"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 contact your local sales office for additional information about Cypress products and solutions.

ADVANCE

S29WS064R

64 Mbit (4M x 16-bit), 1.8 V MirrorBit® Flash

Distinctive Characteristics

- Single 1.8-Volt read, program and erase (1.70V - 1.95V)
- 65 nm MirrorBit process technology
- VersatileIO™ Feature
  - Device generates data output voltages and tolerates data input voltages as determined by the voltage on the V_{IO} pin
  - 1.8 V compatible I/O signals
- 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-word/16-word linear burst with wrap around
- Asynchronous Page Mode
  - 8-word page
  - Page access time of 20 ns
- 32-word write buffer reduces overall programming time for multiple-word updates
- Sector Architecture
  - Four 8-kword sectors in upper most address range
  - One hundred twenty seven 32-kwords sectors
  - Four banks
  - Top or Bottom boot sector configuration
- 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
- Command set compatible with JEDEC (42.4) standard
- 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 ACC input is V_{IL}
  - Low V_{CC} write inhibit
- Handshaking feature
  - Provides host system with minimum possible latency by monitoring RDY
- Supports Common Flash Memory Interface (CFI)
- 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
- Suspend and Resume commands for Program and Erase operations
- Synchronous or Asynchronous program operation, independent of burst control register settings
- ACC input pin to reduce factory programming time
- Offered Packages
  - 84-ball FBGA (8 mm x 11.6 mm)

Performance Characteristics

Read Access Times	
Speed Option (MHz)	**108**
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. Asynch. Page Access Time, ns (t_{PACC})	20
Max CE# Access Time, ns (t_{CE})	80
Max OE# Access Time, ns (t_{OE})	13.5

Current Consumption (typical values)	
Continuous Burst Read @ 108 MHz	32 mA
Simultaneous Operation @ 108 MHz	71 mA
Program/Erase	30 mA
Standby Mode (asynchronous)	20 µA

Typical Program and Erase Times	
Single Word Programming	170 µs
Effective Write Buffer Programming (V_{CC}) Per Word	14.1 µs
Effective Write Buffer Programming (V_{ACC}) Per Word	9.0 µs
Sector Erase (8 kword Sector)	350 ms
Sector Erase (32 kword Sector)	800 ms

Contents

1. General Description

The Spansion S29WS064R is a 64 Megabit 1.8 Volt-only MirrorBit Flash memory organized as 4,194,304 words of 16 bits each. This burst mode Flash device is capable of performing simultaneous read and write operations with zero latency on two separate banks using separate data and address pins. This device can operate up to 108 MHz and uses a single V_{CC} of 1.7V to 1.95V to read, program, and erase the memory array, making it ideal for today's demanding applications requiring higher density, better performance and lowered power consumption. A 9.0-volt ACC may be used for faster program performance if desired. This device can also be programmed in standard EPROM programmers.

The device operates within the temperature range of -25°C to +85°C, and is offered in a Very Thin FBGA package. The device is also available in the temperature range of -40°C to +85°C. Please refer to the Specification Supplement with Publication Number S29WS064R_SP for specification differences for devices offered in the -45°C to +85°C temperature range.

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 devices are structured as shown in the following tables:

Device Structure (Top Boot)

S29WS064R		
Bank	**Sector Size**	**Sector Count**
0	32 kwords	32
1	32 kwords	32
2	32 kwords	32
3	32 kwords	31
	8 kwords	4

Device Structure (Bottom Boot)

S29WS064R		
Bank	**Sector Size**	**Sector Count**
0	8 kwords	4
	32 kwords	31
1	32 kwords	32
2	32 kwords	32
3	32 kwords	32

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_{IO} pin.

The device uses Chip Enable (CE#), Write Enable (WE#), Address Valid (AVD#) and Output Enable (OE#) to control asynchronous read and write operations. For burst operations, the device additionally requires Ready (RDY) and Clock (CLK). This implementation allows easy interface with minimal glue logic to microprocessors/micro controllers 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 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 two types of data protection at the sector level. When ACC is at V_{IL}, the entire flash memory array is protected. Dynamic Sector Protection provides in-system, command-enabled protection of any combination of sectors using a single power supply at V_{CC}.

The device offers 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 device. 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 a 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), 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.

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 simultaneously via Fowler-Nordheim tunneling. The data is programmed using hot electron injection.

2. Input/Output Descriptions and Logic Symbol

Table 2.1 identifies the input and output package connections provided on the device.

Table 2.1 Input/Output Descriptions

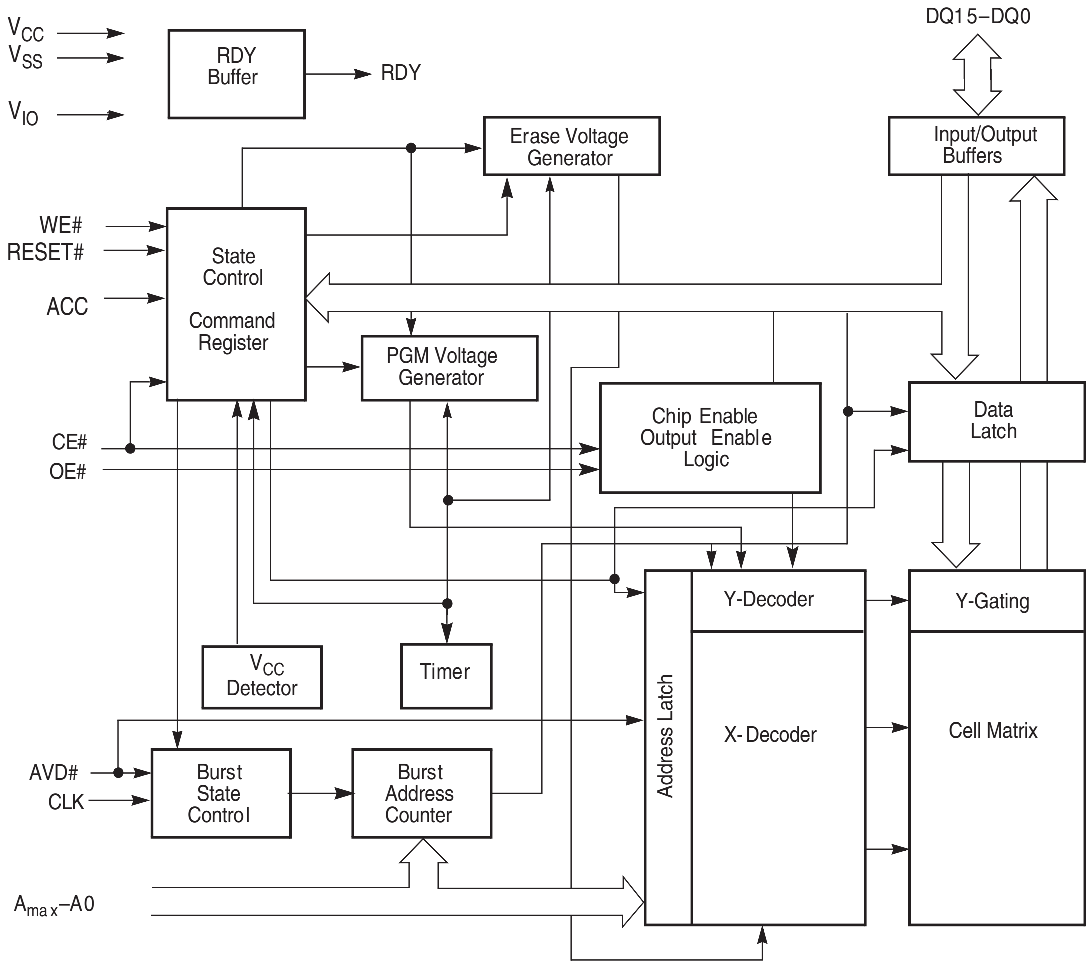
Symbol	Type	Description
A21-A0	Input	Address Inputs.
DQ15-DQ0	I/O	Data input/output.
CE#	Input	Chip Enable. Asynchronous relative to CLK for Burst Mode.
OE#	Input	Output Enable. Asynchronous relative to CLK for Burst Mode.
WE#	Input	Write Enable.
V_{CC}	Supply	Device Power Supply.
V_{IO}	Supply	Versatile IO Input
V_{SS}	Supply	Ground.

Table 2.1 Input/Output Descriptions

Symbol	Type	Description
RDY	Output	Ready. Indicates when valid burst data is ready to be read.
CLK	Input	Clock Input. In burst mode, after the initial word is output, subsequent active edges of CLK increment the internal address counter.
AVD#	Input	Address Valid. Indicates to device that the valid address is present on the address inputs. When low during asynchronous mode, indicates valid address; when low during burst mode, causes starting address to be latched at the next active clock edge. When high, device ignores address inputs.
RESET#	Input	Hardware Reset. Low = device resets and returns to reading array data.
ACC	Input	Acceleration Input. At V_{HH}, accelerates programming. At V_{IL}, disables all program and erase functions. Should be at V_{IH} for all other conditions.
DNU	Do Not Use	A device internal signal may be connected to the package connector. The connection may be used by Spansion for test or other purposes and is not intended for connection to any host system signal. Any DNU signal related function will be inactive when the signal is at V_{IL}. The signal has an internal pull-down resistor and may be left unconnected in the host system or may be tied to V_{SS}. Do not use these connections for PCB signal routing channels. Do not connect any host system signal to these connections.
NC	Not Connected	No device internal signal is connected to the package connector nor is there any future plan to use the connector for a signal. The connection may safely be used for routing space for a signal on a Printed Circuit Board (PCB)
RFU	Reserved for Future Use	No device internal signal is currently connected to the package connector but there is potential future use for the connector for a signal. It is recommended to not use RFU connectors for PCB routing channels so that the PCB may take advantage of future enhanced features in compatible footprint devices.

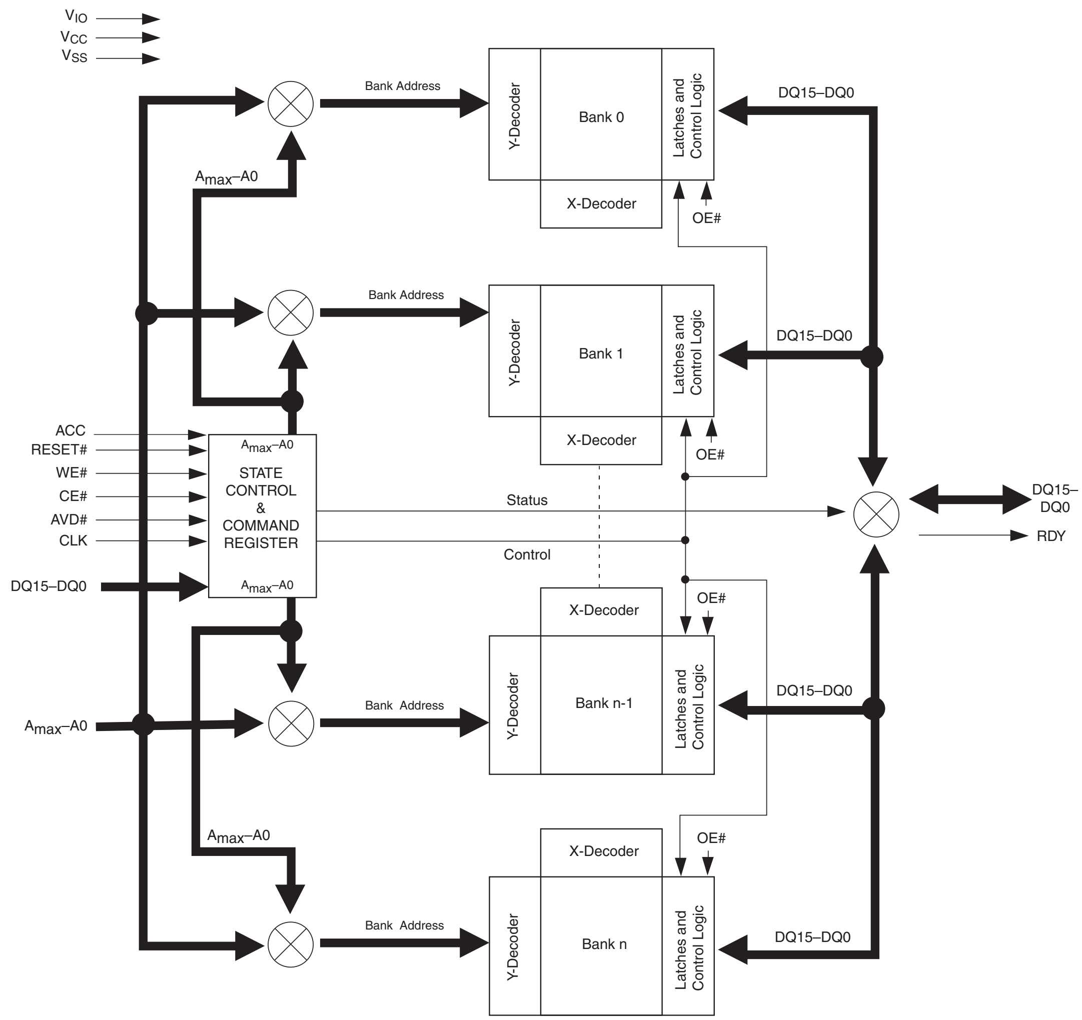
3. Block Diagrams

Figure 3.1 S29WS064R Block Diagram



Notes:
1. A_{max} indicates the highest order address bit. A_{max} equals A21 for S29WS064R.

Figure 3.2 Block Diagram of Simultaneous Operation Circuit



Notes:
1. *A_{max} indicates the highest order address bit. A_{max} equals A21 for WS064R.*
2. *n = 3*

4. Physical Dimensions/Connection Diagrams

This section shows the I/O designations and package specifications.

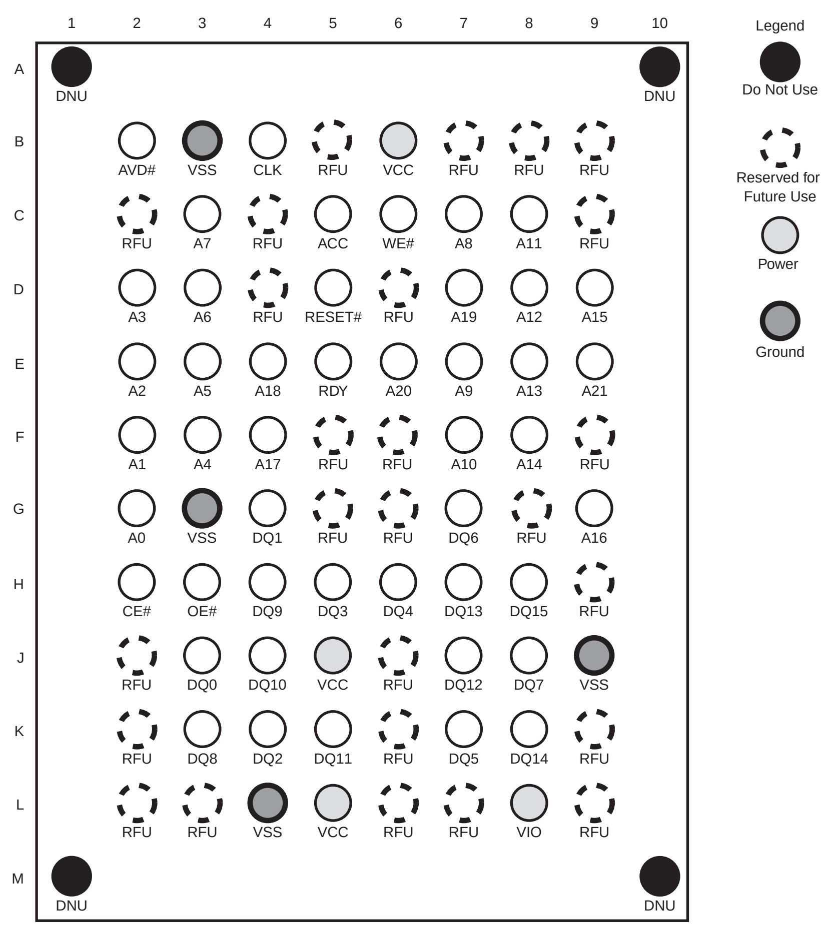
4.1 Special Handling Instructions for FBGA Package

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

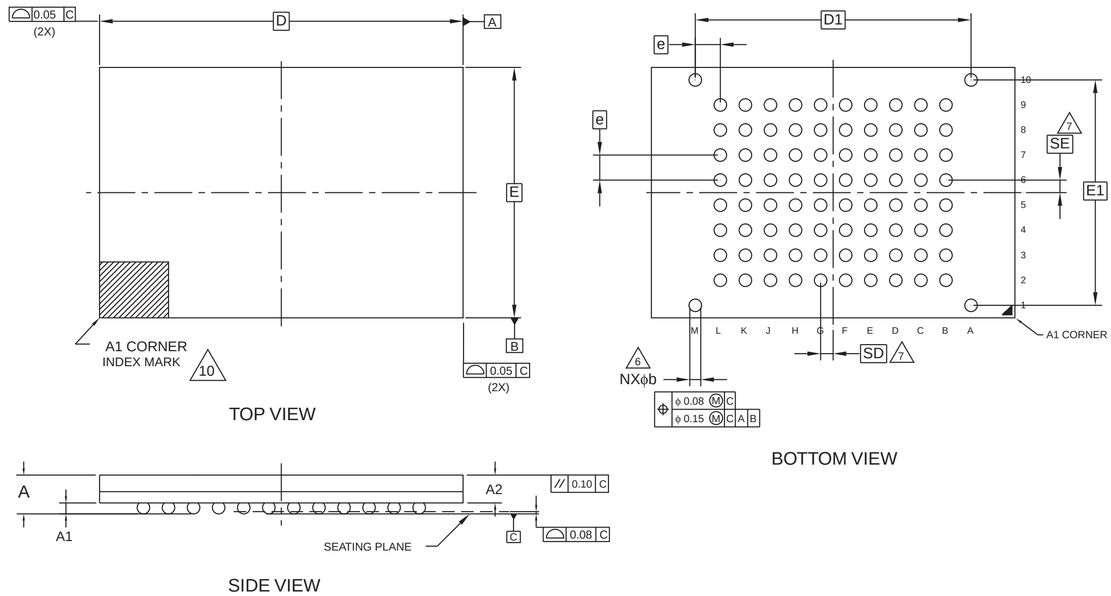
Flash memory devices in FBGA packages may be damaged if exposed to ultrasonic cleaning methods. The package and/or data integrity may be compromised if the package body is exposed to temperatures above 150°C for prolonged periods of time.

4.2 Connection Diagrams and Physical Dimensions

Figure 4.1 84-ball Fine-Pitch Ball Grid Array



VBH084—84-ball Fine-Pitch Ball Grid Array (FBGA) 11.6 x 8 mm Package



PACKAGE	VBH 084			
JEDEC	N/A			
	11.60 mm x 8.00 mm NOM PACKAGE			
SYMBOL	MIN	NOM	MAX	NOTE
A	---	---	1.00	OVERALL THICKNESS
A1	0.18	---	---	BALL HEIGHT
A2	0.62	---	0.76	BODY THICKNESS
D	11.60 BSC.			BODY SIZE
E	8.00 BSC.			BODY SIZE
D1	8.80 BSC.			BALL FOOTPRINT
E1	7.20 BSC.			BALL FOOTPRINT
MD	12			ROW MATRIX SIZE D DIRECTION
ME	10			ROW MATRIX SIZE E DIRECTION
N	84			TOTAL BALL COUNT
ϕb	0.33	---	0.43	BALL DIAMETER
e	0.80 BSC.			BALL PITCH
SD / SE	0.40 BSC.			SOLDER BALL PLACEMENT
?	(A2-A9, B10-L10, M2-M9, B1-L1)			DEPOPULATED SOLDER BALLS

NOTES:

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

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

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

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

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

3339 \ 16-038.25b

Note:

1. BSC is an ANSI standard for Basic Space Centering.

4.3 MCP Look-Ahead Ballout for Future Designs

Refer to the *Design-In Scalable Wireless Solutions with Spansion Products* application note, available on the web or through a Spansion sales office.

5. Ordering Information

The order number is formed by a valid combinations of the following:

S29WS 064 R AB BH W 00 0

Packing Type
0 = Tray (standard; (Note 1))
3 = 13-inch Tape and Reel

Model Number
00 = Top Boot (Note 2)
01 = Bottom Boot

Temperature Range
W = Wireless (–25°C to +85°C)

Package Type & Material Set
BH = Very Thin Fine-Pitch BGA, Low Halogen Lead (Pb)-free Package

Speed Option (Burst Frequency)
0P = 66 MHz
0S = 83 MHz
AB = 108 MHz

Process Technology
R = 65 nm MirrorBit Technology

Flash Density
064= 64 Mb

Product Family
S29WS =1.8 Volt-Only Simultaneous Read/Write, Burst Mode Flash Memory

5.1 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.

S29WS-R Valid Combinations (1), (3)						
Base Ordering Part Number	**Speed Option**	**Package Type and Material**	**Temperature Range**	**Model Number(s)**	**Packing Type**	**Package Type (3)**
S29WS064R	0P, 0S, AB	BH	W	00, 01	0, 3	11.6 mm x 8 mm 84-ball

Notes:
1. *Type 0 is standard. Specify other options as required.*
2. *If a choice exists, Spansion recommends Top Boot.*
3. *BGA package marking omits leading "S29" and packing type designator from ordering part number.*
4. *Industrial Temperature Range (-40°C to +85°C) is also available. For device specification differences, please refer to the Specification Supplement with Publication Number S29WS064R_SP.*

6. Product Overview

The S29WS064R is a 64 Megabit, 1.8 volt-only, simultaneous read/write burst mode Flash device optimized for today's designs that demand a large storage array, rich functionality, and low power consumption. This device is organized in 4 Mwords of 16 bits each and is capable of continuous, synchronous (burst) read. This product also offers single word programming or a 32-word buffer for programming with program and erase suspend functionality. Additional features include:

- Advanced Sector Protection methods for protecting sectors as required
- 256 words of Secured Silicon area for storing customer and factory secured information. The Secured Silicon Sector is One Time Programmable.

