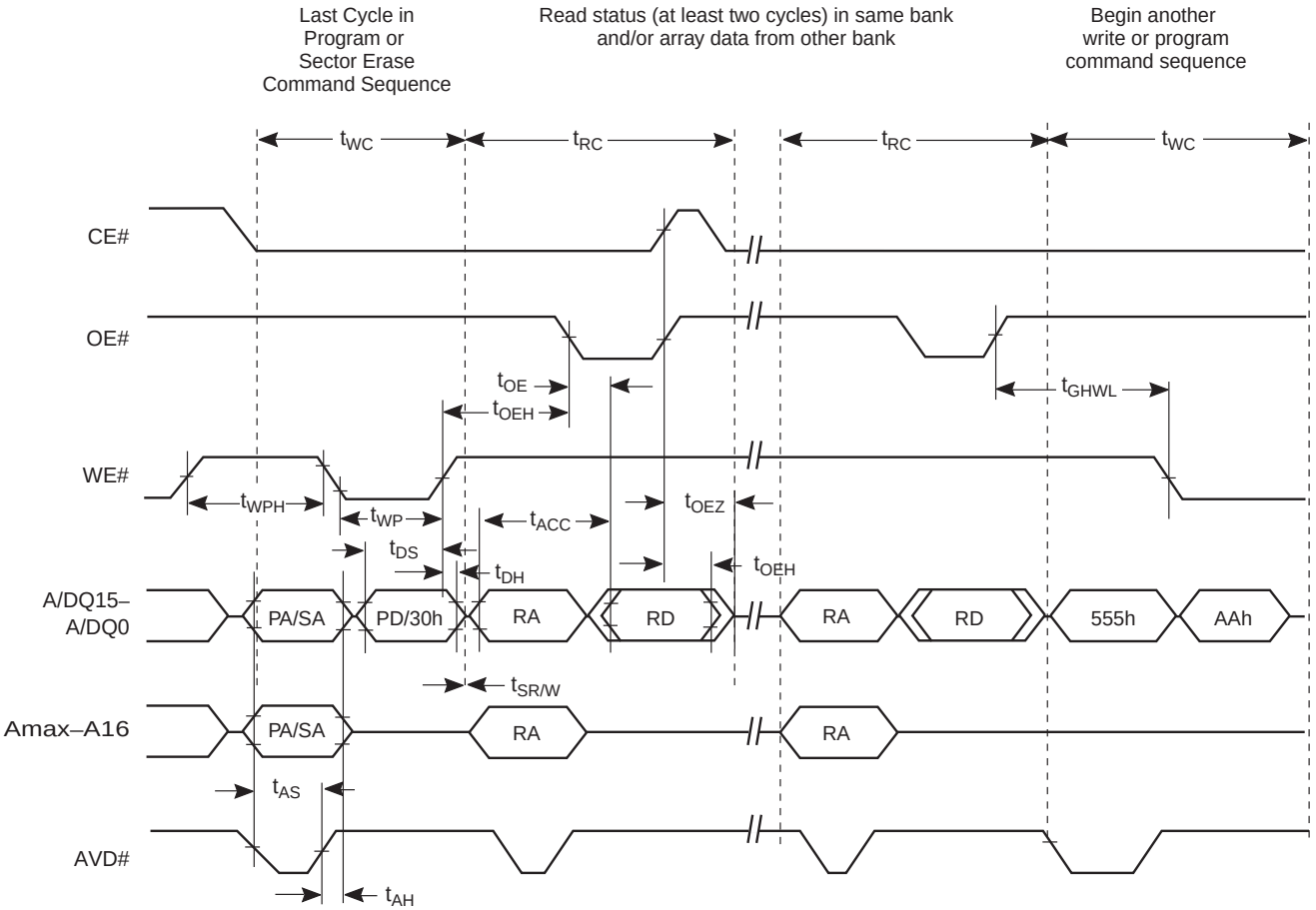
**Figure 19.11** Latency with Boundary Crossing




