S29WSxxxN MirrorBit™ Flash Family

S29WS256N, S29WSI28N, S29WS064N
256/128/64 Megabit (16/8/4 M x 16-Bit) CMOS 1.8 Volt-only Simultaneous Read/Write, Burst Mode Flash Memory

Data Sheet

PRELIMINARY

General Description

The Spansion S29WS256/128/064N are Mirrorbit™ Flash products fabricated on 110 nm process technology. These burst mode Flash devices are capable of performing simultaneous read and write operations with zero latency on two separate banks using separate data and address pins. They operate up to 80 MHz and use a single V_{CC} of 1.7–1.95 volts that makes them ideal for today's demanding wireless applications requiring higher density, better performance and lowered power consumption.

Distinctive Characteristics

- **Single 1.8 V read/program/erase (1.70–1.95 V)**
- **110 nm MirrorBit™ Technology**
- **Simultaneous Read/Write operation with zero latency**
- **32-word Write Buffer**
- **Sixteen-bank architecture consisting of 16/8/4 Mbit for WS256N/128N/064N, respectively**
- **Four 16 Kword sectors at both top and bottom of memory array**
- **254/126/62 64 Kword sectors (WS256N/128N/064N)**
- **Programmable burst read modes**
  - Linear for 32, 16 or 8 words linear read with or without wrap-around
  - Continuous sequential read mode
- **SecSi™ (Secured Silicon) Sector region consisting of 128 words each for factory and customer**
- **20-year data retention (typical)**
- **Cycling Endurance: 100,000 cycles per sector (typical)**
- **RDY output indicates data available to system**
- **Command set compatible with JEDEC standards**
- **Hardware (WP#) protection of top and bottom sectors**
- **Dual boot sector configuration (top and bottom)**
- **Offered Packages**
  - WS064N: 80-ball FBGA (7 mm x 9 mm)
  - WS256N/128N: 84-ball FBGA (8 mm x 11.6 mm)
- **Low V_{CC} write inhibit**
- **Persistent and Password methods of Advanced Sector Protection**
- **Write operation status bits indicate program and erase operation completion**
- **Suspend and Resume commands for Program and Erase operations**
- **Unlock Bypass program command to reduce programming time**
- **Synchronous or Asynchronous program operation, independent of burst control register settings**
- **ACC input pin to reduce factory programming time**
- **Support for Common Flash Interface (CFI)**
- **Industrial Temperature range (contact factory)**

Performance Characteristics

Read Access Times			
Speed Option (MHz)	**80**	**66**	**54**
Max. Synch. Latency, ns (t_{IACC})	69	69	69
Max. Synch. Burst Access, ns (t_{BACC})	9	11.2	13.5
Max. Asynch. Access Time, ns (t_{ACC})	70	70	70
Max CE# Access Time, ns (t_{CE})	70	70	70
Max OE# Access Time, ns (t_{OE})	11.2	11.2	13.5

Current Consumption (typical values)	
Continuous Burst Read @ 66 MHz	35 mA
Simultaneous Operation (asynchronous)	50 mA
Program (asynchronous)	19 mA
Erase (asynchronous)	19 mA
Standby Mode (asynchronous)	20 µA

Typical Program & Erase Times	
Single Word Programming	40 µs
Effective Write Buffer Programming (V_{CC}) Per Word	9.4 µs
Effective Write Buffer Programming (V_{ACC}) Per Word	6 µs
Sector Erase (16 Kword Sector)	150 ms
Sector Erase (64 Kword Sector)	600 ms