6.1 Memory Map

The S29WS064R device consists of 4 banks organized as shown in Table 6.1 and Table 6.2.

Table 6.1 S29WS064R Sector and Memory Address Map (Top Boot) (Sheet 1 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
0	32 kwords	SA000	0h	7FFFh
		SA001	8000h	FFFFh
		SA002	10000h	17FFFh
		SA003	18000h	1FFFFh
		SA004	20000h	27FFFh
		SA005	28000h	2FFFFh
		SA006	30000h	37FFFh
		SA007	38000h	3FFFFh
		SA008	40000h	47FFFh
		SA009	48000h	4FFFFh
		SA010	50000h	57FFFh
		SA011	58000h	5FFFFh
		SA012	60000h	67FFFh
		SA013	68000h	6FFFFh
		SA014	70000h	77FFFh
		SA015	78000h	7FFFFh
		SA016	80000h	87FFFh
		SA017	88000h	8FFFFh
		SA018	90000h	97FFFh
		SA019	98000h	9FFFFh
		SA020	A0000h	A7FFFh
		SA021	A8000h	AFFFFh
		SA022	B0000h	B7FFFh
		SA023	B8000h	BFFFFh
		SA024	C0000h	C7FFFh
		SA025	C8000h	CFFFFh
		SA026	D0000h	D7FFFh
		SA027	D8000h	DFFFFh
		SA028	E0000h	E7FFFh
		SA029	E8000h	EFFFFh
		SA030	F0000h	F7FFFh
		SA031	F8000h	FFFFFh

Table 6.1 S29WS064R Sector and Memory Address Map (Top Boot) (Sheet 2 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
1	32 kwords	SA032	100000h	107FFFh
		SA033	108000h	10FFFFh
		SA034	110000h	117FFFh
		SA035	118000h	11FFFFh
		SA036	120000h	127FFFh
		SA037	128000h	12FFFFh
		SA038	130000h	137FFFh
		SA039	138000h	13FFFFh
		SA040	140000h	147FFFh
		SA041	148000h	14FFFFh
		SA042	150000h	157FFFh
		SA043	158000h	15FFFFh
		SA044	160000h	167FFFh
		SA045	168000h	16FFFFh
		SA046	170000h	177FFFh
		SA047	178000h	17FFFFh
		SA048	180000h	187FFFh
		SA049	188000h	18FFFFh
		SA050	190000h	197FFFh
		SA051	198000h	19FFFFh
		SA052	1A0000h	1A7FFFh
		SA053	1A8000h	1AFFFFh
		SA054	1B0000h	1B7FFFh
		SA055	1B8000h	1BFFFFh
		SA056	1C0000h	1C7FFFh
		SA057	1C8000h	1CFFFFh
		SA058	1D0000h	1D7FFFh
		SA059	1D8000h	1DFFFFh
		SA060	1E0000h	1E7FFFh
		SA061	1E8000h	1EFFFFh
		SA062	1F0000h	1F7FFFh
		SA063	1F8000h	1FFFFFh

Table 6.1 S29WS064R Sector and Memory Address Map (Top Boot) (Sheet 3 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
2	32 kwords	SA064	200000h	207FFFh
		SA065	208000h	20FFFFh
		SA066	210000h	217FFFh
		SA067	218000h	21FFFFh
		SA068	220000h	227FFFh
		SA069	228000h	22FFFFh
		SA070	230000h	237FFFh
		SA071	238000h	23FFFFh
		SA072	240000h	247FFFh
		SA073	248000h	24FFFFh
		SA074	250000h	257FFFh
		SA075	258000h	25FFFFh
		SA076	260000h	267FFFh
		SA077	268000h	26FFFFh
		SA078	270000h	277FFFh
		SA079	278000h	27FFFFh
		SA080	280000h	287FFFh
		SA081	288000h	28FFFFh
		SA082	290000h	297FFFh
		SA083	298000h	29FFFFh
		SA084	2A0000h	2A7FFFh
		SA085	2A8000h	2AFFFFh
		SA086	2B0000h	2B7FFFh
		SA087	2B8000h	2BFFFFh
		SA088	2C0000h	2C7FFFh
		SA089	2C8000h	2CFFFFh
		SA090	2D0000h	2D7FFFh
		SA091	2D8000h	2DFFFFh
		SA092	2E0000h	2E7FFFh
		SA093	2E8000h	2EFFFFh
		SA094	2F0000h	2F7FFFh
		SA095	2F8000h	2FFFFFh

Table 6.1 S29WS064R Sector and Memory Address Map (Top Boot) (Sheet 4 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
3	32 kwords	SA096	300000h	307FFFh
		SA097	308000h	30FFFFh
		SA098	310000h	317FFFh
		SA099	318000h	31FFFFh
		SA100	320000h	327FFFh
		SA101	328000h	32FFFFh
		SA102	330000h	337FFFh
		SA103	338000h	33FFFFh
		SA104	340000h	347FFFh
		SA105	348000h	34FFFFh
		SA106	350000h	357FFFh
		SA107	358000h	35FFFFh
		SA108	360000h	367FFFh
		SA109	368000h	36FFFFh
		SA110	370000h	377FFFh
		SA111	378000h	37FFFFh
		SA112	380000h	387FFFh
		SA113	388000h	38FFFFh
		SA114	390000h	397FFFh
		SA115	398000h	39FFFFh
		SA116	3A0000h	3A7FFFh
		SA117	3A8000h	3AFFFFh
		SA118	3B0000h	3B7FFFh
		SA119	3B8000h	3BFFFFh
		SA120	3C0000h	3C7FFFh
		SA121	3C8000h	3CFFFFh
		SA122	3D0000h	3D7FFFh
		SA123	3D8000h	3DFFFFh
		SA124	3E0000h	3E7FFFh
		SA125	3E8000h	3EFFFFh
		SA126	3F0000h	3F7FFFh
	8 kwords	SA127	3F8000h	3F9FFFh
		SA128	3FA000h	3FBFFFh
		SA129	3FC000h	3FDFFFh
		SA130	3FE000h	3FFFFFh

Table 6.2 S29WS064R Sector and Memory Address Map (Bottom Boot) (Sheet 1 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
0	8 kwords	SA000	0h	1FFFh
		SA001	2000h	3FFFh
		SA002	4000h	5FFFh
		SA003	6000h	7FFFh
	32 kwords	SA004	8000h	FFFFh
		SA005	10000h	17FFFh
		SA006	18000h	1FFFFh
		SA007	20000h	27FFFh
		SA008	28000h	2FFFFh
		SA009	30000h	37FFFh
		SA010	38000h	3FFFFh
		SA011	40000h	47FFFh
		SA012	48000h	4FFFFh
		SA013	50000h	57FFFh
		SA014	58000h	5FFFFh
		SA015	60000h	67FFFh
		SA016	68000h	6FFFFh
		SA017	70000h	77FFFh
		SA018	78000h	7FFFFh
		SA019	80000h	87FFFh
		SA020	88000h	8FFFFh
		SA021	90000h	97FFFh
		SA022	98000h	9FFFFh
		SA023	A0000h	A7FFFh
		SA024	A8000h	AFFFFh
		SA025	B0000h	B7FFFh
		SA026	B8000h	BFFFFh
		SA027	C0000h	C7FFFh
		SA028	C8000h	CFFFFh
		SA029	D0000h	D7FFFh
		SA030	D8000h	DFFFFh
		SA031	E0000h	E7FFFh
		SA032	E8000h	EFFFFh
		SA033	F0000h	F7FFFh
		SA034	F8000h	FFFFFh

Table 6.2 S29WS064R Sector and Memory Address Map (Bottom Boot) (Sheet 2 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
1	32 kwords	SA035	100000h	107FFFh
		SA036	108000h	10FFFFh
		SA037	110000h	117FFFh
		SA038	118000h	11FFFFh
		SA039	120000h	127FFFh
		SA040	128000h	12FFFFh
		SA041	130000h	137FFFh
		SA042	138000h	13FFFFh
		SA043	140000h	147FFFh
		SA044	148000h	14FFFFh
		SA045	150000h	157FFFh
		SA046	158000h	15FFFFh
		SA047	160000h	167FFFh
		SA048	168000h	16FFFFh
		SA049	170000h	177FFFh
		SA050	178000h	17FFFFh
		SA051	180000h	187FFFh
		SA052	188000h	18FFFFh
		SA053	190000h	197FFFh
		SA054	198000h	19FFFFh
		SA055	1A0000h	1A7FFFh
		SA056	1A8000h	1AFFFFh
		SA057	1B0000h	1B7FFFh
		SA058	1B8000h	1BFFFFh
		SA059	1C0000h	1C7FFFh
		SA060	1C8000h	1CFFFFh
		SA061	1D0000h	1D7FFFh
		SA062	1D8000h	1DFFFFh
		SA063	1E0000h	1E7FFFh
		SA064	1E8000h	1EFFFFh
		SA065	1F0000h	1F7FFFh
		SA066	1F8000h	1FFFFFh

Table 6.2 S29WS064R Sector and Memory Address Map (Bottom Boot) (Sheet 3 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
2	32 kwords	SA067	200000h	207FFFh
		SA068	208000h	20FFFFh
		SA069	210000h	217FFFh
		SA070	218000h	21FFFFh
		SA071	220000h	227FFFh
		SA072	228000h	22FFFFh
		SA073	230000h	237FFFh
		SA074	238000h	23FFFFh
		SA075	240000h	247FFFh
		SA076	248000h	24FFFFh
		SA077	250000h	257FFFh
		SA078	258000h	25FFFFh
		SA079	260000h	267FFFh
		SA080	268000h	26FFFFh
		SA081	270000h	277FFFh
		SA082	278000h	27FFFFh
		SA083	280000h	287FFFh
		SA084	288000h	28FFFFh
		SA085	290000h	297FFFh
		SA086	298000h	29FFFFh
		SA087	2A0000h	2A7FFFh
		SA088	2A8000h	2AFFFFh
		SA089	2B0000h	2B7FFFh
		SA090	2B8000h	2BFFFFh
		SA091	2C0000h	2C7FFFh
		SA092	2C8000h	2CFFFFh
		SA093	2D0000h	2D7FFFh
		SA094	2D8000h	2DFFFFh
		SA095	2E0000h	2E7FFFh
		SA096	2E8000h	2EFFFFh
		SA097	2F0000h	2F7FFFh
		SA098	2F8000h	2FFFFFh

Table 6.2 S29WS064R Sector and Memory Address Map (Bottom Boot) (Sheet 4 of 4)

Bank	Sector Size	Sector	Address Start (Word)	Address End (Word)
3	32 kwords	SA099	300000h	307FFFh
		SA100	308000h	30FFFFh
		SA101	310000h	317FFFh
		SA102	318000h	31FFFFh
		SA103	320000h	327FFFh
		SA104	328000h	32FFFFh
		SA105	330000h	337FFFh
		SA106	338000h	33FFFFh
		SA107	340000h	347FFFh
		SA108	348000h	34FFFFh
		SA109	350000h	357FFFh
		SA110	358000h	35FFFFh
		SA111	360000h	367FFFh
		SA112	368000h	36FFFFh
		SA113	370000h	377FFFh
		SA114	378000h	37FFFFh
		SA115	380000h	387FFFh
		SA116	388000h	38FFFFh
		SA117	390000h	397FFFh
		SA118	398000h	39FFFFh
		SA119	3A0000h	3A7FFFh
		SA120	3A8000h	3AFFFFh
		SA121	3B0000h	3B7FFFh
		SA122	3B8000h	3BFFFFh
		SA123	3C0000h	3C7FFFh
		SA124	3C8000h	3CFFFFh
		SA125	3D0000h	3D7FFFh
		SA126	3D8000h	3DFFFFh
		SA127	3E0000h	3E7FFFh
		SA128	3E8000h	3EFFFFh
		SA129	3F0000h	3F7FFFh
		SA130	3F8000h	3FFFFFh

7. Device Operations

This section describes the read, program, erase, simultaneous read/write operations, handshaking, and reset features of the Flash devices.

Operations are initiated by writing specific commands or a sequence with specific address and data patterns into the command registers (see Table 12.1 on page 74 and Table 12.2 on page 75). The command register itself does not occupy any addressable memory location; rather, it 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 input to the internal state machine and the state machine outputs dictate the function of the device. Writing incorrect address and data values or writing them in an improper sequence may place the device in an unknown state, in which case the system must write the reset command to return the device to the reading array data mode.

7.1 Device Operation Table

The device must be setup appropriately for each operation. Table 7.1 describes the required state of each control pin for any particular operation.

Legend

Table 7.1 Device Bus Operations

Operation	CE#	OE#	WE#	CLK	AVD#	Addresses	Data	RDY	RESET#
Asynchronous Operations									
Asynchronous Read - Addresses Latched	L	H	H	X	↑	Addr In	High-Z	H	H
Asynchronous Read AVD# Steady State	L	L	H	X	L	Addr In	Output Valid	H	H
Asynchronous Read - Data on bus	L	L	H	X	H	X	Output Valid	H	H
Asynchronous Write (AVD# Latched Addresses)	L	H	L	X	↑	Addr In	X	H	H
Asynchronous Write (WE# Latched Data)	L	H	↑	X	H	X	Input Valid	H	H
Non-Operations									
Standby (CE#)	H	X	X	X	X	X	High-Z	High-Z	H
Hardware Reset	X	X	X	X	X	X	High-Z	High-Z	↓
Synchronous Operations									
Latch Starting Burst Address by CLK	L	H	H	↑	L	Addr In	Output Invalid	X	H
Advance Burst read to next address	L	L	H	↑	H	X	Output Valid	H	H
Terminate current Burst read cycle	H	X	X	X	X	X	High-Z	High-Z	H
Terminate current Burst read cycle through RESET#	X	X	X	X	X	X	High-Z	High-Z	L
Terminate current Burst read cycle and start new Burst read cycle	L	H	H	↑	L	Addr In	Output Invalid	X	H

L = Logic 0, H = Logic 1, X = can be either V_{IL} or V_{IH}., ↑ = rising edge, ↓ = high to low.

7.2 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_{IO} pin.

7.3 Asynchronous Read

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

The device defaults to reading array data asynchronously after device power-up or hardware reset.

To read data from the memory array, the system must first assert a valid address on A_{max}-A0, while driving AVD# and CE# to V_{IL}. WE# must remain at V_{IH}. The OE# signal must be driven to V_{IL}. CLK may remain at V_{IL} or V_{IH}. The rising edge of AVD# will latch the address, preventing changes to the address lines from affecting the address being accessed. However, AVD# may remain low throughout the read access if the address will remain stable.

Data is output on DQ15-DQ0 pins after the access time (t_{ACC}) has elapsed following the falling edge of AVD#, or the last time the address lines changed while AVD# was low.

7.4 Page Read Mode

The device is capable of fast page mode read. This mode provides faster read access speed for random locations within a page. The random or initial page access is t_{ACC} or t_{CE} and subsequent page read accesses (as long as the locations specified by the microprocessor fall within that page) is equivalent to t_{PACC}. When CE# is de-asserted (= V_{IH}), the reassertion of CE# for subsequent access has access time of t_{ACC} or t_{CE}. Here again, CE# selects the device and OE# is the output control and should be used to gate data to the output inputs if the device is selected. Fast page mode accesses are obtained by keeping A_{max} - A3 constant and changing A2 - A0 to select the specific word within that page.

Address bits A_{max} - A3 select an 8-word page, and address bits A2 - A0 select a specific word within that page. This is an asynchronous operation with the microprocessor supplying the specific word location. See Table 7.2 for details on selecting specific words.

The de-assertion and re-assertion of AVD# creates a new t_{ACC}. It does not matter if AVD# is low or toggles once. However, the address input must always be valid and stable if AVD# is low during the page read. The user must keep AVD# low during and between page reads on address A(2:0).

Table 7.2 Word Select

Word	A2	A1	A0
Word 0	0	0	0
Word 1	0	0	1
Word 2	0	1	0
Word 3	0	1	1
Word 4	1	0	0
Word 5	1	0	1
Word 6	1	1	0
Word 7	1	1	1

7.5 Synchronous (Burst) Read Mode and Configuration Register

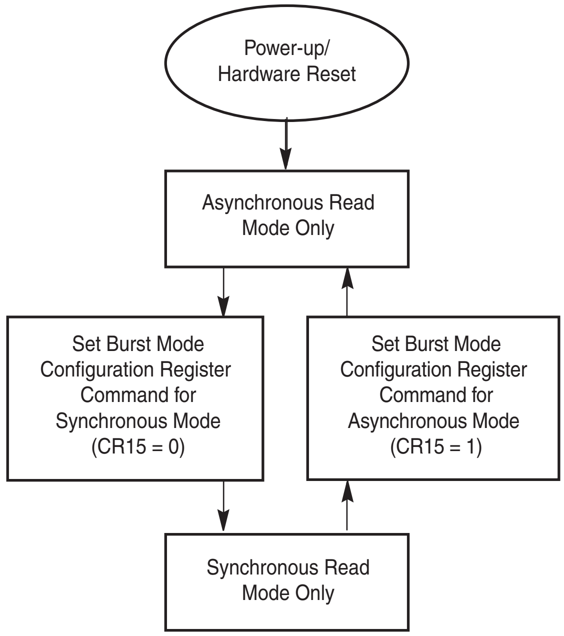
When a series of adjacent addresses needs to be read from the device (in order from lowest to highest address), the synchronous (or burst read) mode can be used to significantly reduce the overall time needed for the device to output array data. After an initial access time required for the data from the first address location, subsequent data is output synchronized to a clock input provided by the system.

The device offers both continuous and linear methods of burst read operation, which are discussed in Section 7.5.1, *Continuous Burst Read Mode* on page 27 and Section 7.5.2, *8-, 16-Word Linear Burst Read with Wrap Around* on page 27.

Since the device defaults to asynchronous read mode after power-up or a hardware reset, the configuration register must be set to enable the burst read mode. Other Configuration Register settings include the number of wait states to insert before the initial word (t_{IACC}) of each burst access, the burst mode in which to operate, and when RDY indicates data is ready to be read.

Prior to entering the burst mode, the system should first determine the configuration register settings (and read the current register settings if desired via the Read Configuration Register command sequence), and then write the configuration register command sequence. See Section 7.5.3, *Configuration Register* on page 27, and Table 12.1, *Memory Array Commands* on page 74 for further details.

Figure 7.1 Synchronous/Asynchronous State Diagram



The device outputs the initial word subject to the following operational conditions:

- t_{IACC} specification: the time from the rising edge of the first clock cycle after addresses are latched to valid data on the device outputs.
- configuration register setting CR13-CR11: the total number of clock cycles (wait states) that occur before valid data appears on the device outputs. The effect is that t_{IACC} is lengthened.

The device outputs subsequent words t_{BACC} after the active edge of each successive clock cycle, which also increments the internal address counter. The device outputs burst data at this rate subject to the following operational conditions:

- starting address: whether the address is divisible by eight (where A[2:0] is 000). A divisible-by-eight address incurs the least number of additional wait states that occur after the initial word.
- boundary crossing: There is a boundary at every 128 words due to the internal architecture of the device. One additional wait state must be inserted when crossing this boundary if the memory bus is operating at a high clock frequency. Please refer to the tables below.
- clock frequency: the speed at which the device is expected to burst data. Higher speeds require additional wait states after the initial word for proper operation.

In all cases, with or without latency, the RDY output indicates when the next data is available to be read.

Table 7.3 to Table 7.8 reflect wait states required for S29WS064R devices. Refer to Table 7.11, *Configuration Register* on page 27 (CR13 - CR11) and timing diagrams for more details.