***Notes***

1. *Cxx indicates the clock that triggers data Dxx on the outputs; for example, C60 triggers D60.*
2. *At frequencies less than or equal to 66 Mhz, there is no latency.*

**Figure 19.12** Back-to-Back Read/Write Cycle Timings




***Note***
*Breakpoints in waveforms indicate that system may alternately read array data from the "non-busy bank" while checking the status of the program or erase operation in the "busy" bank. The system should read status twice to ensure valid information.*

# 20. Erase and Programming Performance

| Parameter | | | Typ (1) | Max (2) | Unit | Comments |
|---|---|---|---|---|---|---|
| Sector Erase Time | 32 Kword | $V_{CC}$ | 0.8 | 3.5 | s | Excludes 00h programming prior to erasure (Note 5) |
| | 8 Kword | $V_{CC}$ | 0.35 | 2 | | |
| Chip Erase Time | | $V_{CC}$ | 103 (VS/XS064R); 52 (VS/XS032R) | 453 (VS/XS064R); 227 (VS/XS032R) | | |
| | | $V_{PP}$ | 103 (VS/XS064R); 52 (VS/XS032R) | 453 (VS/XS064R); 227 (VS/XS032R) | | |
| Single Word Programming Time | | $V_{CC}$ | 130 | 400 | µs | |
| Effective Word Programming Time utilizing Program Write Buffer | | $V_{CC}$ | 9.4 | 94 | | |
| | | $V_{PP}$ | 6 | 60 | | |
| Total 32-Word Buffer Programming Time | | $V_{CC}$ | 300 | 3000 | | |
| | | $V_{PP}$ | 192 | 1920 | | |
| Chip Programming Time (4) | | $V_{CC}$ | 39.3 (VS/XS064R); 20 (VS/XS032R) | 78.6 (VS/XS064R); 39.3 (VS/XS032R) | s | Excludes system level overhead (Note 6) |
| | | $V_{PP}$ | 26 (VS/XS064R); 13 (VS/XS032R) | 52 (VS/XS064R); 26 (VS/XS032R) | | |

***Notes***

1. *Typical program and erase times assume the following conditions: 25°C, 1.8 V $V_{CC}$, 10,000 cycles typical. Additionally, programming typicals assume checkerboard pattern.*
2. *Under worst case conditions of 90°C, $V_{CC}$ = 1.70 V, 100,000 cycles.*
3. *Effective write buffer specification is based upon a 32-word write buffer operation.*
4. *The typical chip programming time is considerably less than the maximum chip programming time listed, since most words program faster than the maximum program times listed.*
5. *In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 00h before erasure.*
6. *System-level overhead is the time required to execute the two- or four-bus-cycle sequence for the program command. See Table 12.3 on page 37 for further information on command definitions.*

# 21. BGA Ball Capacitance

| Parameter Symbol | Parameter Description | Test Setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0 | 4.2 | 5.0 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0 | 5.4 | 6.5 | pF |
| $C_{IN2}$ | Control Pin Capacitance | $V_{IN}$ = 0 | 3.9 | 4.7 | pF |

***Notes***

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

# 22. Revision History

**Spansion Publication Number:** S29VS_XS064R_032R_00

| Section | Description |
|---|---|
| **Revision 01 (August 5, 2008)** | |
| | Initial release |

## Document History Page

| **Document Title: S29VS064R, S29VS032R, S29XS064R, S29XS032R 64/32 Mbit (4/2M x 16-bit), 1.8 V MirrorBit Flash<br>Document Number: 002-01468** | | | | |
|---|---|---|---|---|
| **Rev.** | **ECN No.** | **Orig. of Change** | **Submission Date** | **Description of Change** |
| ** | – | WIOB | 08/05/2008 | Initial release |
| *A | 4968291 | WIOB | 10/19/2015 | Updated to Cypress template |
| *B | 5967631 | AESATMP8 | 11/15/2017 | Updated logo and Copyright. |

## 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

### Cypress Developer Community

Forums | WICED IOT Forums | Projects | Video | Blogs | Training | Components

### Technical Support

cypress.com/support

© Cypress Semiconductor Corporation, 2008-2017. 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. 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.