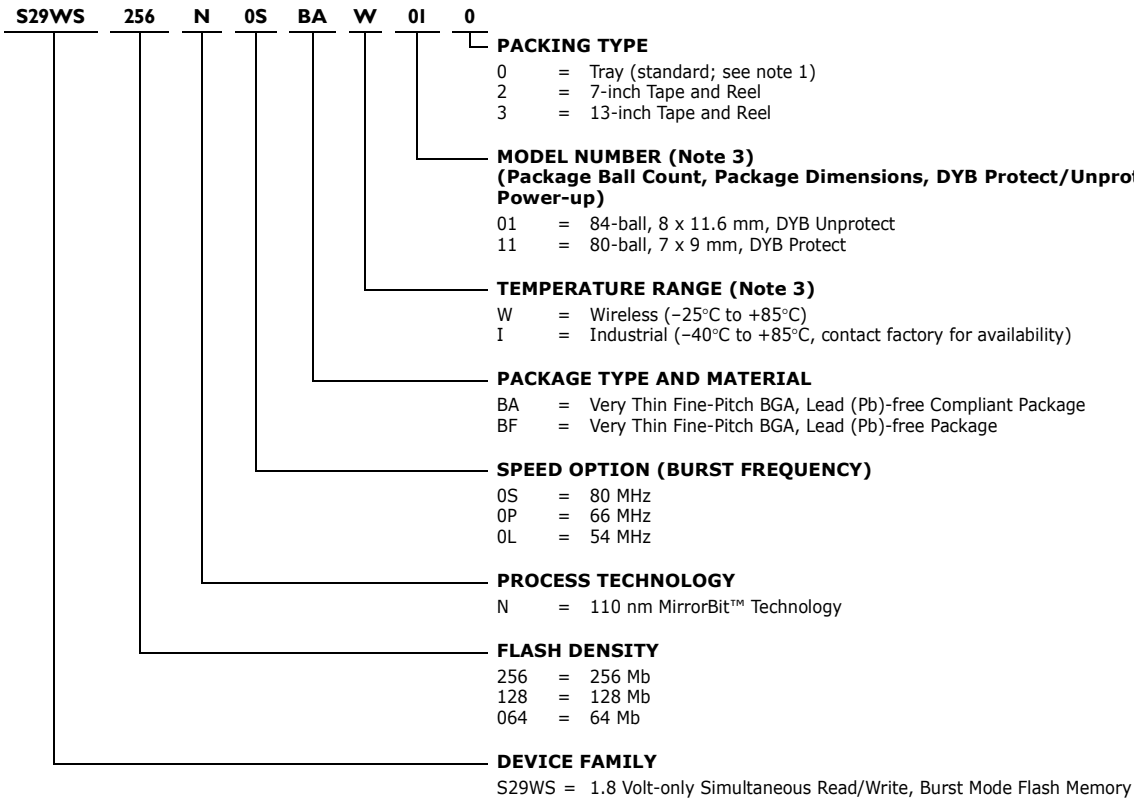
Table of Contents

List of Figures

List of Tables

I Ordering Information

The ordering part number is formed by a valid combination of the following:



S29WSxxxN Valid Combinations (Notes 1, 2, 3)					V_{IO} Range	DYB Power Up State	Package Type (Note 2)
Base Ordering Part Number	**Speed Option**	**Package Type, Material, & Temperature Range**	**Model Number**	**Packing Type**			
S29WS256N	0S, 0P, 0L	BAW (Lead (Pb)-free Compliant), BFW (Lead (Pb)-free)	01	0, 2, 3 (Note 1)	1.70–1.95 V	Unprotect	8 mm x 11.6 mm 84-ball MCP-Compatible
			11			Protect	
S29WS128N			01			Unprotect	
			11			Protect	
S29WS064N			01			Unprotect	7 mm x 9 mm 80-ball MCP-Compatible
			11			Protect	

Notes:

1. *Type 0 is standard. Specify other options as required.*
2. *BGA package marking omits leading "S29" and packing type designator from ordering part number.*
3. *For 1.5 V_{IO} option, other boot options, or industrial temperature range, contact your local sales office.*

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.

2 Input/Output Descriptions & Logic Symbol

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

Table 2.1. Input/Output Descriptions

Symbol	Type	Description
A23–A0	Input	Address lines for WS256N (A22-A0 for WS128 and A21-A0 for WS064N).
DQ15–DQ0	I/O	Data input/output.
CE#	Input	Chip Enable. Asynchronous relative to CLK.
OE#	Input	Output Enable. Asynchronous relative to CLK.
WE#	Input	Write Enable.
V_{CC}	Supply	Device Power Supply.
V_{IO}	Input	VersatileIO Input. Should be tied to V_{CC}.
V_{SS}	I/O	Ground.
NC	No Connect	Not connected internally.
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. Should be at V_{IL} or V_{IH} while in asynchronous mode.
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.
WP#	Input	Write Protect. At V_{IL}, disables program and erase functions in the four outermost sectors. Should be at V_{IH} for all other conditions.
ACC	Input	Acceleration Input. At V_{HH}, accelerates programming; automatically places device in unlock bypass mode. At V_{IL}, disables all program and erase functions. Should be at V_{IH} for all other conditions.
RFU	Reserved	Reserved for future use (see MCP look-ahead pinout for use with MCP).