Table 7.3 Address Latency for 8 or More Wait States

Word	Initial Wait	Subsequent Clock Cycles After Initial Wait States								
0	8 or more 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 7.4 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 7.5 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 7.6 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 7.7 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

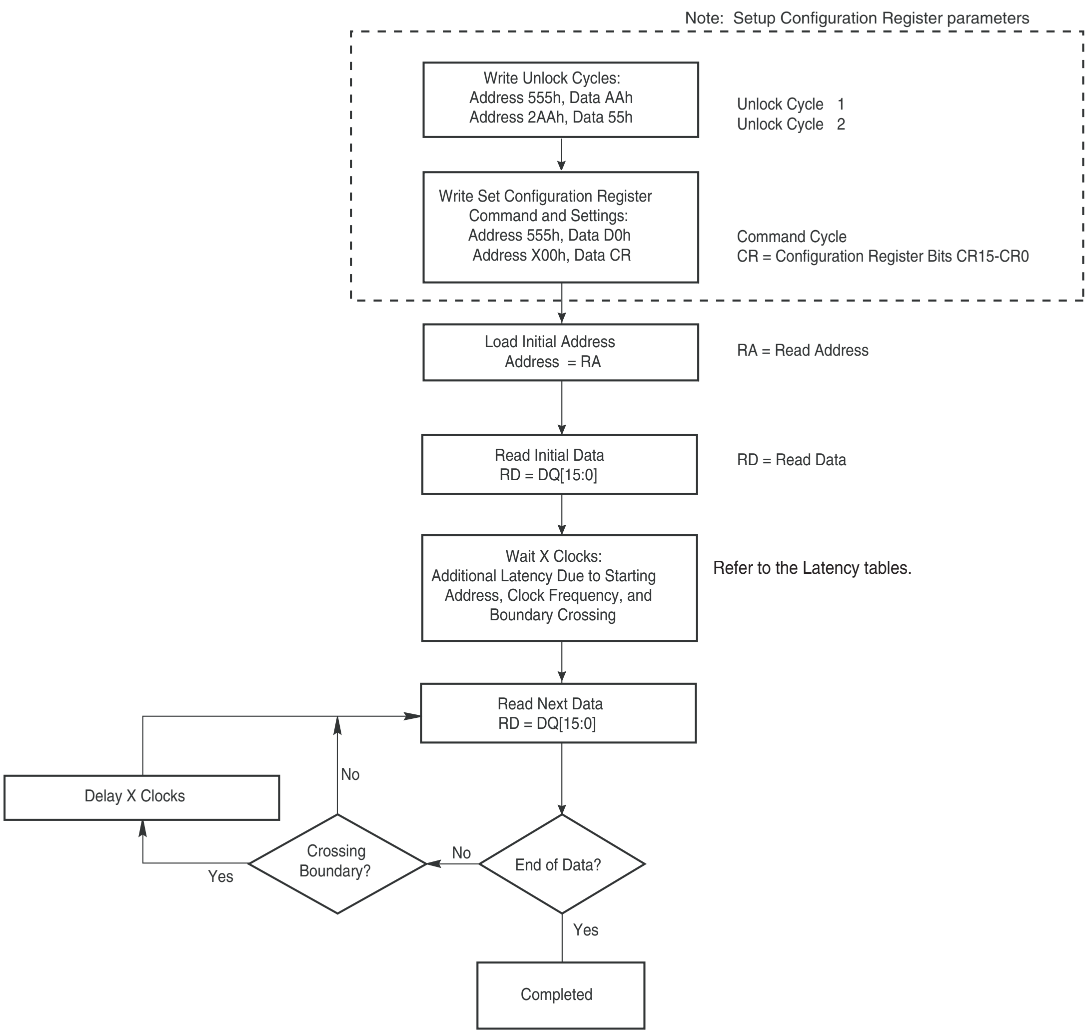
Table 7.8 Address Latency for 3 Wait States

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

Table 7.9 128 Word Boundary Crossing Latency - Additional Wait States

Initial Wait States	Boundary Crossing Latency
3	0 ws
4	
5	
6	
7	
8	
9	1 ws
10 to 13	2 ws

Figure 7.2 Synchronous Read



7.5.1 Continuous Burst Read Mode

In the continuous burst read mode, the device outputs sequential burst data from the starting address given and then wrap around to address 000000h when it reaches the highest addressable memory location. The burst read mode continues until the system drives CE# high, or RESET= V_{IL}. Continuous burst mode can also be aborted by asserting AVD# low and providing a new address to the device.

If the address being read crosses a 128-word line boundary (as mentioned above) and the subsequent word line is not being programmed or erased, additional latency cycles are required as reflected by the configuration register table (Table 7.11).

If the address crosses a bank boundary while the subsequent bank is programming or erasing, the device provides read status information and the clock is ignored. Upon completion of status read or program or erase operation, the host can restart a burst read operation using a new address and AVD# pulse.

7.5.2 8-, 16-Word Linear Burst Read with Wrap Around

In a linear burst read operation, a fixed number of words (8, 16 words) are read from consecutive addresses that 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 7.10).

Table 7.10 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...

For example, if the starting address in the 8-word mode is 3Ch, the address range to be read would be 38-3Fh, and the burst sequence would be 3C-3D-3E-3F-38-39-3A-3Bh. Thus, the device outputs all words in that burst address group until all word are read, regardless of where the starting address occurs in the address group, and then terminates the burst read.

In a similar fashion, the 16-word Linear Wrap mode begins its burst sequence on the starting address provided to the device, then wraps back to the first address in the selected address group.

Note: in this mode the address pointer does not cross the boundary that occurs every 128 words; thus, no additional wait states are inserted due to boundary crossing.

7.5.3 Configuration Register

The configuration register sets various operational parameters associated with burst mode. Upon power-up or hardware reset, the device defaults to the asynchronous read mode, and the configuration register settings are in their default state. The host system should determine the proper settings for the entire configuration register, and then execute the Set Configuration Register command sequence, before attempting burst operations. The configuration register is not reset after de-asserting CE#. The Configuration Register can also be read using a command sequence (see Table 12.1, *Memory Array Commands* on page 74). The following list describes the register settings.

Table 7.11 Configuration Register

CR Bit	Function	Settings (Binary)
CR15	Set Device Read Mode	0 = Synchronous Read (Burst Mode) Enabled 1 = Asynchronous Read Mode (default) Enabled
CR14	Reserved	0 = Default
CR13 CR12 CR11	Programmable Wait State	001 = Data is valid on the 3rd active CLK edge after AVD# transition to V_{IH} 010 = Data is valid on the 4th active CLK edge after AVD# transition to V_{IH} 011 = Data is valid on the 5th active CLK edge after AVD# transition to V_{IH} 100 = Data is valid on the 6th active CLK edge after AVD# transition to V_{IH} 101 = Data is valid on the 7th active CLK edge after AVD# transition to V_{IH} (default) 110 = Data is valid on the 8th active CLK edge after AVD# transition to V_{IH} 111 = Data is valid on the 9th active CLK edge after AVD# transition to V_{IH}

Table 7.11 Configuration Register

CR Bit	Function	Settings (Binary)
CR10	RDY Polarity	0 = RDY signal active low 1 = RDY signal active high (default)
CR9	Reserved	1 = Default
CR8	RDY	0 = RDY active one clock cycle before data 1 = RDY active with data (default) When CR13-CR11 are set to 000, RDY is active with data regardless of CR8 setting.
CR7	Reserved	1 = Default
CR6	Reserved	1 = Default
CR5	Reserved	0 = Default
CR4	Reserved	0 = Default
CR3	Reserved	1 = Default
CR2 CR1 CR0	Burst Length	000 = Continuous (default) 010 = 8-Word Linear Burst 011 = 16-Word Linear Burst (All other bit settings are reserved)

Notes:
1. *Refer to Table 7.3 to Table 7.8 for wait states requirements.*
2. *Refer to Section 11.7.2, Synchronous/Burst Read on page 57 timing diagrams*
3. *Configuration Register is in the default state upon power-up or hardware reset.*

The configuration register can be read with a four-cycle command sequence. See Table 12.1, *Memory Array Commands* on page 74 for sequence details. A software reset command is required after reading or setting the configuration register to set the device into the correct state.

7.6 Autoselect

The Autoselect is used for manufacturer ID, Device identification, and sector protection information. This mode is primarily intended for programming equipment to automatically match a device 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 (see Table 7.12). The remaining address bits are don't care. The most significant four bits of the address during the third write cycle selects the bank from which the Autoselect codes are read by the host. All other banks can be accessed normally for data read without exiting the Autoselect mode.

- To access the Autoselect codes, the host system must issue the Autoselect command.
- 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. Autoselect does not support simultaneous operations or burst mode.
- 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).

See Table 12.1, *Memory Array Commands* on page 74 for command sequence details.

Table 7.12 Autoselect Addresses

Description	Address	Read Data
Manufacturer ID	(BA) + 00h	0001h
Device ID, Word 1	(BA) + 01h	007Eh
Device ID, Word 2	(BA) + 0Eh	Top Boot: 004Fh Bottom Boot: 0057h
Device ID, Word 3	(BA) + 0Fh	0000h

Table 7.12 Autoselect Addresses

Description	Address	Read Data
Indicator Bits	(BA) + 07h	DQ15-DQ8 = Reserved DQ7 (Factory Lock Bit): 1 = Locked, 0 = Not Locked DQ6 (Customer Lock Bit): 1 = Locked, 0 = Not Locked DQ5 - DQ0 = Reserved
Sector Protect Verify	(SA) + 02h	0001h = Locked, 0000h = Unlocked

Software Functions and Sample Code

Table 7.13 Autoselect Entry

(LLD Function = lld_AutoselectEntryCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	BAxAAAh	BAx555h	00AAh
Unlock Cycle 2	Write	BAx555h	BAx2AAh	0055h
Autoselect Command	Write	BAxAAAh	BAx555h	0090h

Table 7.14 Autoselect Exit

(LLD Function = lld_AutoselectExitCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	base + XXXh	base + XXXh	00F0h

Notes:
1. *Any offset within the device works.*
2. *BA = Bank Address. The bank address is required.*
3. *base = base address.*

The following is a C source code example of using the autoselect function to read the manufacturer ID. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Here is an example of Autoselect mode (getting manufacturer ID) */
/* Define UINT16 example: typedef unsigned short UINT16; */

UINT16 manuf_id;

/* Auto Select Entry */

*( (UINT16 *)bank_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (UINT16 *)bank_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (UINT16 *)bank_addr + 0x555 ) = 0x0090; /* write autoselect command */

/* multiple reads can be performed after entry */

manuf_id = *( (UINT16 *)bank_addr + 0x000 ); /* read manuf. id */

/* Autoselect exit */

*( (UINT16 *)base_addr + 0x000 ) = 0x00F0; /* exit autoselect (write reset command) */
```

7.7 Program/Erase Operations

These devices are capable of several modes of programming and or erase operations which are described in detail in the following sections. However, prior to any programming and or erase operation, devices must be setup appropriately as outlined in the configuration register (Table 7.11).

For any synchronous program and or erase operations, including writing command sequences, 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 programming data.

During asynchronous write operations, addresses are latched on the rising edge of AVD# while data is latched on the first rising edge of WE#. If AVD# is kept at V_{IL}, addresses and data are latched on the first rising edge of WE#.

Note the following:

- When the Embedded Program algorithm is complete, the device returns to the read mode.
- The system can determine the status of the program operation by using DQ7 or DQ6. Refer to Section 7.7.8, *Write Operation Status* on page 41 for information on these status bits.
- A “0” cannot be programmed back to a “1.” Attempting to do so causes the device to set DQ5 = 1 (halting any further operation and requiring a reset command). A succeeding read shows that the data is still “0.” Only erase operations can convert a “0” to a “1.”
- Any commands written to the device during the Embedded Program Algorithm are ignored except the Program Suspend command.
- Secured Silicon Sector, Autoselect, and CFI functions are unavailable when a program operation is in progress.
- A hardware reset immediately terminates the program operation and the program command sequence should be reinitiated once the device has returned to the read mode, to ensure data integrity.
- Programming is allowed in any sequence and across sector boundaries for single word programming operation.

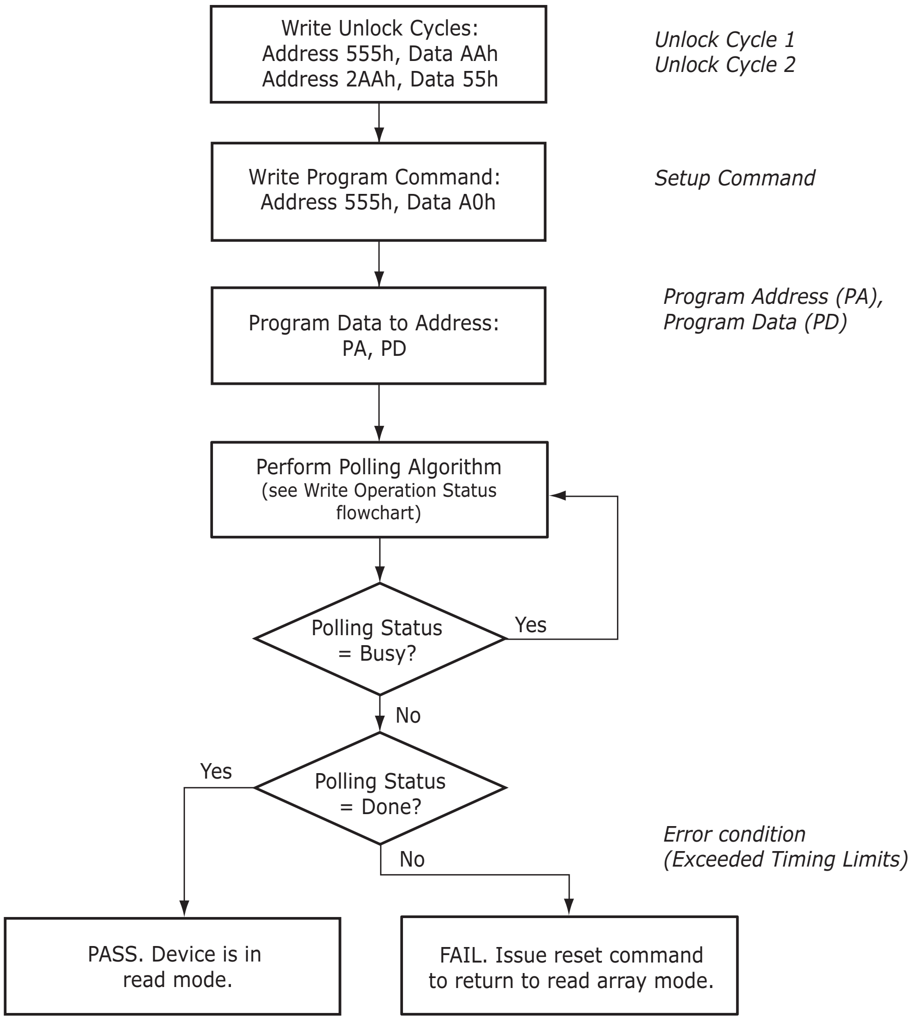
7.7.1 Single Word Programming

Single word programming mode is the simplest method of programming. In this mode, four Flash command write cycles are used to program an individual Flash address. The data for this programming operation could be 8- or 16-bits wide. While this method is supported by all Spansion devices, in general it is not recommended for devices that support Write Buffer Programming. See Table 12.1, *Memory Array Commands* on page 74 for the required bus cycles and Figure 7.3 for the flowchart.

When the Embedded Program algorithm is complete, the device then returns to the read mode and addresses are no longer latched. The system can determine the status of the program operation by using DQ7 or DQ6. Refer to Section 7.7.8, *Write Operation Status* on page 41 for information on these status bits.

- During programming, any command (except the Suspend Program command) is ignored.
- The Secured Silicon Sector, Autoselect, and CFI functions are unavailable when a program operation is in progress.
- A hardware reset immediately terminates the program operation. The program command sequence should be reinitiated once the device has returned to the read mode, to ensure data integrity.

Figure 7.3 Single Word Program



Software Functions and Sample Code

Table 7.15 Single Word Program

(LLD Function = lld_ProgramCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	BAxAAAh	BAx555h	00AAh
Unlock Cycle 2	Write	BAx554h	BAx2AAh	0055h
Program Setup	Write	BAxAAAh	BAx555h	00A0h
Program	Write	Word Address	Word Address	Data Word

Note:
1. Base = Base Address

The following is a C source code example of using the single word program function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Program Command */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00A0; /* write program setup command */
  *( (UINT16 *)pa ) = data; /* write data to be programmed */
/* Poll for program completion */
```

7.7.2 Write Buffer Programming

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 occurs. At this point, the system writes the number of “word locations minus 1” that are loaded into the page buffer at the Sector Address in which programming occurs. This tells the device how many write buffer addresses are 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 aborts. (Number loaded = the number of locations to program minus 1. For example, if the system programs 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 elected-write-buffer-page.

The “write-buffer-page” is selected by using the addresses A_{MAX} - A5.

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 ABORTs.)

After writing the Starting Address/Data pair, the system then writes the remaining address/data pairs into the write buffer.

Note that if a Write Buffer address location is loaded multiple times, the “address/data pair” counter is decremented for every data load operation. Also, the last data loaded at a location before the “Program Buffer to Flash” confirm command is 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 abort the Write Buffer Programming operation. The device goes “busy.” The Data# 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# 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 returns to READ mode.

The Write Buffer Programming Sequence is 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 that the Secured Silicon sector, autoselect, and CFI functions are unavailable when a program operation is in progress.

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.

Use of the write buffer is strongly recommended for programming when multiple words are to be programmed. Write buffer programming is approximately eight times faster than programming one word at a time.

Software Functions and Sample Code

Table 7.16 Write Buffer Program

(LLD Functions Used = lld_WriteToBufferCmd, lld_ProgramBufferToFlashCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	Base + AAAh	Base + 555h	00AAh
2	Unlock	Write	Base + 554h	Base + 2AAh	0055h
3	Write Buffer Load Command	Write	Program Address		0025h
4	Write Word Count	Write	Program Address		Word Count (N-1)h
Number of words (N) loaded into the write buffer can be from 1 to 32 words.					
5 to 36	Load Buffer Word N	Write	Program Address, Word N		Word N
Last	Write Buffer to Flash	Write	Sector Address		0029h

Notes:

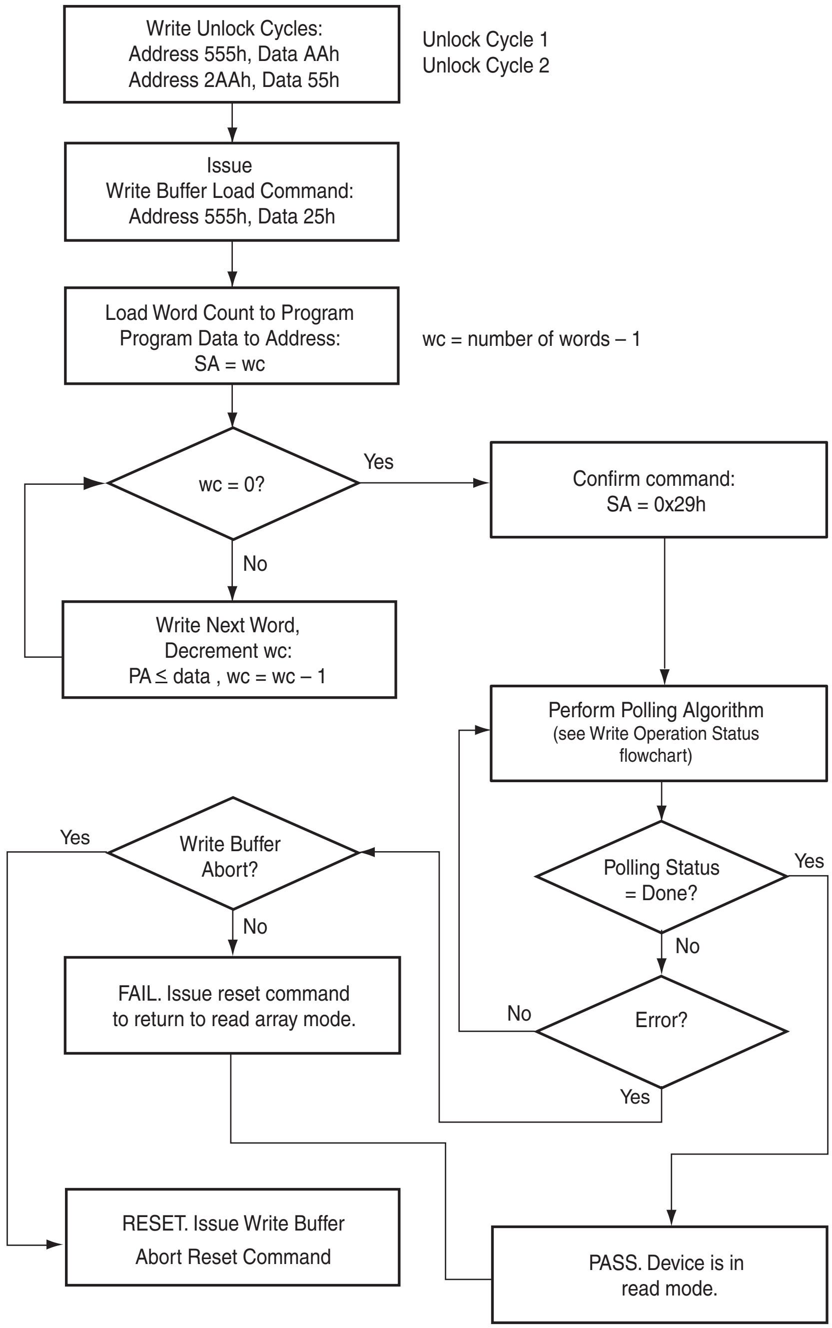
1. *Base = Base Address.*
2. *Last = Last cycle of write buffer program operation; depending on number of words written, the total number of cycles may be from 6 to 37.*
3. *For maximum efficiency, it is recommended that the write buffer be loaded with the highest number of words (N words) possible.*

The following is a C source code example of using the write buffer program function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Write Buffer Programming Command */
/* NOTES: Write buffer programming limited to 16 words. */
/* All addresses to be written to the flash in */
/* one operation must be within the same flash */
/* page. A flash page begins at addresses */
/* evenly divisible by 0x20. */
  UINT16 *src = source_of_data; /* address of source data */
  UINT16 *dst = destination_of_data; /* flash destination address */
  UINT16 wc = words_to_program -1; /* word count (minus 1) */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)sector_address ) = 0x0025; /* write write buffer load command */
  *( (UINT16 *)sector_address ) = wc; /* write word count (minus 1) */
loop:
  *dst = *src; /* ALL dst MUST BE SAME PAGE */ /* write source data to destination */
  dst++; /* increment destination pointer */
  src++; /* increment source pointer */
  if (wc == 0) goto confirm /* done when word count equals zero */
  wc--; /* decrement word count */
  goto loop; /* do it again */
confirm:
  *( (UINT16 *)sector_address ) = 0x0029; /* write confirm command */
  /* poll for completion */

/* Example: Write Buffer Abort Reset */
  *( (UINT16 *)addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)addr + 0x555 ) = 0x00F0; /* write buffer abort reset */
```

Figure 7.4 Write Buffer Programming Operation



7.7.3 Sector Erase

The sector erase function erases one sector in the memory array. (See Table 12.1, *Memory Array Commands* on page 74; and Figure 7.5, *Sector Erase Operation* on page 37.) 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. After a successful sector erase, all locations within the erased sector contain FFFFh. The system is not required to provide any controls or timings during these operations.

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 Section 7.7.8, *Write Operation Status* on page 41 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.

Figure 7.5, *Sector Erase Operation* on page 37 illustrates the algorithm for the erase operation. Refer to Section 11.7.6, *Erase and Programming Performance* on page 73 for parameters and timing diagrams.

Software Functions and Sample Code

Table 7.17 Sector Erase

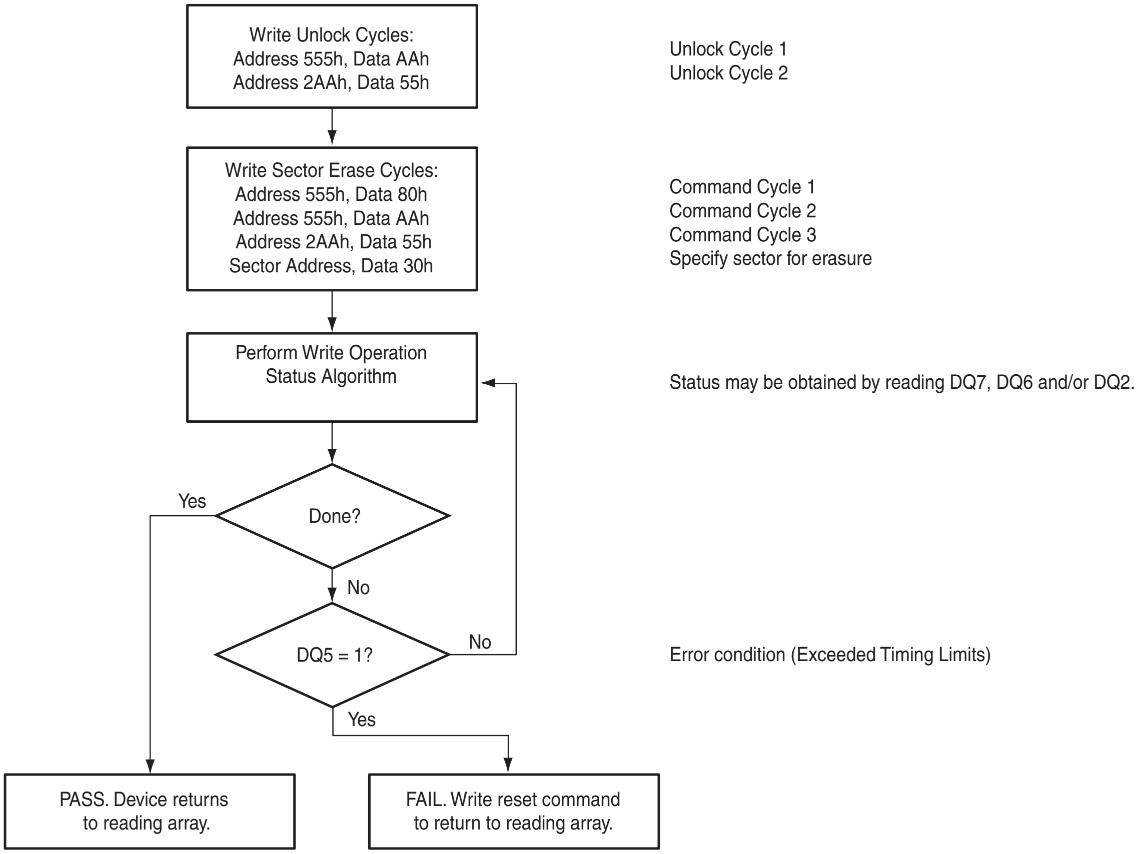
(LLD Function = lld_SectorEraseCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	Base + AAAh	Base + 555h	00AAh
2	Unlock	Write	Base + 554h	Base + 2AAh	0055h
3	Setup Command	Write	Base + AAAh	Base + 555h	0080h
4	Unlock	Write	Base + AAAh	Base + 555h	00AAh
5	Unlock	Write	Base + 554h	Base + 2AAh	0055h
6	Sector Erase Command	Write	Sector Address	Sector Address	0030h

The following is a C source code example of using the sector erase function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Sector Erase Command */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)base_addr + 0x555 ) = 0x0080; /* write setup command */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write additional unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write additional unlock cycle 2 */
  *( (UINT16 *)sector_address )    = 0x0030; /* write sector erase command */
```

Figure 7.5 Sector Erase Operation



Note:
1. See Table 12.1 for erase command sequence.

7.7.4 Chip Erase Command Sequence

Chip erase is a six-bus cycle operation as indicated by Table 12.1, *Memory Array Commands* on page 74. These commands invoke the Embedded Erase algorithm, which 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. After a successful chip erase, all locations of the chip contain FFFFh. The system is not required to provide any controls or timings during these operations. Table 12.1, *Memory Array Commands* on page 74 in the Appendix 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 Section 7.7.8, *Write Operation Status* on page 41 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.

Software Functions and Sample Code

Table 7.18 Chip Erase

(LLD Function = lld_ChipEraseCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	Base + AAAh	Base + 555h	00AAh
2	Unlock	Write	Base + 554h	Base + 2AAh	0055h
3	Setup Command	Write	Base + AAAh	Base + 555h	0080h
4	Unlock	Write	Base + AAAh	Base + 555h	00AAh
5	Unlock	Write	Base + 554h	Base + 2AAh	0055h
6	Chip Erase Command	Write	Base + AAAh	Base + 555h	0010h

The following is a C source code example of using the chip erase function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Chip Erase Command */
/* Note: Cannot be suspended */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)base_addr + 0x555 ) = 0x0080; /* write setup command */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write additional unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write additional unlock cycle 2 */
  *( (UINT16 *)base_addr + 0x000 ) = 0x0010; /* write chip erase command */
```

7.7.5 Erase Suspend/Erase Resume Commands

The Erase Suspend command allows the system to interrupt a sector erase operation and then read data from, or program data to, any sector not 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.

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. Additionally, when an Erase Suspend command is written during an active erase operation, status information is unavailable during the transition from the sector erase operation to the erase suspended state. 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” the sector selected for erasure.) Reading at any address within the erase suspended sector 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 Table 7.24, *Write Operation Status* on page 45 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. In the erase-suspend-read mode, the system can also issue the Autoselect command sequence. Refer to Section 7.7.2, *Write Buffer Programming* on page 32 and Section 7.6, *Autoselect* on page 28 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.

Software Functions and Sample Code

Table 7.19 Erase Suspend

(LLD Function = lld_EraseSuspendCmd)

Cycle	Description	Byte Address	Word Address	Data
1	Write	Bank Address	Bank Address	00B0h

The following is a C source code example of using the erase suspend function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Erase suspend command */
  *( (UINT16 *)bank_addr + 0x000 ) = 0x00B0; /* write suspend command */
```

Table 7.20 Erase Resume

(LLD Function = lld_EraseResumeCmd)

Cycle	Description	Byte Address	Word Address	Data
1	Write	Bank Address	Bank Address	0030h

The following is a C source code example of using the erase resume function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Erase resume command */
  *( (UINT16 *)bank_addr + 0x000 ) = 0x0030; /* write resume command */
  /* The flash needs adequate time in the resume state */
```

7.7.6 Program Suspend/Program Resume Commands

The Program Suspend command allows the system to interrupt an 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 “don't-cares” 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, 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 Section 7.6, *Autoselect* on page 28 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 Section 7.7.8, *Write Operation Status* on page 41 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 resumed programming.

Software Functions and Sample Code

Table 7.21 Program Suspend

(LLD Function = lld_ProgramSuspendCmd)

Cycle	Description	Byte Address	Word Address	Data
1	Write	Bank Address	Bank Address	00B0h

The following is a C source code example of using the program suspend function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Program suspend command */
  *( (UINT16 *)bank_addr + 0x000 ) = 0x00B0; /* write suspend command */
```

Table 7.22 Program Resume

(LLD Function = lld_ProgramResumeCmd)

Cycle	Description	Byte Address	Word Address	Data
1	Write	Bank Address	Bank Address	0030h

The following is a C source code example of using the program resume function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Program resume command */
  *( (UINT16 *)bank_addr + 0x000 ) = 0x0030; /* write resume command */
```

7.7.7 Accelerated Program/Chip Erase

Accelerated single word programming, write buffer programming, sector erase, and chip erase operations are enabled through the ACC function. This method is faster than the standard chip program and erase command sequences.

The accelerated chip program and erase functions must not be used more than 10 times per sector. In addition, accelerated chip program and erase should be performed at room temperature (25°C ±10°C).

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 ACC input, upon completion of the embedded program or erase operation, returns the device to normal operation.

- Sectors must be unlocked prior to raising ACC to V_{HH}.
- The ACC pin must not be at V_{HH} for operations other than accelerated programming and accelerated chip erase, or device damage may result.
- The ACC pin must not be left floating or unconnected; inconsistent behavior of the device may result.
- ACC locks all sectors if set to V_{IL}; ACC should be set to V_{IH} for all other conditions.

7.7.8 Write Operation Status

The device provides several bits to determine the status of a program or erase operation. The following subsections describe the function of DQ1, DQ2, DQ5, DQ6, and DQ7.

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 returns 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.

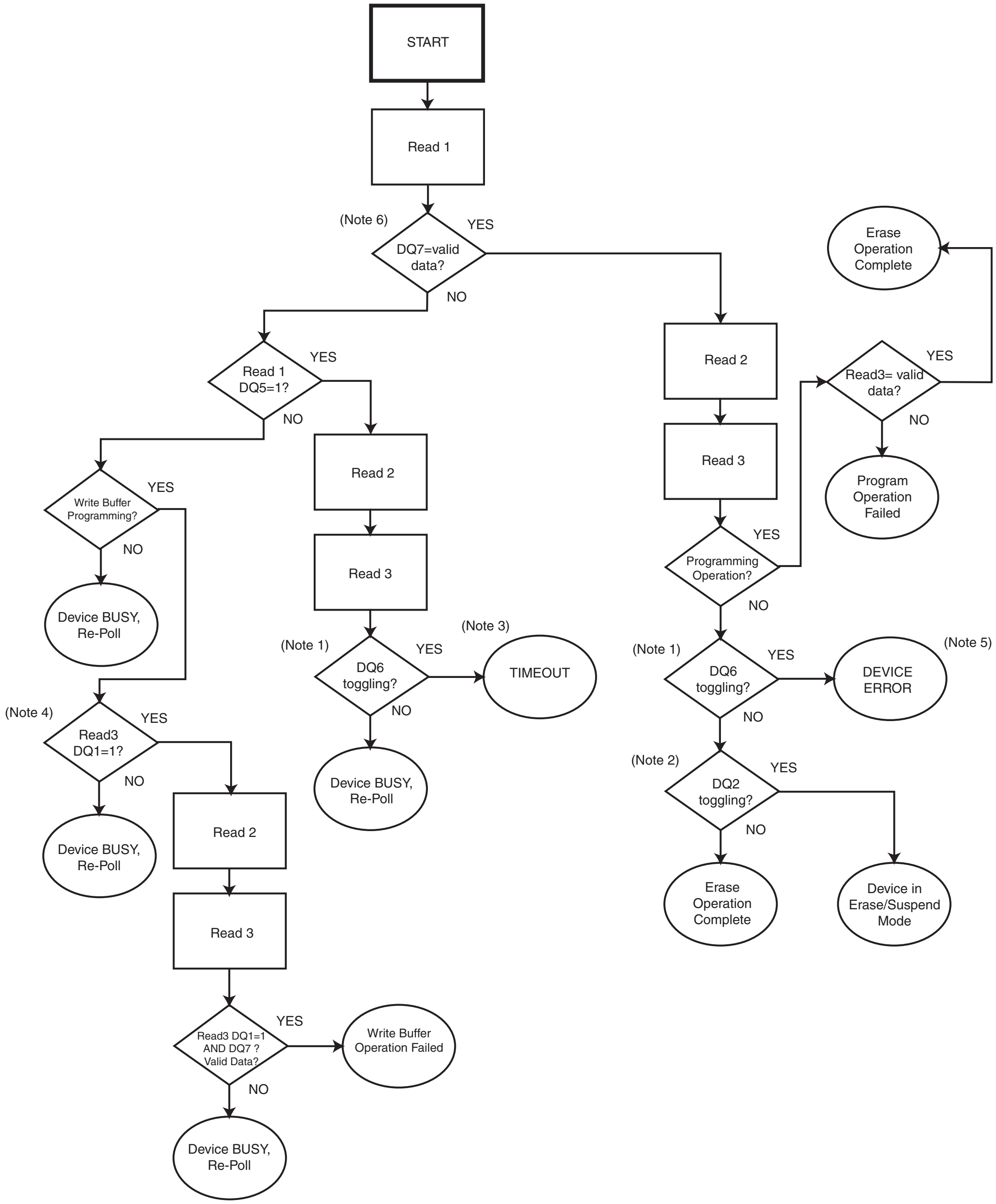
During the Embedded Erase Algorithm, Data# polling produces a “0” on DQ7. When the Embedded Erase algorithm is complete, or if the bank enters the Erase Suspend mode, Data# Polling produces a “1” on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status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

After an erase command sequence is written, if the sector selected for erasing is protected, Data# Polling on DQ7 is active for approximately t_{ASP}, then the bank returns to the read mode. If the selected sector is not protected, the Embedded Erase algorithm erases the sector.

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 appears on successive read cycles.

See the following for more information: Table 7.24, *Write Operation Status* on page 45, shows the outputs for Data# Polling on DQ7. Figure 7.6, *Write Operation Status Flowchart* on page 42 shows the Data# Polling algorithm; and Figure 11.18, *Data# Polling Timings (During Embedded Algorithm)* on page 67 shows the Data# Polling timing diagram.

Figure 7.6 Write Operation Status Flowchart



Notes:
1. *DQ6 is toggling if Read2 DQ6 does not equal Read3 DQ6.*
2. *DQ2 is toggling if Read2 DQ2 does not equal Read3 DQ2.*
3. *May be due to an attempt to program a 0 to 1. Use the RESET command to exit operation.*
4. *Write buffer error if DQ1 of last read =1.*
5. *Invalid state, use RESET command to exit operation.*
6. *Valid data is the data that is intended to be programmed or all 1's for an erase operation.*
7. *Data polling algorithm valid for all operations except advanced sector protection.*

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).

During an Embedded Program or Erase algorithm operation, successive read cycles to any address cause DQ6 to toggle. When the operation is complete, DQ6 stops toggling.

After an erase command sequence is written, if the sector selected for erasing is protected, DQ6 toggles for approximately t_{ASP} [all sectors protected toggle time], then returns to reading array data.

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 under Section 7.7.8, *Write Operation Status* on page 41.

If a program address falls within a protected sector, DQ6 toggles for approximately t_{PAP} 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: Figure 7.6, *Write Operation Status Flowchart* on page 42; Figure 11.19, *Toggle Bit Timings (During Embedded Algorithm)* on page 68, and Table 7.23, *DQ6 and DQ2 Indications* on page 43 and Table 7.24, *Write Operation Status* on page 45.

Toggle Bit I on DQ6 requires either OE# or CE# to be de-asserted and reasserted to show the change in state.

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 the sector that has been selected for erasure. 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 sector is selected for erasure. Thus, both status bits are required for sector and mode information. Refer to Table 7.23 to compare outputs for DQ2 and DQ6. See the following for additional information: Figure 7.6, *Write Operation Status Flowchart* on page 42, the DQ6: Toggle Bit I section under Section 7.7.8, *Write Operation Status* on page 41, and Figure 11.18 to Figure 11.25.

Table 7.23 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.
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.

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 erases operation. If it is still toggling, the device did not complete 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. Refer to Figure 7.6 on page 42 for more details.

Note:

- When verifying the status of a write operation (embedded program/erase) of a memory bank, DQ6 and DQ2 toggle between high and low states in a series of consecutive and contiguous status read cycles. In order for this toggling behavior to be properly observed, the consecutive status bit reads must not be interleaved with read accesses to other memory banks. If it is not possible to temporarily prevent reads to other memory banks, then it is recommended to use the DQ7 status bit as the alternative method of determining the active or inactive status of the write operation.
- Data polling provides erroneous results during erase suspend operation using DQ2 or DQ6 for any address changes after CE# assertion or without AVD# pulsing low. The user is required to pulse AVD# following an address change or assert CE# after address is stable during status polling. See Figure 11.21 through Figure 11.24.

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).

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 Section 7.7.2, *Write Buffer Programming* on page 32 for more details.

Table 7.24 Write Operation Status

Status			DQ7 (Note 2)	DQ6	DQ5 (Note 1)	DQ2 (Note 2)	DQ1 (Note 4)
Standard Mode	Embedded Program Algorithm		DQ7#	Toggle	0	No toggle	0
	Embedded Erase Algorithm		0	Toggle	0	Toggle	N/A
Program Suspend Mode (Note 3)	Reading within Program Suspended Sector		INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)
	Reading within Non-Program Suspended Sector		Data	Data	Data	Data	Data
Erase Suspend Mode (Note)	Erase-Suspend-Read	Erase Suspended Sector	1	No toggle	0	Toggle	N/A
		Non-Erase Suspended Sector	Data	Data	Data	Data	Data
	Erase-Suspend-Program		DQ7#	Toggle	0	N/A	N/A
Write to Buffer (Note 5)	BUSY State		DQ7#	Toggle	0	N/A	0
	Exceeded Timing Limits		DQ7#	Toggle	1	N/A	0
	ABORT State		DQ7#	Toggle	0	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 subsection DQ5: Exceeded Timing Limits under 7.7.8, Write Operation Status on page 41 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**.*

For any address changes after CE# assertion, re-assertion of CE# might be required after the addresses become stable for data polling during the erase suspend operation using DQ2/DQ6.

7.8 Simultaneous Read/Write

The simultaneous read/write feature allows the host system to read data from one bank of memory while programming or erasing another bank of memory. An erase operation may also be suspended to read from or program another location within the same bank (except the sector being erased). Figure 11.28, *Back-to-Back Read/Write Cycle Timings* on page 72, shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to the table in Section 11.6.1, *CMOS Compatible* on page 56 for read-while-program and read-while-erase current specifications.

7.9 Writing Commands/Command Sequences

When the device is configured for Asynchronous read, only Asynchronous write operations are allowed, and CLK is ignored. When in the Synchronous read mode configuration, the device is able to perform both Asynchronous and Synchronous write operations. CLK and AVD# induced address latches are supported in the Synchronous programming mode. During a synchronous write operation, 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. During an asynchronous write operation, the system must drive CE# and WE# to V_{IL} and OE# to V_{IH} when providing an address, command, and data.

Addresses are latched on the last falling edge of WE# or CE#, while data is latched on the 1st rising edge of WE# or CE#. An erase operation can erase one sector, or the entire device. Table 6.1 and Table 6.2 indicate the address space that each sector occupies. The device address space is divided into four banks. For top boot devices, Banks 0 through 2 contain only 64 kword sectors, while Bank 3 contains 8 kword boot sectors in addition to 64 kword sectors. For bottom boot devices, Bank 0 contains 8 kword boot sectors in addition to 64 kword sectors, while Banks 1 through 3 contain only 64 kword sectors. A "bank address" is the set of address bits required to uniquely select a bank. Similarly, a "sector address" is the address bits required to uniquely select a sector. I_{CC2} in Section 11.6, *DC Characteristics* on page 56 represents the active current specification for the write mode. Section 11.7, *AC Characteristics* on page 57 contains timing specification tables and timing diagrams for write operations.

7.10 Handshaking

The handshaking feature allows the host system to detect when data is ready to be read by simply monitoring the RDY (Ready) pin, which is a dedicated output and controlled by CE#.

When the device is configured to operate in synchronous mode, and OE# is low (active), the initial word of burst data becomes available after either the falling or rising edge of the RDY pin (depending on the setting for bit 10 in the Configuration Register). It is recommended that the host system set CR13-CR11 in the Configuration Register to the appropriate number of wait states to ensure optimal burst mode operation (see Table 7.11, *Configuration Register* on page 27).

Bit 8 in the Configuration Register allows the host to specify whether RDY is active at the same time that data is ready, or one cycle before data is ready.

7.11 Hardware Reset

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, resets the configuration register, 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.

To ensure data integrity the operation that was interrupted should be reinitiated once the device is ready to accept another command sequence.

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

RESET# may be tied to the system reset circuitry which enables the system to read the boot-up firmware from the Flash memory upon a system reset.

See Figure 11.5, *V_{CC} Power-up Diagram* on page 55 and Figure 11.13, *Reset Timings* on page 62 for timing diagrams.

7.12 Software Reset

Software reset is part of the command set (see Table 12.1) that also returns the device to array read mode and must be used for the following conditions:

- to exit Autoselect mode
- when DQ5 goes high during write status operation that indicates program or erase cycle was not successfully completed
- exit sector lock/unlock operation
- to return to erase-suspend-read mode if the device was previously in Erase Suspend mode
- after any aborted operations
- exiting Read Configuration Register Mode

Software Functions and Sample Code

Table 7.25 Reset

(LLD Function = lld_ResetCmd)

Cycle	Operation	Byte Address	Word Address	Data
Reset Command	Write	Base + xxxh	Base + xxxh	00F0h

Note:

1. Base = Base Address.

The following is a C source code example of using the reset function. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: Reset (software reset of Flash state machine) */
  *( (UINT16 *)base_addr + 0x000 ) = 0x00F0;
```

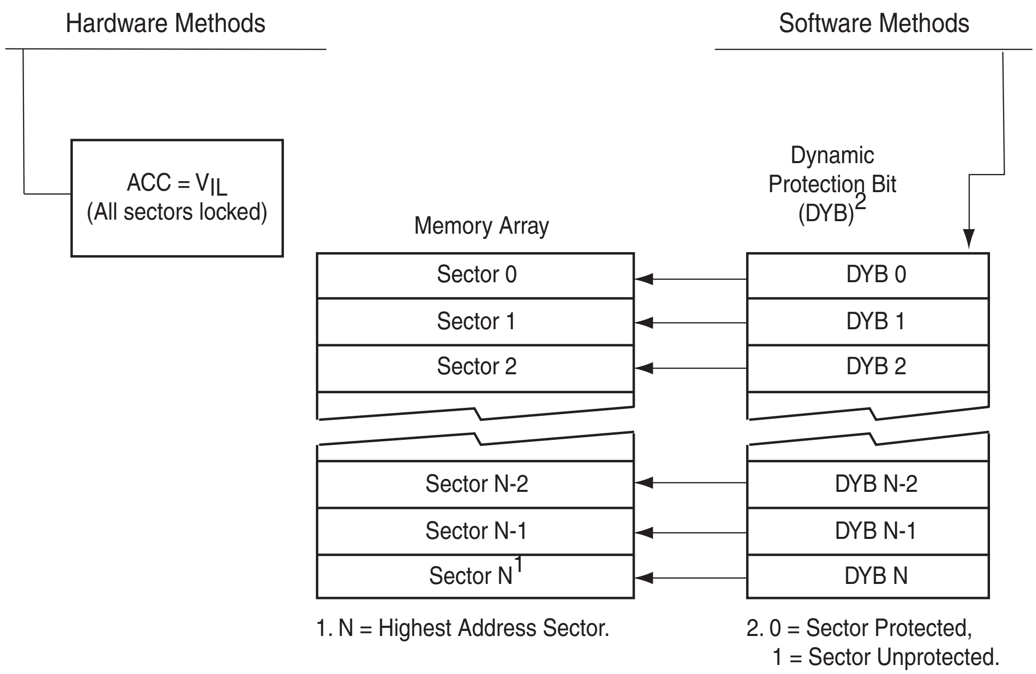
The following are additional points to consider when using the reset command:

- This command resets the banks to the read and address bits are ignored.
- Reset commands are ignored once erasure has begun until the operation is complete.
- Once programming begins, the device ignores reset commands until the operation is complete.
- The reset command may be written between the cycles in a program command sequence before programming begins (prior to the third cycle). 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.
- The reset command may be also written during an Autoselect command sequence.
- If a bank has entered the Autoselect mode while in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.
- 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 does not work during this condition.