3 Block Diagram

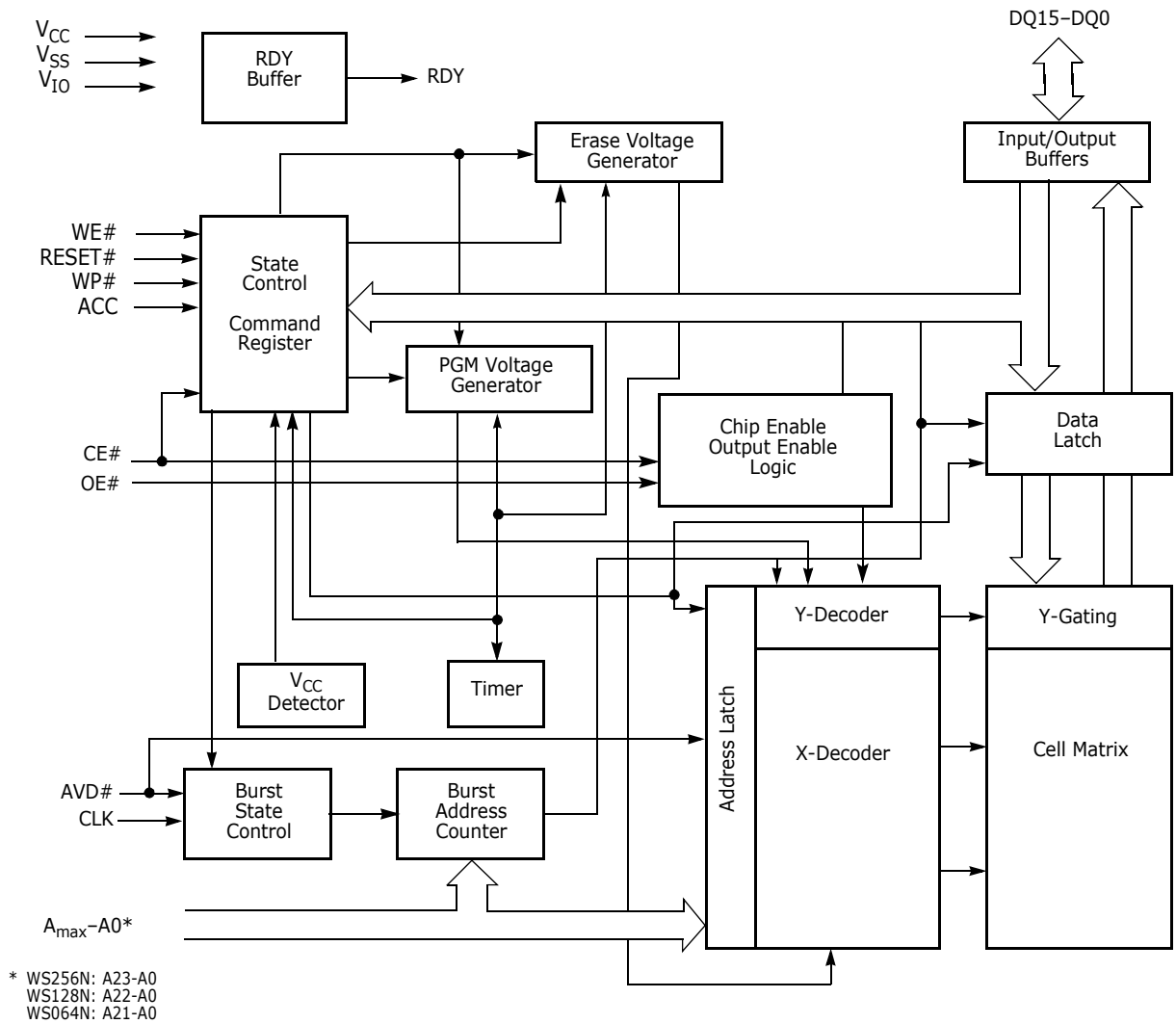


Figure 3.1. S29WSxxxN Block Diagram

4 Physical Dimensions/Connection Diagrams

This section shows the I/O designations and package specifications for the S29WSxxxN.