8. Advanced Sector Protection/Unprotection

The Advanced Sector Protection/Unprotection feature disables or enables programming or erase operations in any or all sectors and can be implemented through software and/or hardware methods, which are independent of each other. This section describes the various methods of protecting data stored in the memory array. An overview of these methods in shown in Figure 8.1.

Figure 8.1 Advanced Sector Protection/Unprotection



8.1 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 8.1 Lock Register

Device	DQ15-01	DQ0
S29WS064R	Undefined	Secured Silicon Sector Protection Bit

Note:
1. When the device lock register is programmed, all DYBs revert to the power-on default state.

8.2 Dynamic Protection Bits

Dynamic Protection Bits are volatile and unique for each sector and can be individually modified. By issuing the DYB Set or Clear command sequences, the DYBs are 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.

Notes

1. The DYBs can be set (programmed to “0”) or cleared (erased to “1”) as often as needed.
2. When the parts are first shipped, upon power up or reset, the DYBs are set (erased to “1”) by default, putting the sectors in the unprotected state.
3. The DYB Set or Clear commands for the dynamic sectors signify protected or unprotected state of the sectors respectively.

8.3 Hardware Data Protection Methods

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

- When ACC is at V_{IL}, all sectors are locked.

There are additional methods by which intended or accidental erasure of any sectors can be prevented via hardware means. The following subsections describe these methods:

8.3.1 ACC Method

If the system asserts V_{IL} on the ACC input pin, all program and erase functions are disabled and hence all sectors are protected.

8.3.2 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 V_{CC} is greater than V_{LKO}.

8.3.3 Write Pulse “Glitch Protection”

Noise pulses of less than 3 ns (typical) on OE#, CE# or WE# do not initiate a write cycle.

8.3.4 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. Power Conservation Modes

9.1 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 $V_{CC} \pm 0.2$ V. The device requires standard access time (t_{CE}) for read access, 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 Section 11.6, *DC Characteristics* on page 56 represents the standby current specification

9.2 Automatic Sleep Mode

The automatic sleep mode minimizes Flash device energy consumption while in asynchronous mode. The device automatically enables this mode when addresses 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. While in synchronous mode, the automatic sleep mode is disabled. Note that a new burst operation is required to provide new data. I_{CC6} in DC Characteristics (CMOS Compatible) represents the automatic sleep mode current specification.

9.3 Hardware RESET# Input Operation

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, resets the configuration register, 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.

When RESET# is held at $V_{SS} \pm 0.2V$, the device draws CMOS standby current (I_{CC4}). If RESET# is held at V_{IL} but not within $V_{SS} \pm 0.2V$, the standby current is greater.

RESET# may be tied to the system reset circuitry and thus, a system reset would also reset the Flash memory, enabling the system to read the boot-up firmware from the Flash memory.

9.4 Output Disable (OE#)

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

10. Secured Silicon Sector Flash Memory Region

The Secured Silicon Sector provides an extra Flash memory region that enables permanent part identification through an Electronic Serial Number (ESN). The Secured Silicon Sector is 256 words in length that consists of 128 words for factory data and 128 words for customer-secured areas. All Secured Silicon reads outside of the 256-word address range returns invalid data. The Factory Indicator Bit, DQ7, (at Autoselect address 03h) 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.

Please note the following general conditions:

- While Secured Silicon Sector access is enabled, simultaneous operations are allowed except for Bank 0.
- On power-up, or following a hardware reset, the device reverts to sending commands to the normal address space.
- Reads can be performed in the Asynchronous or Synchronous mode.
- Burst mode reads within Secured Silicon Sector wrap from address FFh back to address 00h.
- Reads outside of sector 0 return memory array data.
- Continuous burst read past the maximum address is undefined.
- Sector 0 is remapped from memory array to Secured Silicon Sector array.
- Once the Secured Silicon Sector Entry Command is issued, the Secured Silicon Sector Exit command must be issued to exit Secured Silicon Sector Mode.
- The Secured Silicon Sector is not accessible when the device is executing an Embedded Program or Embedded Erase algorithm.

Table 10.1 Secured Silicon Sector Addresses

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

10.1 Factory Secured Silicon Sector

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”. This prevents cloning of a factory locked part and ensures the security of the ESN and customer code once the product is shipped to the field. The Factory Secured Silicon Sector is unprogrammed by default.

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

- A random, 8 Word secure ESN only within the Factory Secured Silicon Sector.
- Customer code within the Customer Secured Silicon Sector through the Spansion programming service.
- Both a random, secure ESN and customer code through the Spansion programming service.

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

10.2 Customer Secured Silicon Sector

The Customer Secured Silicon Sector is typically shipped unprotected (DQ6 set to “0”), allowing customers to utilize that sector in any manner they choose. If the security feature is not required, the Customer Secured Silicon Sector can be treated as an additional Flash memory space.

Please note the following:

- Once the Customer Secured Silicon Sector area is protected, the Customer Indicator Bit is permanently set to “1.”
- The Customer Secured Silicon Sector can be read any number of times, but can be programmed and locked only once. The Customer Secured Silicon Sector lock must be used with caution as 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.
- The accelerated programming (ACC) function is not available when programming the Customer Secured Silicon Sector, but reading in Banks 1 through 3 is available.
- Once the Customer Secured Silicon Sector is locked and verified, the system must write the Exit Secured Silicon Sector Region command sequence which returns the device to the memory array at sector 0.

10.3 Secured Silicon Sector Entry/Exit Command Sequences

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. See Table 12.1, *Memory Array Commands* on page 74 for address and data requirements for both command sequences.

The Secured Silicon Sector Entry Command allows the following commands to be executed:

- Read customer and factory Secured Silicon areas
- Program the customer Secured Silicon Sector

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 within 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.

Software Functions and Sample Code

The following is a C source code example of using the Secured Silicon Sector Entry, Program, and Exit commands. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

Table 10.2 Secured Silicon Sector Entry

(LLD Function = lld_SecSiSectorEntryCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	Base + AAAh	Base + 555h	00AAh
Unlock Cycle 2	Write	Base + 554h	Base + 2AAh	0055h
Entry Cycle	Write	Base + AAAh	Base + 555h	0088h

Note:
1. Base = Base Address.

```
/* Example: SecSi Sector Entry Command */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)base_addr + 0x555 ) = 0x0088; /* write Secsi Sector Entry Cmd */
```

Table 10.3 Secured Silicon Sector Program

(LLD Function = lld_ProgramCmd)

Cycle	Operation	Byte Address	Word Address	Data
Program Setup	Write	XXXh	XXXh	00A0h
Program	Write	Word Address	Word Address	Data Word

Note:
1. Base = Base Address.

```
/* Example: SecSi Sector Program Command */
  *( (UINT16 *)base_addr + 0x000 ) = 0x00A0; /* write program setup command */
  *( (UINT16 *)pa ) = data;                  /* write data to be programmed */
```

Table 10.4 Secured Silicon Sector Exit

(LLD Function = lld_SecSiSectorExitCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	Base + AAAh	Base + 555h	00AAh
Unlock Cycle 2	Write	Base + 554h	Base + 2AAh	0055h
Exit Cycle	Write	Base + AAAh	Base + 555h	0090h

Note:
1. Base = Base Address.

```
/* Example: SecSi Sector Exit Command */
  *( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
  *( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
  *( (UINT16 *)base_addr + 0x555 ) = 0x0090; /* write SecSi Sector Exit cycle 3 */
  *( (UINT16 *)base_addr + 0x000 ) = 0x0000; /* write SecSi Sector Exit cycle 4 */
```

11. Electrical Specifications

11.1 Absolute Maximum Ratings

Description		Rating
Storage Temperature, Plastic Packages		–65°C to +150°C
Ambient Temperature with Power Applied		–65°C to +125°C
Voltage with Respect to Ground:	All Inputs and I/Os except as noted below (Note 1)	–0.5V to V_{CC} + 0.5V
	V_{CC} (Note 1)	–0.5V to +2.5V
	V_{IO}	–0.5V to +2.5V
	ACC (Note 2)	–0.5V to +9.5V
Output Short Circuit Current (Note 3)		100 mA

Notes

1. *Minimum DC voltage on input or I/Os is –0.5V. During voltage transitions, inputs or I/Os may undershoot V_{SS} to –2.0V for periods of up to 20 ns. See Figure 11.1. Maximum DC voltage on input or I/Os is V_{CC} + 0.5V. During voltage transitions outputs may overshoot to V_{CC} + 2.0V for periods up to 20 ns. See Figure 11.2.*
2. *Minimum DC input voltage on pin ACC is -0.5V. During voltage transitions, ACC may overshoot V_{SS} to –2.0V for periods of up to 20 ns. See Figure 11.1 Maximum DC voltage on pin ACC is +9.5V, which may overshoot to 10.5V 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 11.1 Maximum Negative Overshoot Waveform

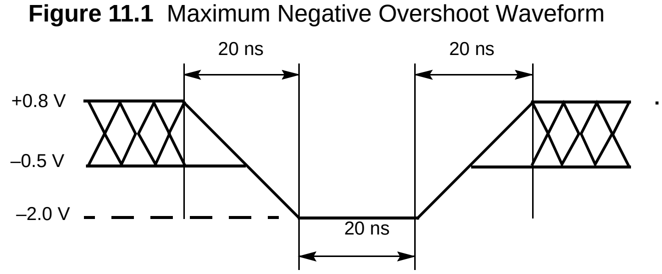
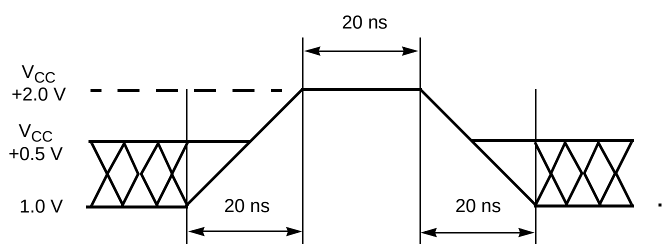


Figure 11.2 Maximum Positive Overshoot Waveform



11.2 Operating Ranges

Specifications		Range
Ambient Temperature (T_A), Wireless (W) Device		–25°C to +85°C
Ambient Temperature (T_A), during Accelerated Program/Erase		+20°C to +40°C
Supply Voltages	V_{CC}	+1.70V to +1.95V
	V_{IO}	+1.70V to +1.95V

Notes

1. *Operating ranges define those limits between which the functionality of the device is guaranteed.*
2. *Industrial Temperature Range (-40°C to +85°C) is also available. For device specification differences, please refer to the Specification Supplement with Publication Number S29WS064R_SP.*

11.3 Test Conditions

Figure 11.3 Test Setup

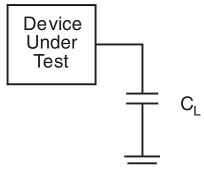


Table 11.1 Test Specifications

Test Condition	All Speed Options	Unit
Output Load Capacitance, CL (including jig capacitance)	30	pF
Input Rise and Fall Times	3.0 @ 66 MHz 2.5 @ 83 MHz 1.85 @ 108 MHz	ns
Input Pulse Levels	0.0-V_{IO}	V
Input timing measurement reference levels	V_{IO}/2	V
Output timing measurement reference levels	V_{IO}/2	V

11.4 Key to Switching Waveforms

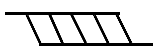
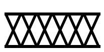
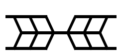
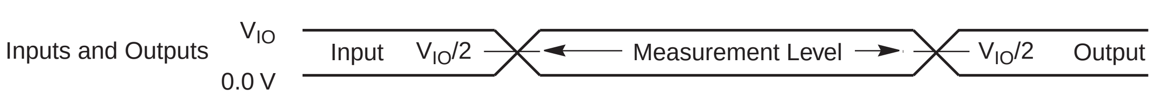
Waveform	Inputs	Outputs
	Steady	
	Changing from H to L	
	Changing from L to H	
	Don’t Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High Impedance State (High-Z)

Figure 11.4 Input Waveforms and Measurement Levels



11.5 V_{CC} Power Up

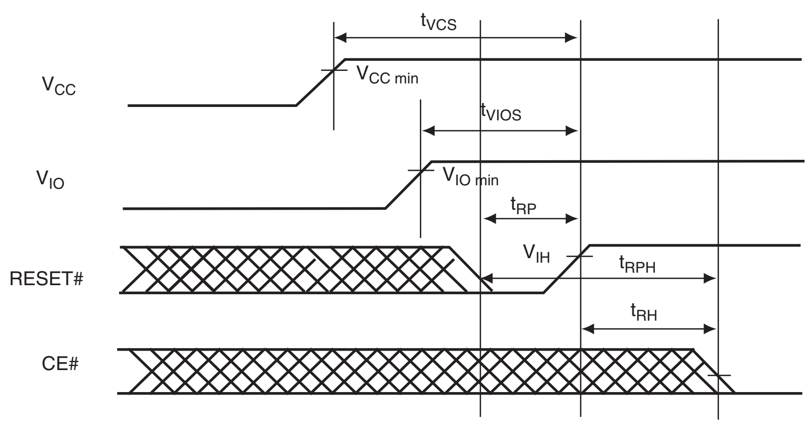
Table 11.2 V_{CC} Power-up

Parameter	Description	Test Setup	Speed	Unit
t_{VCS}	V_{CC} Setup Time	Min	300	µs
t_{VIOS}	V_{IO} Setup Time	Min	300	µs
t_{RH}	Time between RESET# (high) and CE# (low)	Min	200	ns
t_{RP}	RESET# Pulse Width	Min	50	ns
t_{RPH}	RESET# Low to CE# Low	Min	10	µs

Notes

1. *RESET# must be high after V_{CC} and V_{IO} are higher than V_{CC} minimum.*
2. *$V_{CC} \geq V_{IO}$ – 200 mV during power-up.*
3. *V_{CC} and V_{IO} ramp rate could be non-linear.*
4. *V_{CC} and V_{IO} are recommended to be ramped up simultaneously.*
5. *All V_{CC} signals must be ramped simultaneously to ensure correct power-up.*
6. *V_{CC} ramp rate is > 1V/ 100 µs and for V_{CC} ramp rate of < 1 V /100 µs a hardware reset is required.*

Figure 11.5 V_{CC} Power-up Diagram



11.6 DC Characteristics

11.6.1 CMOS Compatible

Parameter	Description	Test Conditions (Note 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 (1)	$V_{OUT} = V_{SS}$ to V_{CC}, $V_{CC} = V_{CC}$max				±1	µA
I_{CCB}	V_{CC} Active burst Read Current	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 8	66 MHz		31	34	mA
			83 MHz		35	38	mA
			108 MHz		39	44	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 16	66 MHz		24	26	mA
			83 MHz		28	30	mA
			108 MHz		32	36	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = Continuous	66 MHz		24	26	mA
			83 MHz		28	30	mA
			108 MHz		32	36	mA
I_{CC1}	V_{CC} Active Asynchronous Read Current (3)	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}	10 MHz		40	80	mA
			5 MHz		20	40	mA
			1 MHz		10	20	mA
I_{CC2}	V_{CC} Active Write Current (4)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}	ACC		1	5	µA
			V_{CC}		30	40	mA
I_{CC3}	V_{CC} Standby Current (5) (6)	CE# = RESET# = V_{CC} ± 0.2V	ACC		1	5	µA
			V_{CC}		40	70	µA
I_{CC4}	V_{CC} Reset Current (6)	RESET# = $V_{IL,}$ CLK = V_{IL}			150	250	µA
I_{CC5}	V_{CC} Active Current (Read While Write) (6)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}	Asynchronous 5 MHz		50	60	mA
			66 MHz		61	66	mA
			83 MHz		65	70	mA
			108 MHz		71	76	mA
I_{CC6}	V_{CC} Sleep Current (6)	CE# = V_{IL}, OE# = V_{IH}			40	70	µA
I_{CC7}	V_{CC} Page Mode Read Current	OE# = V_{IH}, CE# = V_{IL}			10	15	mA
I_{ACC}	Accelerated Program Current (7)	CE# = V_{IL}, OE# = $V_{IH,}$ ACC = 9.5V	ACC		6	20	mA
			V_{CC}		14	20	mA
V_{IL}	Input Low Voltage	V_{CC} = 1.8V		–0.5		0.4	V
V_{IH}	Input High Voltage	V_{CC} = 1.8V		V_{CC} – 0.4		V_{CC} + 0.4	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_{HH}	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. *CE# must be set high when measuring the RDY pin.*
3. *The I_{CC} current listed is typically less than 3.5 mA/MHz, with OE# at V_{IH}.*
4. *I_{CC} active while Embedded Erase or Embedded Program is in progress.*
5. *Device enters automatic sleep mode when addresses are stable for t_{ACC} + 20 ns. Typical sleep mode current is equal to I_{CC3}.*
6. *$V_{IH} = V_{CC}$ ± 0.2V and V_{IL} > –0.1V.*
7. *Total current during accelerated programming is the sum of V_{ACC} and V_{CC} currents.*

11.7 AC Characteristics

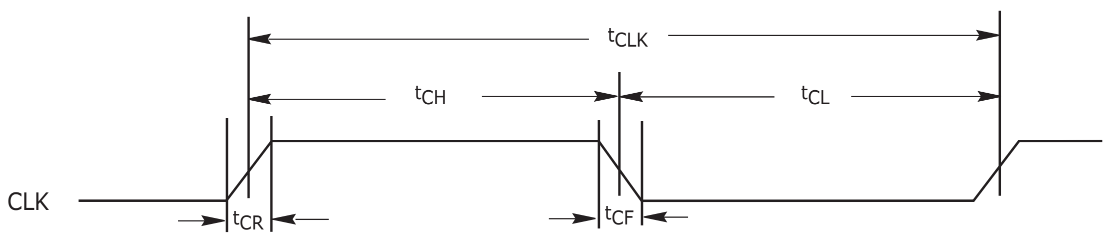
11.7.1 CLK Characterization

Table 11.3 CLK Characterization

Parameter	Description		66 MHz	83 MHz	108 MHz	Unit
f_{CLK}	CLK Frequency	Max	66	83	108	MHz
t_{CLK}	CLK Period	Min	15.1	12.0	9.26	ns
t_{CH}	CLK High Time	Min	0.4 t_{CLK}	0.4 t_{CLK}	0.4 t_{CLK}	ns
t_{CL}	CLK Low Time					
t_{CR}	CLK Rise Time	Max	3	2.5	1.85	ns
t_{CF}	CLK Fall Time					

Note:
1. Not 100% tested.

Figure 11.6 CLK Characterization



11.7.2 Synchronous/Burst Read

Table 11.4 Synchronous/Burst Read

Parameter		Description		66 MHz	83 MHz	108 MHz	Unit
JEDEC	Standard						
	t_{IACC}	Latency	Max	80			ns
	t_{BACC}	Burst Access Time Valid Clock to Output Delay	Max	11.2	9	7.6	ns
	t_{ACS}	Address Setup Time to CLK (Note 1)	Min	4		4	ns
	t_{ACH}	Address Hold Time from CLK (Note 1)	Min	6		6	ns
	t_{BDH}	Data Hold Time from Next Clock Cycle	Min	3		2	ns
	t_{CR}	Chip Enable to RDY Valid	Max	11.2	9	7.6	ns
	t_{OE}	Output Enable to RDY Low	Max	11.2			ns
	t_{CEZ}	Chip Enable to High-Z (Note 2)	Max	10			ns
	t_{OEZ}	Output Enable to High-Z (Note 2)	Max	10			ns
	t_{CES}	CE# Setup Time to CLK	Min	4		4	ns
	t_{RDYS}	RDY Setup Time to CLK	Min	4	3.5	1.66	ns
	t_{RACC}	Ready Access Time from CLK	Max	11.2	9	7.6	ns
	t_{AVDS}	AVD# Low to CLK	Min	4			ns
	t_{AVDP}	AVD# Pulse	Min	7			ns
	t_{AVDH}	AVD# Hold	Min	3			ns
	f_{CLK}	Minimum clock frequency	Min	1	1	1	MHz

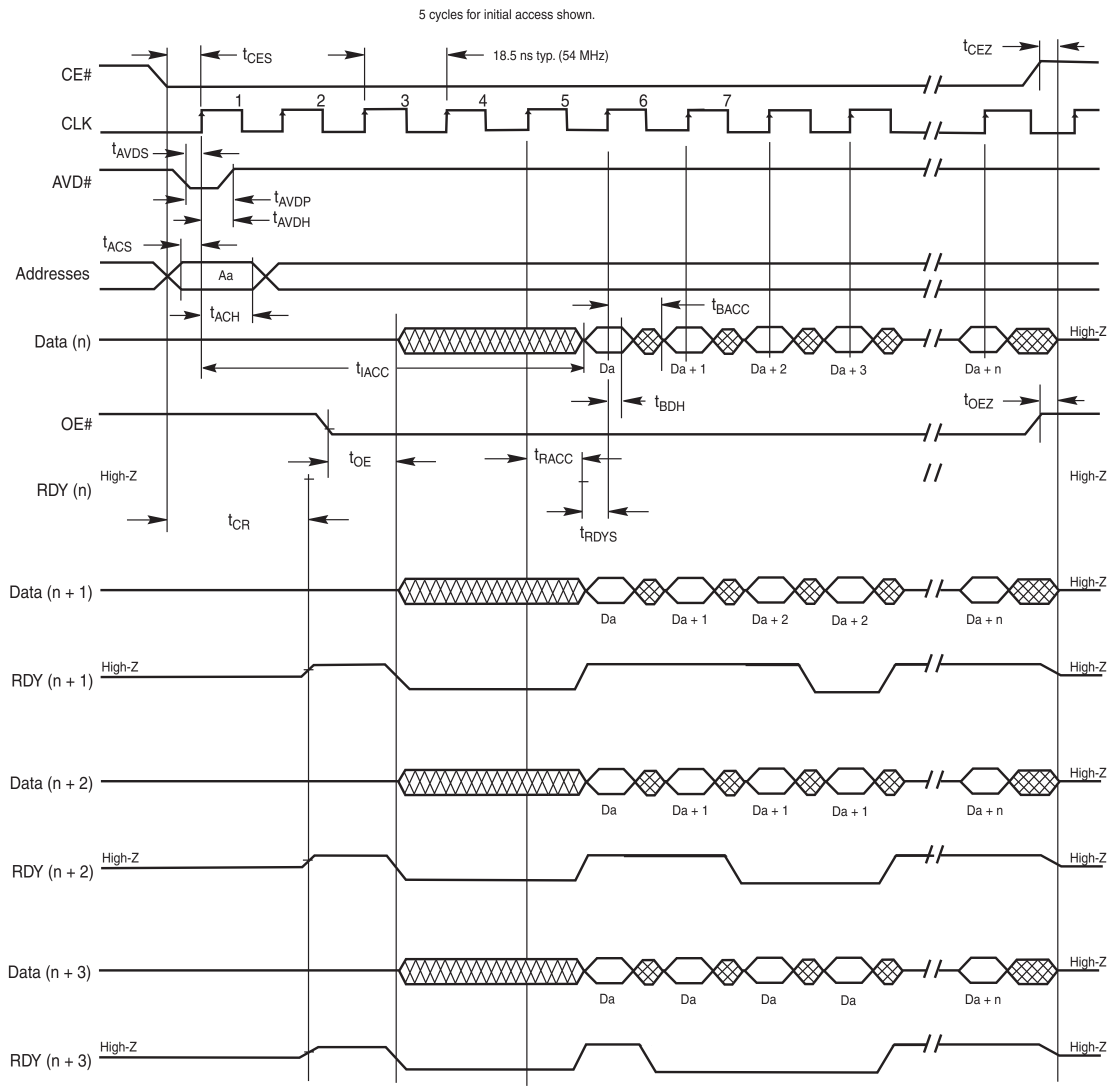
Notes:
1. Addresses are latched on the first rising edge of CLK.

2. Not 100% tested.

Table 11.5 Synchronous Wait State Requirements

Wait State	Frequency Setting (MHz)
3	27
4	40
5	54
6	66
7	80
8	95
9	108

Figure 11.7 CLK Synchronous Burst Mode Read

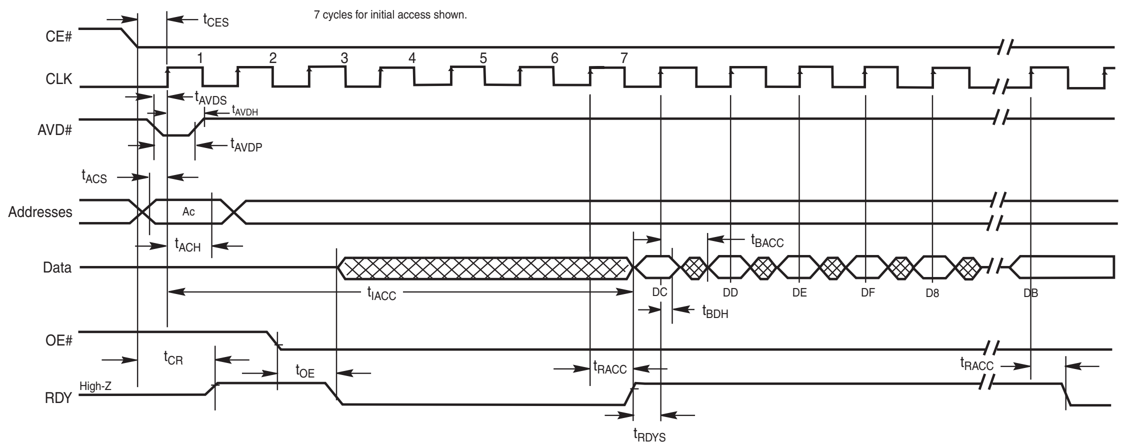


Notes:

1. Figure shows total number of wait states set to five cycles. The total number of wait states can be programmed from two cycles to seven cycles.

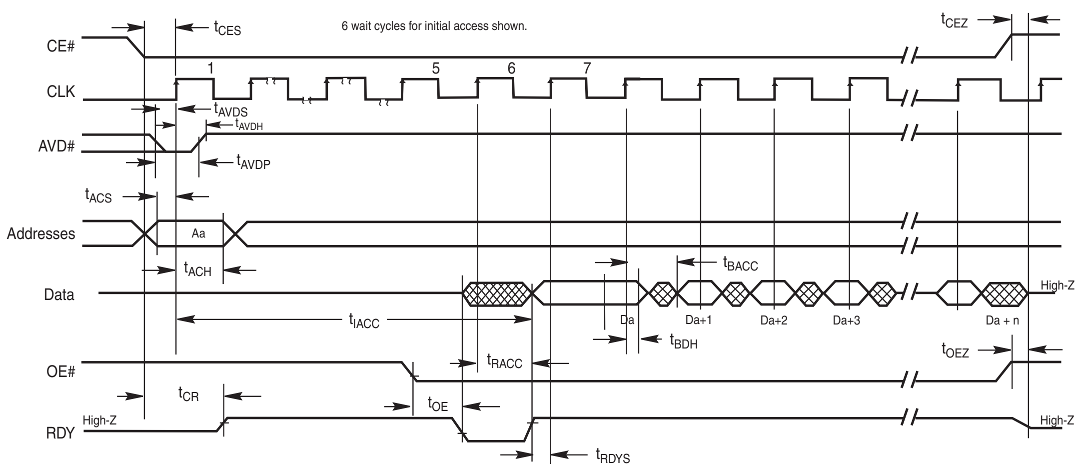
2. If any burst address occurs at “address + 1”, “address + 2”, or “address + 3”, additional clock delay cycles are inserted, and are indicated by RDY.
3. The device is in synchronous mode.

Figure 11.8 8-word Linear Burst with Wrap Around



Notes:
1. Figure shows total number of wait states set to seven cycles. The total number of wait states can be programmed from two cycles to seven cycles.
2. If any burst address occurs at “address + 1”, “address + 2”, or “address + 3”, additional clock delay cycles are inserted, and are indicated by RDY.
3. The device is in synchronous mode with wrap around.
4. D8–DF in data waveform indicate the order of data within a given 8-word address range, from lowest to highest. Starting address in figure is the 4th address in range (0-F).

Figure 11.9 Linear Burst with RDY Set One Cycle Before Data



Notes:
1. Figure assumes 6 wait states for initial access and synchronous read.
2. The Set Configuration Register command sequence has been written with CR8=0; device outputs RDY one cycle before valid data.

11.7.3 AC Characteristics-Asynchronous Read

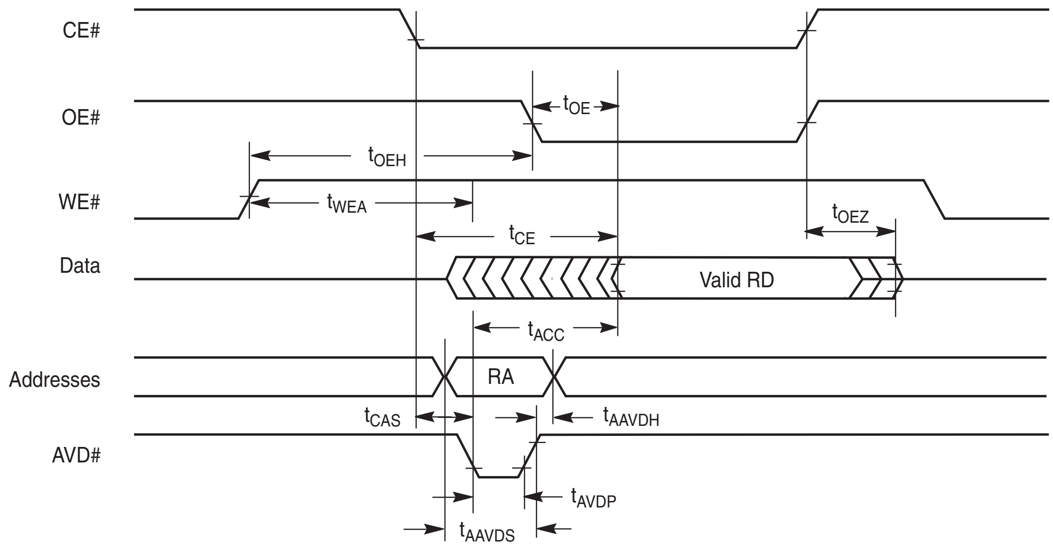
Table 11.6 AC Characteristics-Asynchronous Read

Parameter		Description			66 MHz	83 MHz	108 MHz	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	8			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	6			ns
	t_{OE}	Output Enable to Output Valid		Max	18			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 (Note 1)		Max	10			ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0			ns
	t_{PACC}	Page Access Time		Max	20			ns
	t_{OH}	Output Hold Time From Addresses, CE# or OE#, whichever occurs first		Min	0			ns
	t_{CR}	Chip Enable to RDY Valid		Max	10			ns
	t_{CEZ}	CE# disable to Output High-Z		Max	10			ns
	t_{WEA}	WE# Disable to AVD#		Min	9.6			ns

Notes:

1. Not 100% tested.

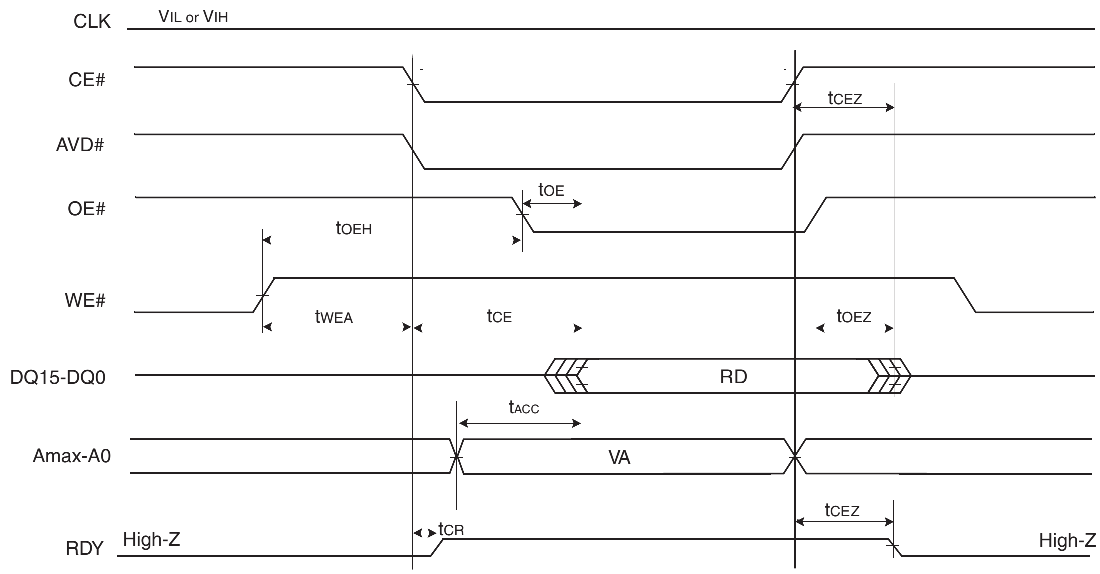
Figure 11.10 Asynchronous Mode Read



Notes:

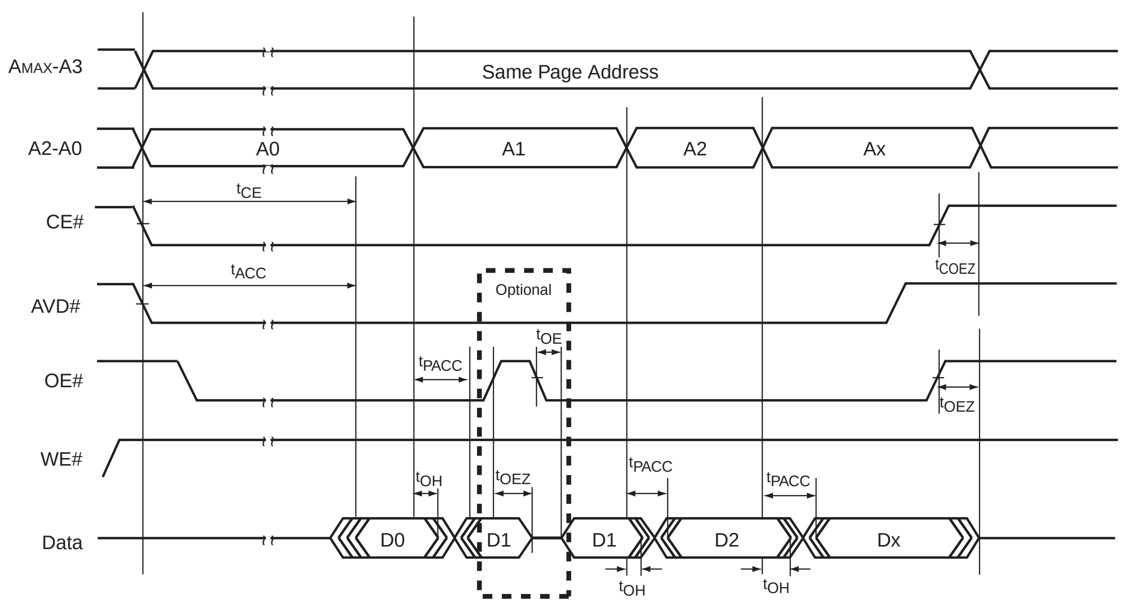
RA = Read Address, RD = Read Data.

Figure 11.11 Asynchronous Mode Read (AVD# tied to CE#)



Notes:
1. *AVD# is tied to CE#*
2. *VA = Valid Read Address, RD = Read Data.*

Figure 11.12 Asynchronous Page-Mode Read

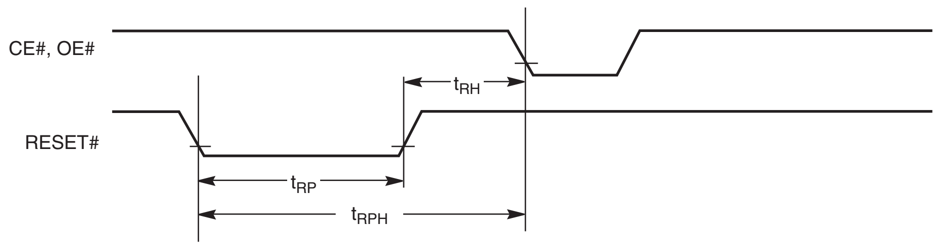


11.7.4 Hardware Reset (RESET#)

Parameter		Description		All Speed Options	Unit
JEDEC	Std.				
	t_{RP}	RESET# Pulse Width	Min	50	ns
	t_{RH}	Reset High Time Before Read (See Note)	Min	200	ns
	t_{RPH}	RESET# Low to CE# Low	Min	10	µs

Note:
Not 100% tested.

Figure 11.13 Reset Timings

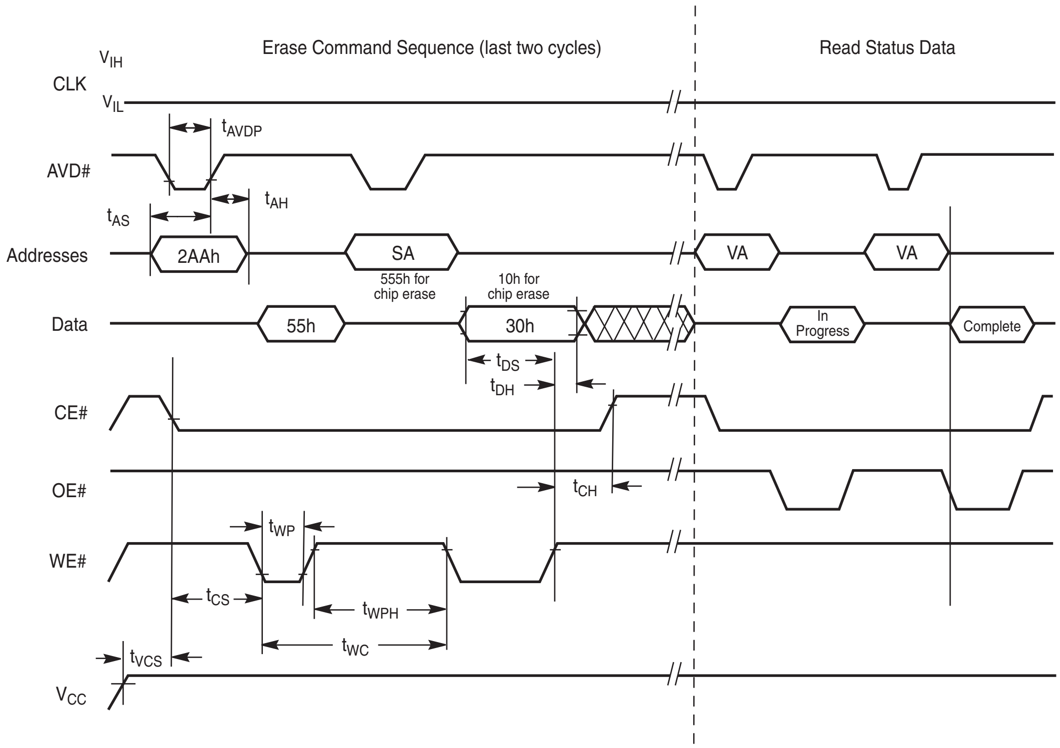


11.7.5 Erase/Program Timing

Parameter		Description			66 MHz	83 MHz	108 MHz	Unit
JEDEC	Standard							
t_{AVAV}	t_{WC}	Write Cycle Time (1)		Min	60			ns
t_{AVWL}	t_{AS}	Address Setup Time (2) (3)	Synchronous	Min	4			ns
			Asynchronous		4			ns
t_{WLAX}	t_{AH}	Address Hold Time (2) (3)	Synchronous	Min	3.5			ns
			Asynchronous		3.5			
	t_{AVDP}	AVD# Low Time		Min	6			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		Min	0			ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0			ns
t_{WHEH}	t_{CH}	CE# Hold Time		Min	0			ns
t_{WLWH}	t_{WP}	Write Pulse Width		Min	25			ns
t_{WHWL}	t_{WPH}	Write Pulse Width High		Min	20			ns
	$t_{SR/W}$	Latency Between Read and Write Operations		Min	0			ns
	t_{CR}	Chip Enable to RDY Valid		Max	10			ns
	t_{CEZ}	CE# disable to Output High-Z		Max	10			ns
	t_{VID}	V_{ACC} Rise and Fall Time		Min	500			ns
	t_{VIDS}	V_{ACC} Setup Time (During Accelerated Programming)		Min	1			µs
t_{ELWL}	t_{CS}	CE# Setup Time to WE#		Min	4			ns
	t_{AVSC}	AVD# Setup Time to CLK		Min	5			ns
	t_{AVHC}	AVD# Hold Time to CLK		Min	5			ns
	t_{CSW}	Clock Setup Time to WE#		Min	5			ns
	t_{WEP}	Noise Pulse Margin on WE#		Max	3			ns
	t_{ESL}	Erase Suspend Latency		Max	30			µs
	t_{PSL}	Program Suspend Latency		Max	30			µs
	t_{ASP}	Toggle Time During Erase within a Protected Sector		Typ	20			µs
	t_{PSP}	Toggle Time During Programming Within a Protected Sector		Typ	20			µs