4.1 Related Documents

The following documents contain information relating to the S29WSxxxN devices. Click on the title or go to www.amd.com/flash (click on Technical Documentation) or www.fujitsu.com to download the PDF file, or request a copy from your sales office.

- **Migration to the S29WS256N Family Application Note**
- **Considerations for X-ray Inspection of Surface-Mounted Flash Integrated Circuits**

4.2 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.

84-Ball Fine-Pitch Ball Grid Array, 256 & 128 Mb
(Top View, Balls Facing Down,
MCP Compatible)

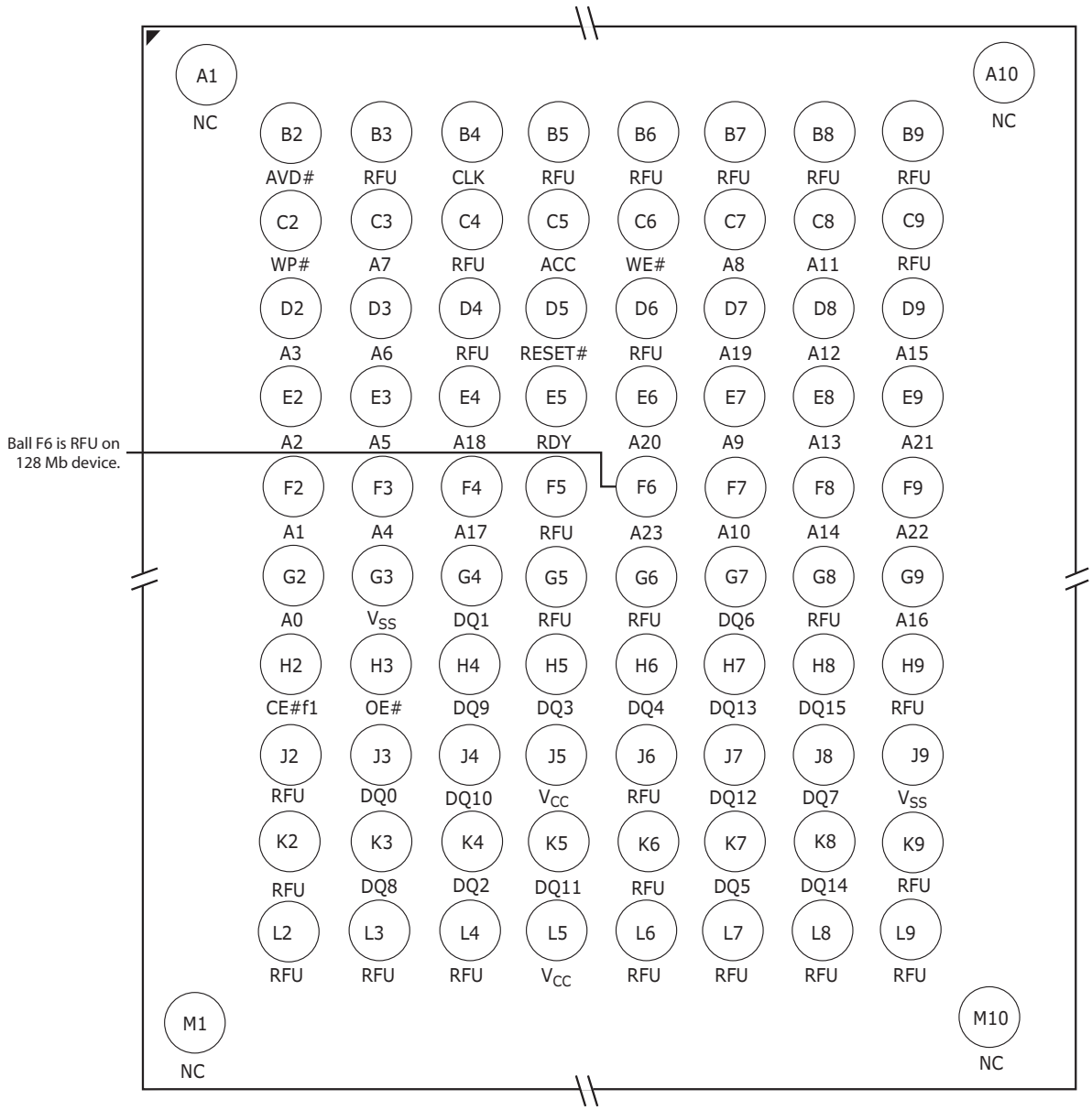
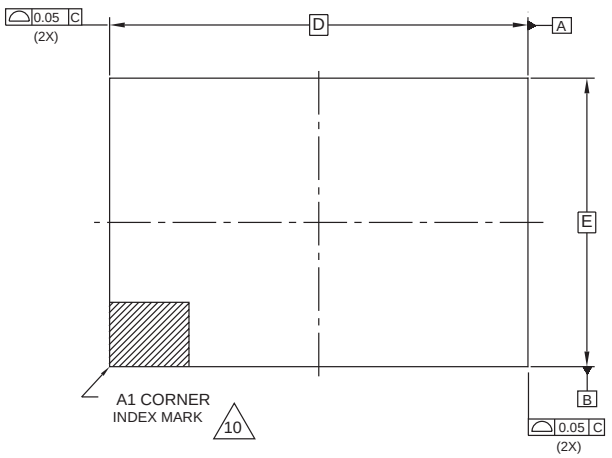
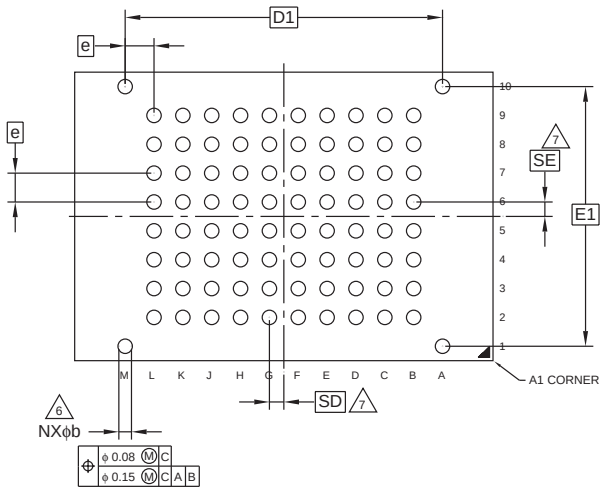