Notes:
1. *Not 100% tested.*
2. *Asynchronous read mode allows Asynchronous program operation only. Synchronous read mode allows both Asynchronous and Synchronous program operation.*
3. *In asynchronous program operation timing, addresses are latched on the rising edge of AVD# or WE#. In synchronous program operation timing, addresses are latched on the rising edge of CLK.*
4. *See Section 11.7.6, Erase and Programming Performance on page 73 for more information.*
5. *Does not include the preprogramming time.*

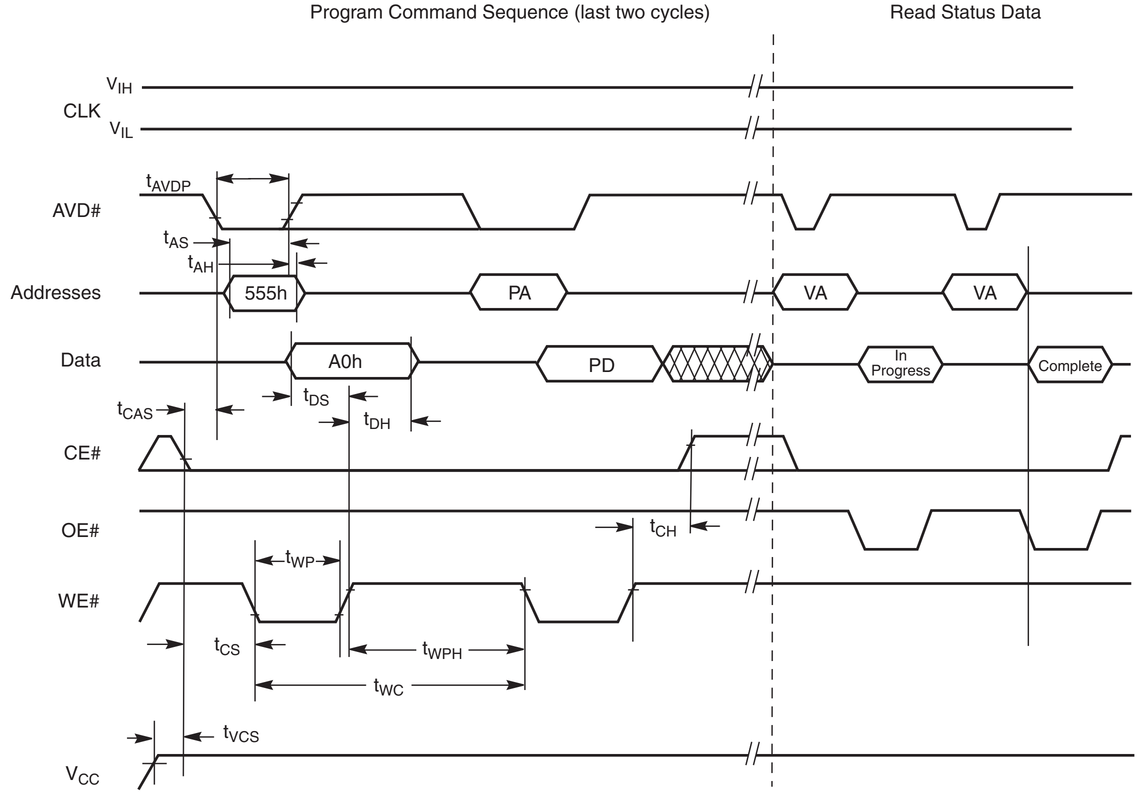
Figure 11.14 Chip/Sector Erase Operation Timings



Note:

1. Addresses latched by rising edge of AVD#.

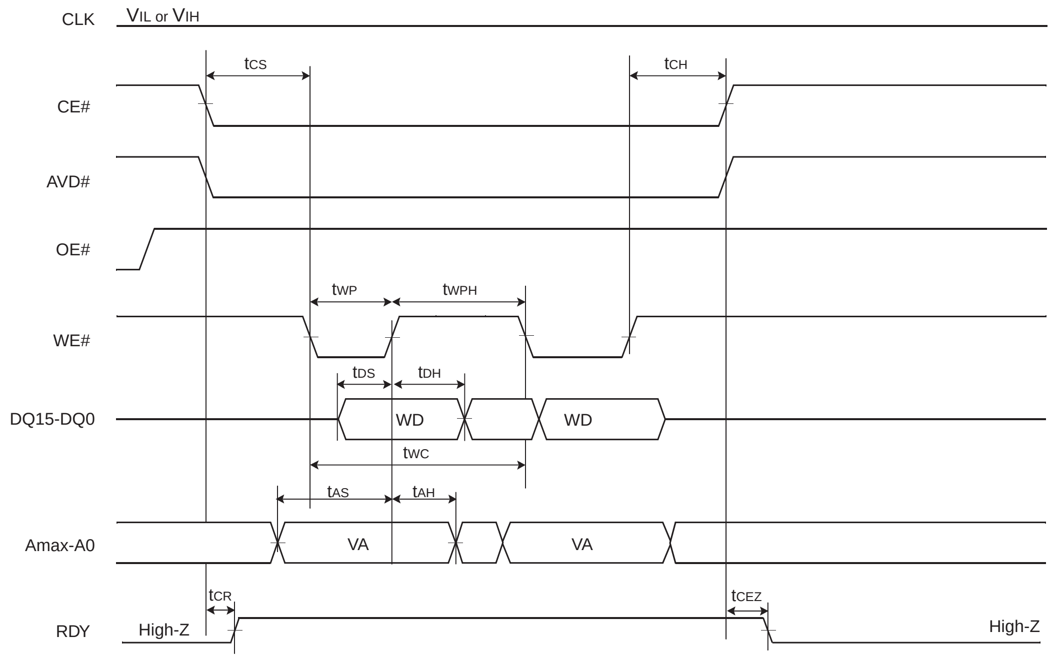
Figure 11.15 Program Operation Timing Using AVD#



Notes:

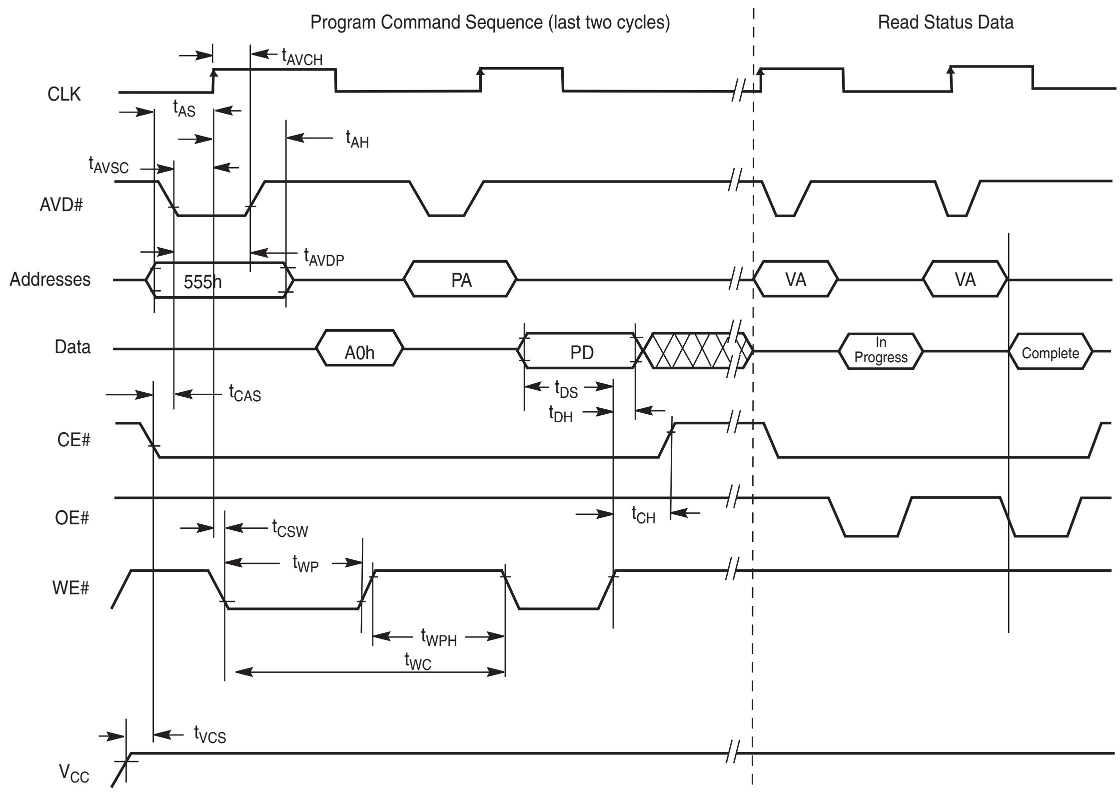
1. *PA = Program Address, PD = Program Data, VA = Valid Address for reading status bits.*
2. *"In progress" and "complete" refer to status of program operation.*
3. *A21–A14 are don't care during command sequence unlock cycles.*
4. *CLK can be either V_{IL} or V_{IH}.*
5. *The Asynchronous programming operation is independent of the Set Device Read Mode bit in the Configuration Register.*
6. *Addresses latched by rising edge of AVD#.*

Figure 11.16 Asynchronous Program Operation (AVD# Tied to CE#)



Notes:
1. *VA = Valid Read Address, WD = Write Data.*
2. *Addresses and data latched by rising edge of WE#.*

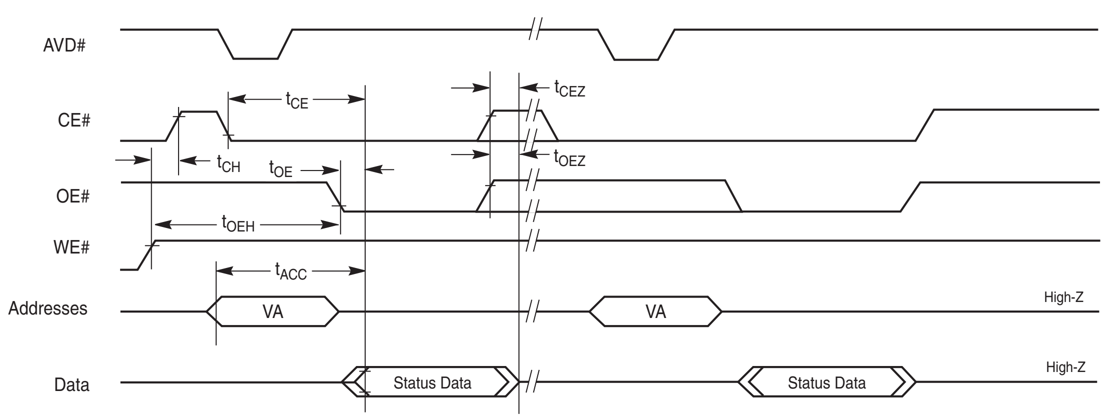
Figure 11.17 Program Operation Timing Using CLK in Relationship to AVD#



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. *A21–A14 are don't care during command sequence unlock cycles.*
4. *Addresses are latched on the first rising edge of CLK.*
5. *Either CE# or AVD# is required to go from low to high in between programming command sequences.*
6. *The Synchronous programming operation is dependent of the Set Device Read Mode bit in the Configuration Register. The Configuration Register must be set to the Synchronous Read Mode.*

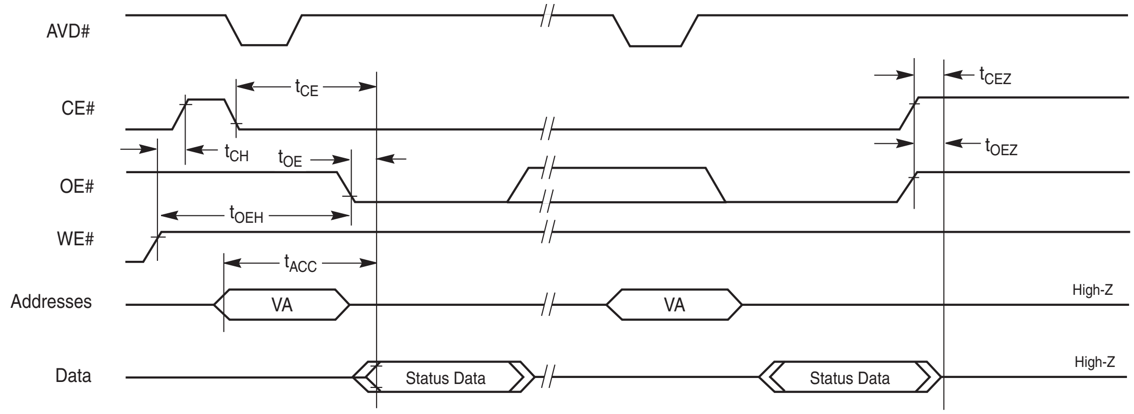
Figure 11.18 Data# Polling Timings (During Embedded Algorithm)



Notes:

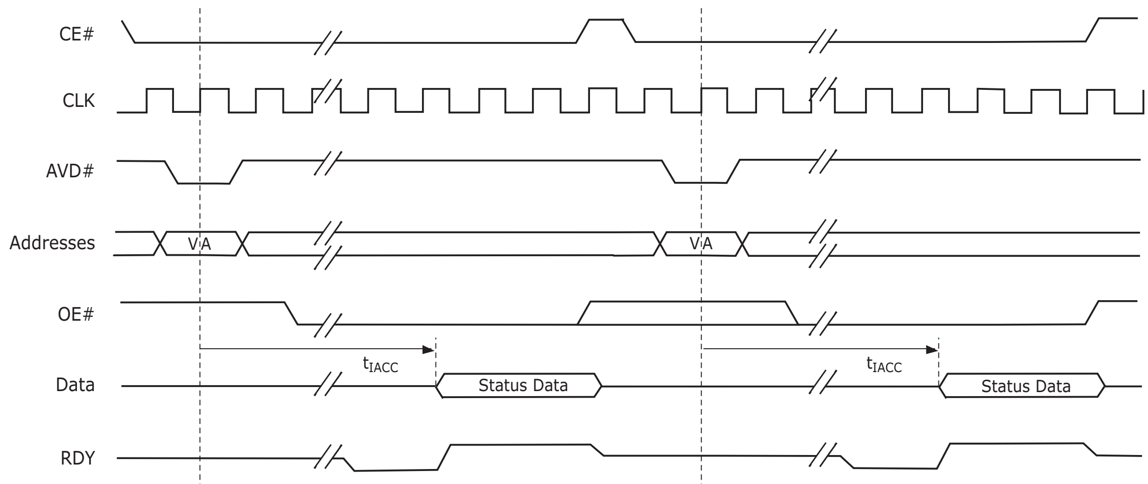
1. *Status reads in figure are shown as asynchronous.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, and Data# Polling outputs true data.*

Figure 11.19 Toggle Bit Timings (During Embedded Algorithm)



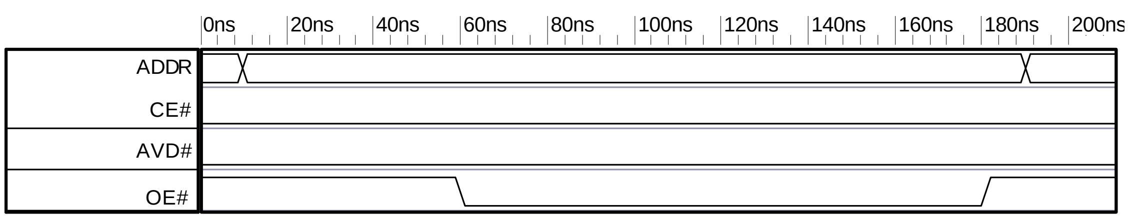
Notes:
1. *Status reads in figure are shown as asynchronous.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, the toggle bits stop toggling.*

Figure 11.20 Synchronous Data Polling Timings/Toggle Bit Timings



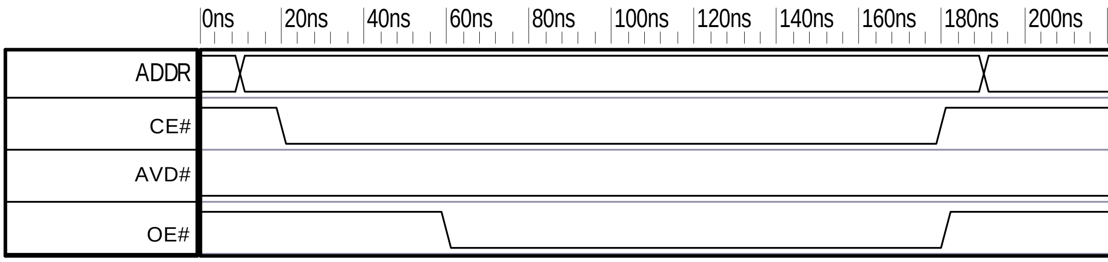
Notes:
1. *The timings are similar to synchronous read timings.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, the toggle bits stop toggling.*
3. *RDY is active with data (D8 = 1 in the Configuration Register). When D8 = 0 in the Configuration Register, RDY is active one clock cycle before data.*

Figure 11.21 Conditions for Incorrect DQ2 Polling During Erase Suspend



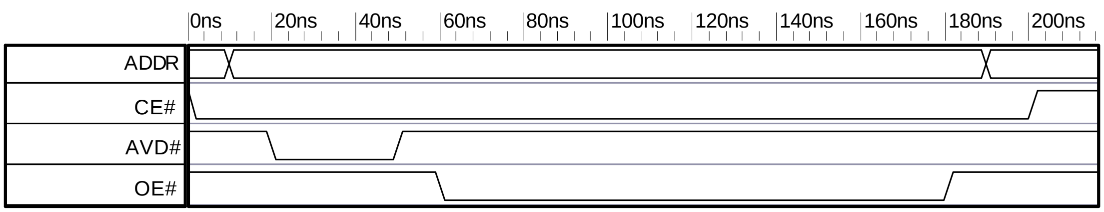
Note:
DQ2 does not toggle correctly during erase suspend if AVD# or CE# are held low after valid address.

Figure 11.22 Correct DQ2 Polling during Erase Suspend #1



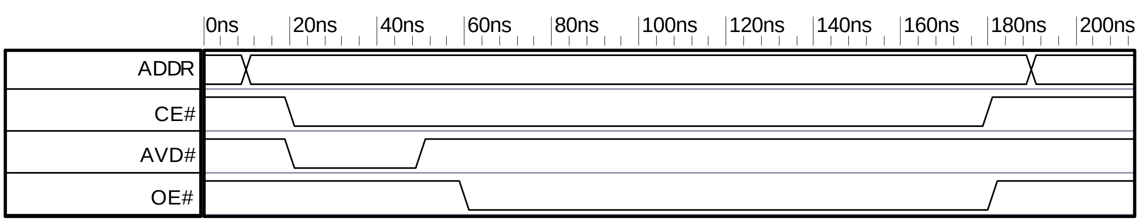
Note:
DQ2 polling during erase suspend behaves normally if CE# pulses low at or after valid Address, even if AVD# does not.

Figure 11.23 Correct DQ2 Polling during Erase Suspend #2



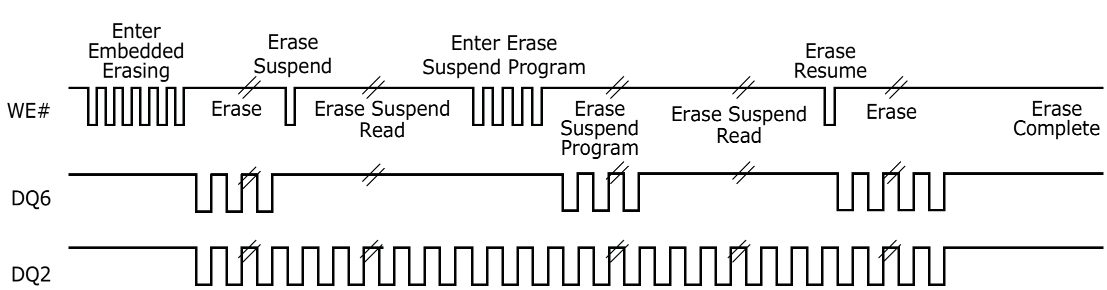
Note:
DQ2 polling during erase suspend behaves normally if AVD# pulses low at or after valid Address, even if CE# does not.