Figure 4.1. 84-ball Fine-Pitch Ball Grid Array (S29WS256N, S29WSI28N)

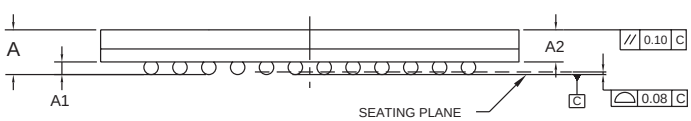
VBH084—84-ball Fine-Pitch Ball Grid Array, 8 x 11.6 mm



TOP VIEW



BOTTOM VIEW



SIDE VIEW

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: *BSC is an ANSI standard for Basic Space Centering*

Figure 4.2. VBH084—84-ball Fine-Pitch Ball Grid Array (FBGA) 8 x 11.6 mm MCP Compatible Package

80-ball Fine-Pitch Ball Grid Array, 64 Mb
(Top View, Balls Facing Down,
MCP Compatible)

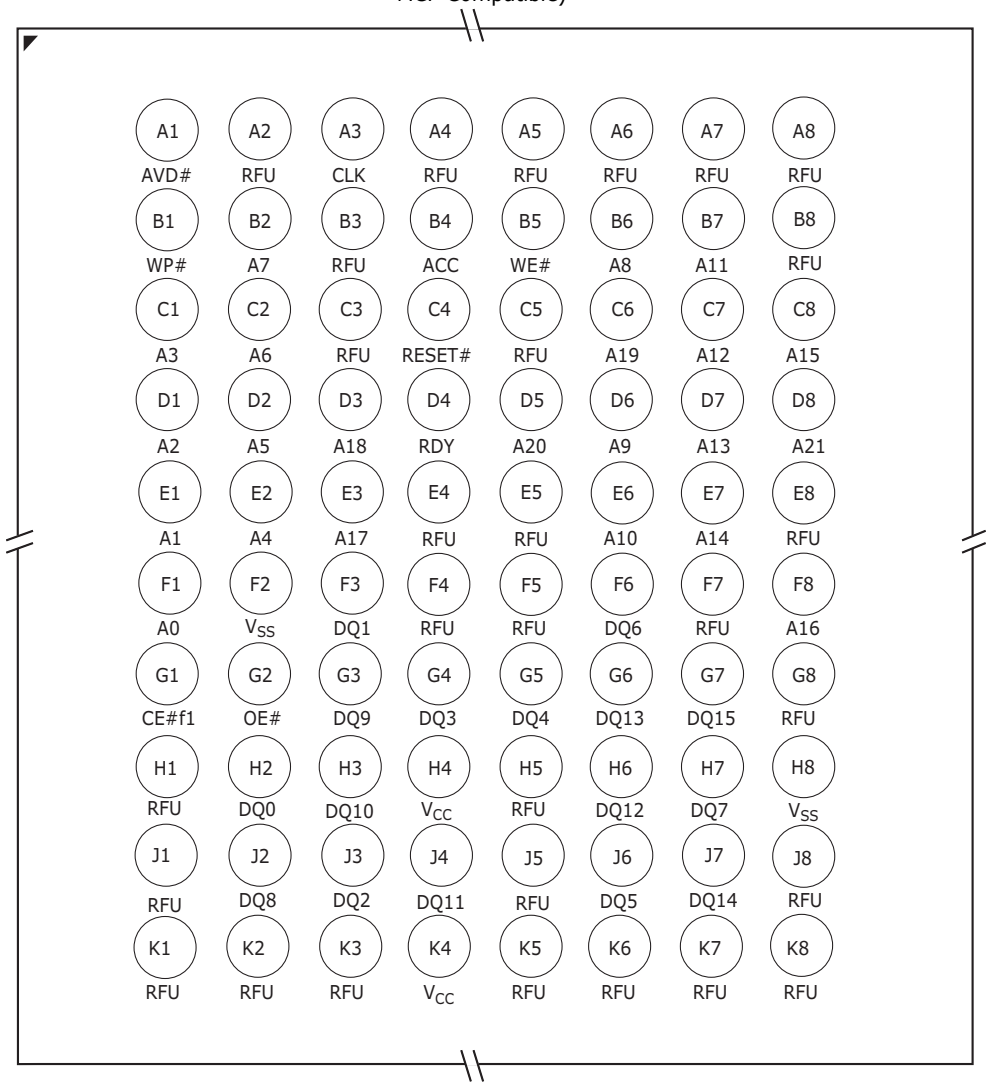
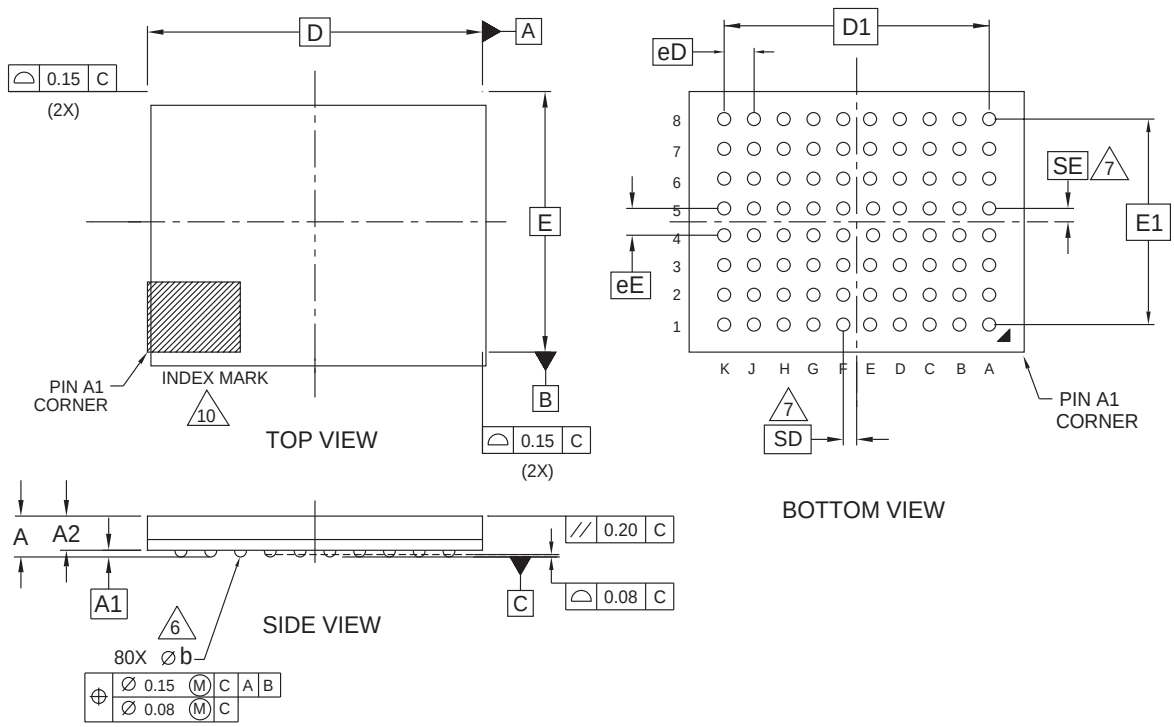