Figure 11.24 Correct DQ2 Polling during Erase Suspend #3



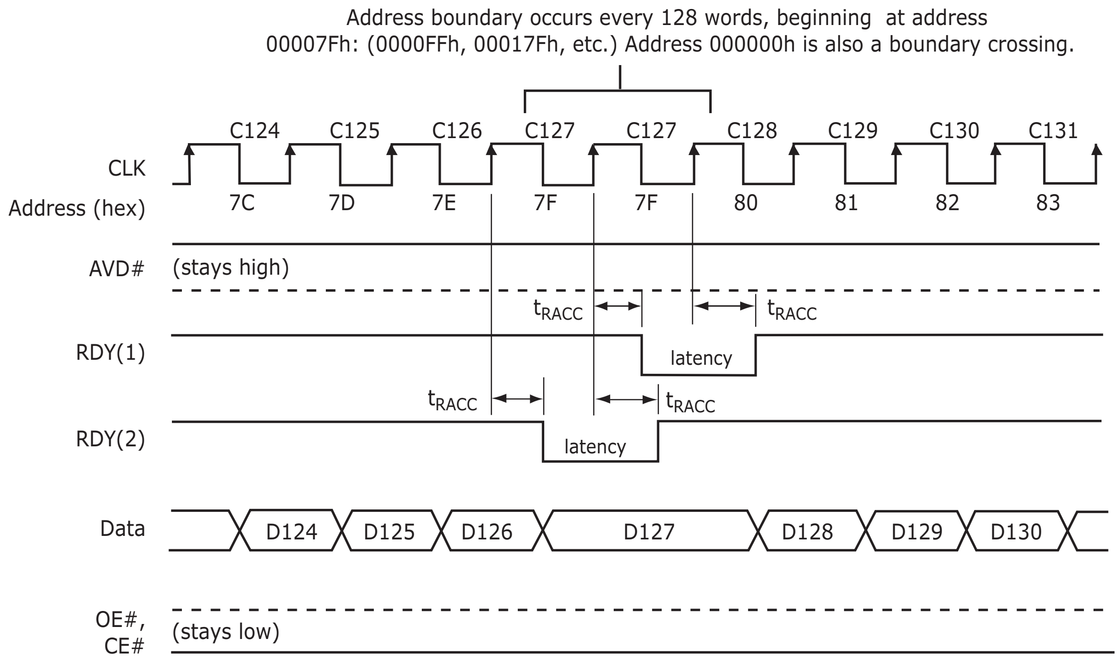
Note:
DQ2 polling during erase suspend behaves normally if both AVD# and CE# pulse low at or after valid Address.

Figure 11.25 DQ2 vs. DQ6



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

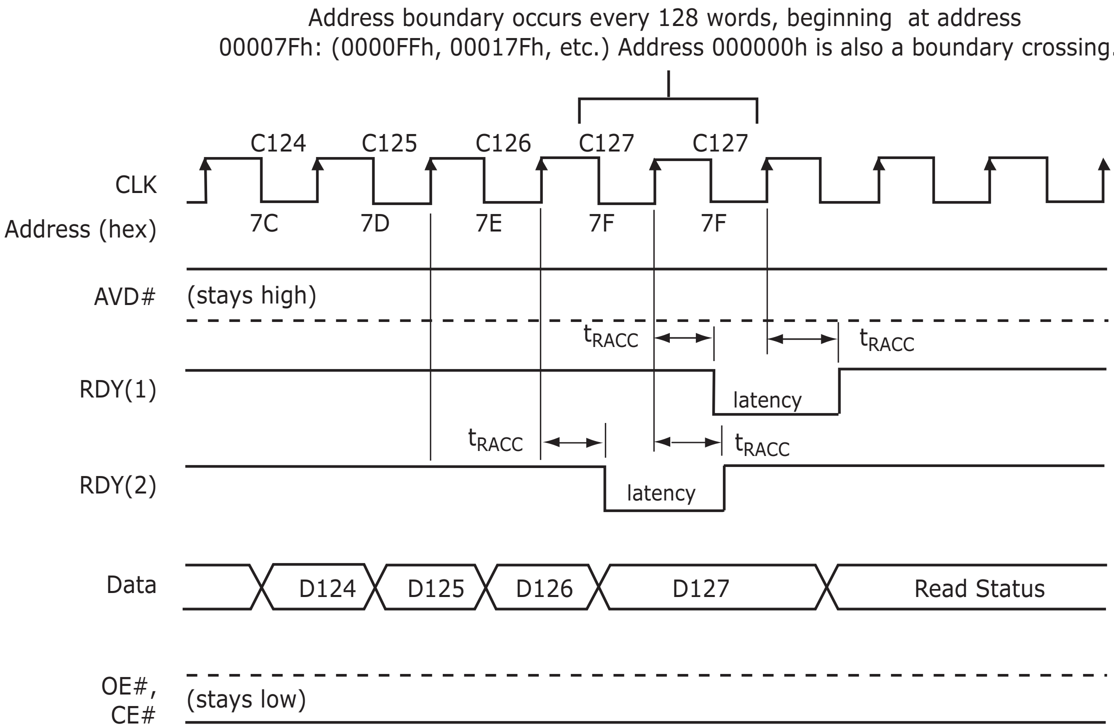
Figure 11.26 Latency with Boundary Crossing



Notes:

1. *RDY(1) active with data (CR8 = 1 in the Configuration Register).*
2. *RDY(2) active one clock cycle before data (CR8 = 0 in the Configuration Register).*
3. *Cxx indicates the clock that triggers Dxx on the outputs; for example, C60 triggers D60.*
4. *Figure shows the device not crossing a bank in the process of performing an erase or program.*

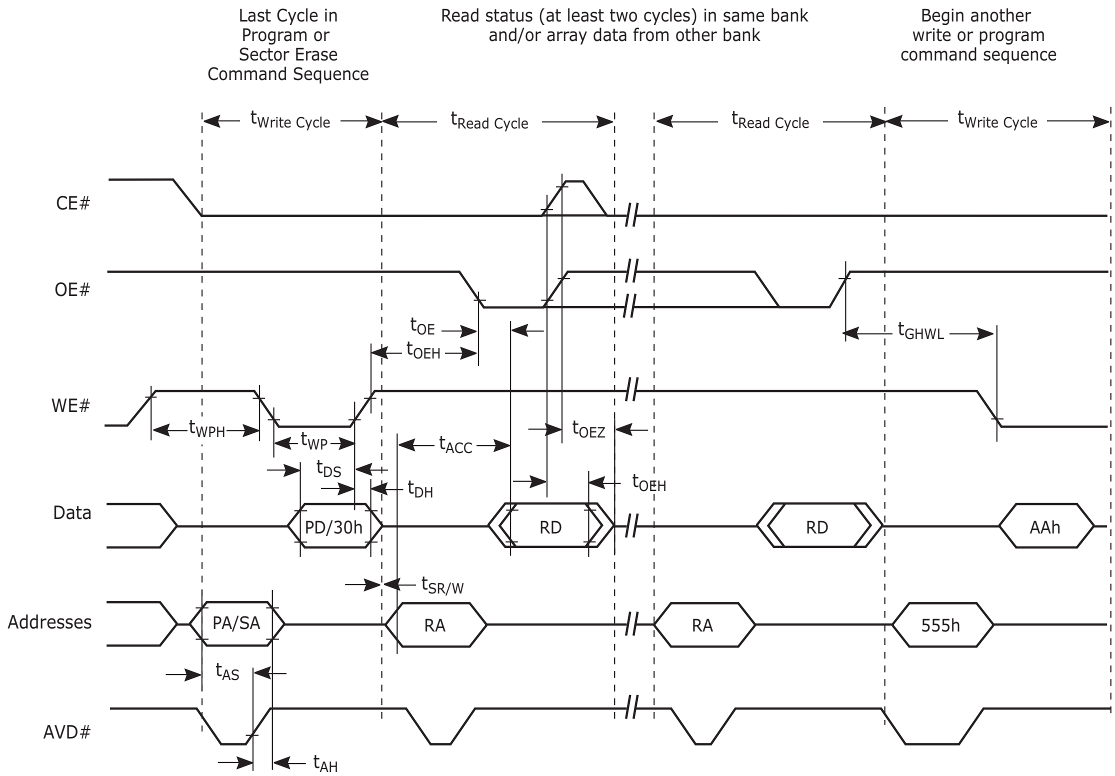
Figure 11.27 Latency with Boundary Crossing into Program/Erase Bank



Notes:

1. *RDY(1) active with data (CR8 = 1 in the Configuration Register).*
2. *RDY(2) active one clock cycle before data (CR8 = 0 in the Configuration Register).*
3. *Cxx indicates the clock that triggers Dxx on the outputs; for example, C60 triggers D60.*
4. *Figure shows the device crossing a bank in the process of performing an erase or program.*

Figure 11.28 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.

11.7.6 Erase and Programming Performance

Parameter		Typ (Note 1)	Max (Note 2)	Unit	Comments
Sector Erase Time	32 kword	0.8	3.5	s	Excludes 00h programming prior to erasure (Note 4)
	8 kword	0.35	2		
Chip Erase Time	V_{CC}	103	453		
	ACC	103	453		
Single Word Programming Time (Note 6)	V_{CC}	170	800	µs	
Effective 32-Word Buffer Programming Time	V_{CC}	14.1	94		
	ACC	9	60		
Total 32-Word Buffer Programming Time	V_{CC}	450	3000		
	ACC	288	1920		
Chip Programming Time (Note 3)	V_{CC}	59	78.6	s	Excludes system level overhead (Note 5)
	ACC	38	52		

Note:

1. *Typical program and erase times assume the following conditions: 25°C, 1.8V V_{CC}, 10,000 cycles; checkerboard data pattern.*
2. *Under worst case conditions of -25°C, V_{CC} = 1.70V, 100,000 cycles.*
3. *Typical chip programming time is considerably less than the maximum chip programming time listed, and is based on utilizing the Write Buffer.*
4. *In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 00h before erasure.*
5. *System-level overhead is the time required to execute the two- or four-bus-cycle sequence for the program command. See Section 12., Appendix on page 74 for further information on command definitions.*
6. *Word programming specification is based upon a single word programming operation not utilizing the write buffer.*

11.7.7 BGA Ball Capacitance

Parameter Symbol	Parameter Description	Test Setup	Typ.	Max	Unit
C_{IN}	Input Capacitance	$V_{IN} = 0$	5.3	6.3	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0$	5.8	6.8	pF
C_{IN2}	Control Pin Capacitance	$V_{IN} = 0$	6.3	7.3	pF

Notes:

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

12. Appendix

Table 12.1 Memory Array Commands

Command Sequence (Notes)		Cycles	Bus Cycles (1), (2), (3), (4), (5)											
			First		Second		Third		Fourth		Fifth		Sixth	
			Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Asynchronous Read (6)		1	RA	RD										
Reset (7)		1	XXX	F0										
Auto-select(8)	Manufacturer ID	4	555	AA	2AA	55	[BA]555	90	[BA]X00	0001				
Auto-select(8)	Device ID (9)	6	555	AA	2AA	55	[BA]555	90	[BA]X01	007E	BA+X0E	Data	BA+X0F	0000
Auto-select(8)	Indicator Bits (10)	4	555	AA	2AA	55	[BA]555	90	[BA]X07	Data				
Program		4	555	AA	2AA	55	555	A0	PA	PD				
Write to Buffer (11)		6	555	AA	2AA	55	PA	25	PA	WC	PA	PD	WBL	PD
Program Buffer to Flash		1	SA	29										
Write to Buffer Abort Reset (12)		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/Program Suspend (13)		1	BA	B0										
Erase/Program Resume (14)		1	BA	30										
Set Configuration Register (17)		4	555	AA	2AA	55	555	D0	X00	CR				
Read Configuration Register		4	555	AA	2AA	55	555	C6	X00	CR				
CFI Query (15)		1	55	98										
Secured Silicon Sector	Entry	3	555	AA	2AA	55	555	88						
Secured Silicon Sector	Program (16)	4	XX	A0	PA	PD								
Secured Silicon Sector	Read (16)	1	RA	Data										
Secured Silicon Sector	Exit (16)	4	555	AA	2AA	55	555	90	XXX	00				

Legend:

X = Don't care.

RA = Read Address.

RD = Read Data.

PA = Program Address. Addresses latch on the rising edge of the AVD# pulse or active edge of CLK, whichever occurs first.

PD = Program Data. Data latches on the rising edge of WE# or CE# pulse, whichever occurs first.

SA = Sector Address: WS064R = A21–A13.

BA = Bank Address: WS064R = A21–A20.

CR = Configuration Register data bits D15–D0.

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 7.1 on page 21 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Shaded cells indicate read cycles.*
4. *Address and data bits not specified in table, legend, or notes are don't cares (each hex digit implies 4 bits of data).*
5. *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.*
6. *No unlock or command cycles required when bank is reading array data.*
7. *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) or performing sector lock/unlock.*
8. *The system must provide the bank address. See See Autoselect on page 28. for more information.*
9. *Data in cycle 5 is 004F for Top Boot devices and 0057 for Bottom Boot devices.*
10. *See Table 7.12 on page 28 for indicator bit values.*
11. *Total number of cycles in the command sequence is determined by the number of words written to the write buffer.*
12. *Command sequence resets device for next command after write-to-buffer operation.*
13. *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.*

14. Erase Resume command is valid only during the Erase Suspend mode, and requires the bank address.
15. Command is valid when device is ready to read array data or when device is in autoselect mode.
16. Requires Entry command sequence prior to execution. Secured Silicon Sector Exit Reset command is required to exit this mode; device may otherwise be placed in an unknown state.
17. Requires reset command to configure the Configuration Register.

Table 12.2 Sector Protection Commands

Command Sequence (Notes)		Cycles	Bus Cycles (1), (2), (3), (4)													
			First		Second		Third		Fourth		Fifth		Sixth		Seventh	
			Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Lock Register Bits	Command Set Entry (5)	3	555	AA	2AA	55	555	40								
	Program	2	XX	A0	00	data										
	Read	1	00	data												
	Command Set Exit (6)	2	XX	90	XX	00										
Volatile Sector Protection (DYB)	Command Set Entry (5)	3	555	AA	2AA	55	[BA]555	E0								
	DYB Set	2	XX	A0	SA	00										
	DYB Clear	2	XX	A0	SA	01										
	DYB Status Read	1	SA	RD(0)												
	Command Set Exit (6)	2	XX	90	XX	00										

Legend:
X = Don't care.
RA = Address of the memory location to be read.
SA = Sector Address.
BA = Bank Address.
RD(0) = DQ0 protection indicator bit. If protected, DQ0 = 0. If unprotected, DQ0 = 1.

Notes:
1. *All values are in hexadecimal.*
2. *Shaded cells indicate read cycles.*
3. *Address and data bits not specified in table, legend, or notes are don't cares (each hex digit implies 4 bits of data).*
4. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*
5. *Entry commands are required to enter a specific mode to enable instructions only available within that mode.*
6. *Exit command must be issued to reset the device into read mode; device may otherwise be placed in an unknown state.*

12.1 Common Flash Memory Interface

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

This device enters the CFI Query mode when the system writes the CFI Query command, 98h, to address 55h any time the device is ready to read array data. The system can read CFI information at the addresses given in Table 12.3 through Table 12.6 within that bank. All reads outside of the CFI address range, within the bank, returns non-valid data. Reads from other banks are allowed, writes are not. To terminate reading CFI data, the system must write the reset command.

The following is a C source code example of using the CFI Entry and Exit functions. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: CFI Entry command */
  *( (UINT16 *)bank_addr + 0x55 ) = 0x0098;    /* write CFI entry command         */

/* Example: CFI Exit command  */
  *( (UINT16 *)bank_addr + 0x000 ) = 0x00F0;    /* write CFI exit command          */
```

For further information, please refer to the CFI Specification (see JEDEC publications JEP137-A and JESD68.01and CFI Publication 100). Please contact your sales office for copies of these documents.

Table 12.3 ID/CFI Data

	Addresses	Data	Description
Device Identification	(SA) + 00h	0001h	Spansion Manufacturer ID
	(SA) + 01h	007Eh	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)	Sector Protect Verify
	(SA) + 03h	0000h	Reserved
	(SA) + 04h	00FFh	Reserved
	(SA) + 05h	00FFh	Reserved
	(SA) + 06h	0010h	ID Version
	(SA) + 07h	00BFh	Indicator Bits DQ15 - DQ8 = Reserved DQ7 (Factory Lock Bit): 1 = Locked; 0 = Not Locked DQ6 (Customer Lock Bit): 1 = Locked; 0 = Not locked DQ5 - DQ0 = Reserved
	(SA) + 08h	00FFh	Reserved
	(SA) + 09h	00FFh	Reserved
	(SA) + 0Ah	00FFh	Reserved
	(SA) + 0Bh	00FFh	Reserved
	(SA) + 0Ch	00F2h	Lower Software Bits Bit 0 - Status Register Support 1 = Status Register Supported 0 = Status register not Supported Bit 1 - DQ Polling Support 1 = DQ bits polling supported 0 = DQ bits polling not supported Bit 3-2 - Command Set Support 11 = Reserved 10 = Reserved 01 = Reduced Command Set 00 = Old Command Set Bit 4-F - 00Fh - Reserved
	(SA) + 0Dh	00FFh	Upper Software Bits Reserved
	(SA) + 0Eh	004Fh (Top) / 0057h (Bottom)	High Order Device ID, Word 2
	(SA) + 0Fh	0000h	Low Order Device ID, Word 3

Table 12.4 CFI Query Identification String

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

Table 12.5 System Interface String

Addresses	Data	Description
1Bh	0017h	V_{CC} Min. (write/erase) D7–D4: volt, D3–D0: 100 millivolt
1Ch	0019h	V_{CC} Max. (write/erase) 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 12.6 Device Geometry Definition

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

Table 12.7 Primary Vendor-Specific Extended Query

Addresses	Data	Description
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 Read Only, 2 = To Read and Write
47h	0001h	Sector Protect 0 = Not Supported, X = Number of sectors in per group
48h	0000h	Sector Temporary Unprotect 00 = Not Supported, 01 = Supported
49h	0008h	Sector Protect/Unprotect scheme 08 = Advanced Sector Protection
4Ah	0020h	Simultaneous Operation 00 = Not Supported, X = Number of Sectors in all banks except boot bank
4Bh	0001h	Burst Mode Type 00 = Not Supported, 01 = Supported
4Ch	0001h	Page Mode 00 = Not Supported, 01 = Supported
4Dh	0085h	ACC (Acceleration) Supply Minimum 00h = Not Supported, D7-D4: Volt, D3-D0: 100 mV
4Eh	0095h	ACC (Acceleration) Supply Maximum 00h = Not Supported, D7-D4: Volt, D3-D0: 100 mV
4Fh	0003h (top boot)	Top/Bottom Boot Sector Flag 01h = Top/Middle Boot Device, 02h = Bottom Boot Device, 03h = Top Boot Device
	0002h (bottom boot)	
50h	0001h	Program Suspend. 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 Suspend Time-out Maximum 2^N ns
57h	0004h	Bank Organization: X = Number of banks
58h	0020h (top boot)	Bank 0 Region Information. X = Number of sectors in banks
	0023h (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)	Bank 3 Region Information. X = Number of sectors in banks
	0020h (bottom boot)	

13. Revision History

Spansion Publication Number: S29WS064R

Section	Description
Revision 01 (April 9, 2010)	
	Initial release.
Revision 02 (August 6, 2010)	
Global	Removed 54 MHz speed option.
Distinctive Characteristics	Add page mode feature.
Product Overview	Corrected typo in table: S29WS064R Sector and Memory Address Map (Bottom Boot).
DC Characteristics: CMOS Compatible	Changed values for I_{CCB}, I_{CC2}, and I_{CC5}.
AC Characteristics: Erase and Programming Performance	Changed Typ and Max values for Single Word Programming Time. Changed Typ values for buffer and chip programming times.
Synchronous/Burst Read	Updated t_{OE} description.
AC Characteristics-Asynchronous Read	Updated t_{OE} value.
Revision 03 (September 30, 2010)	
DC Characteristics	CMOS Compatible table: Changed typical values for I_{CC3} and I_{CC6}.
Revision 04 (December 9, 2010)	
Global	Added references to Industrial Specification Supplement.
Revision 05 (July 22, 2011)	
Factory Secured Silicon Sector Erase and Programming Performance	Added sentence to indicate that sector is unprogrammed by default. Corrected note 2 for worst case condition temperature.
Revision 06 (May 29, 2012)	
Appendix	Memory Array Commands table: In the legend, corrected “SA = A21 - A14” to “SA = A21 - A13”

Document History Page

Document Title: S29WS064R 64 Mbit (4M x 16-bit), 1.8 V MirrorBit® Flash Document Number: 002-00755				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	--	WIOB	04/09/2010	Initial release
*A	--	WIOB	08/06/2010	Global: Removed 54 MHz speed option. Distinctive Characteristics: Add page mode feature. Product Overview: Corrected typo in table: S29WS064R Sector and Memory Address Map (Bottom Boot). DC Characteristics: CMOS Compatible: Changed values for I_{CCB}, I_{CC2}, and I_{CC5}. AC Characteristics: Erase and Programming Performance: Changed Typ and Max values for Single Word Programming Time. Changed Typ values for buffer and chip programming times. Synchronous/Burst Read: Updated tOE description. AC Characteristics-Asynchronous Read: Updated tOE value.
*B	--	WIOB	09/18/2010	DC Characteristics: CMOS Compatible table - Changed typical values for I_{CC3} and I_{CC6}.

Document Title: S29WS064R 64 Mbit (4M x 16-bit), 1.8 V MirrorBit® Flash Document Number: 002-00755				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
*C	--	WIOB	12/09/2010	Global: Added references to Industrial Specification Supplement.
*D	--	WIOB	07/22/2011	Factory Secured Silicon Sector Erase and Programming Performance: Added sentence to indicate that sector is unprogrammed by default. Corrected note 2 for worst case condition temperature.
*E	--	WIOB	05/29/2012	Appendix: Memory Array Commands table: In the legend, corrected “SA = A21 - A14” to “SA = A21 - A13”
*F	4952008	WIOB	10/08/2015	Updated to Cypress template.

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

Automotive.................................cypress.com/go/automotive

Clocks & Buffers .... cypress.com/go/clocks

Interface.. cypress.com/go/interface

Lighting & Power Control...........cypress.com/go/powerpsoc

Memory.. cypress.com/go/memory

PSoC ....cypress.com/go/psoc

Touch Sensing .... cypress.com/go/touch

USB Controllers...................................cypress.com/go/USB

Wireless/RF .... cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support