Figure 4.3. 80-ball Fine-Pitch Ball Grid Array (S29WS064N)

TLC080—80-ball Fine-Pitch Ball Grid Array, 7 x 9 mm



PACKAGE	TLC 080			
JEDEC	N/A			
D x E	9.00 mm x 7.00 mm PACKAGE			
SYMBOL	MIN	NOM	MAX	NOTE
A	---	---	1.20	PROFILE
A1	0.17	---	---	BALL HEIGHT
A2	0.81	---	0.97	BODY THICKNESS
D		9.00 BSC.		BODY SIZE
E		7.00 BSC.		BODY SIZE
D1		7.20 BSC.		MATRIX FOOTPRINT
E1		5.60 BSC.		MATRIX FOOTPRINT
MD		10		MATRIX SIZE D DIRECTION
ME		8		MATRIX SIZE E DIRECTION
n		80		BALL COUNT
ϕb	0.35	0.40	0.45	BALL DIAMETER
eE		0.80 BSC.		BALL PITCH
eD		0.80 BSC		BALL PITCH
SD / SE		0.40 BSC.		SOLDER BALL PLACEMENT
				DEPOPULATED SOLDER BALLS

NOTES:

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

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

   n IS THE NUMBER OF POPULTED SOLDER BALL POSITIONS FOR MATRIX SIZE MD X ME.

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

   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW SD OR SE = 0.000.

   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = e/2
8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
9. N/A
10. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

3430 \ 16-038.22 \ 10.15.04

Note: *BSC is an ANSI standard for Basic Space Centering*

Figure 4.4. TLC080—80-ball Fine-Pitch Ball Grid Array (FBGA) 7 x 9 mm MCP Compatible Package

4.3 MCP Look-ahead Connection Diagram

Figure 4.5 shows a migration path from the S29WSxxxN to higher densities and the option to include additional die within a single package. Spansion LLC provides this standard look-ahead connection diagram that supports

- NOR Flash and SRAM densities up to 4 Gigabits
- NOR Flash and pSRAM densities up to 4 Gigabits
- NOR Flash and pSRAM and data storage densities up to 4 Gigabits

The following multi-chip package (MCP) data sheet(s) are based on the S29WSxxxN. Refer to these documents for input/output descriptions for each product:

Publication Number S71WS256_512NC0.

The physical package outline may vary between connection diagrams and densities. The connection diagram for any MCP, however, will be a subset of the pinout in Figure 4.5.

In some cases, outrigger balls may exist in locations outside the grid shown. These outrigger balls are reserved; do not connect them to any other signal.

For further information about the MCP look-ahead pinout, refer to the *Design-In Scalable Wireless Solutions with Spansion Products* application note, available on the web or through an AMD or Fujitsu sales office.

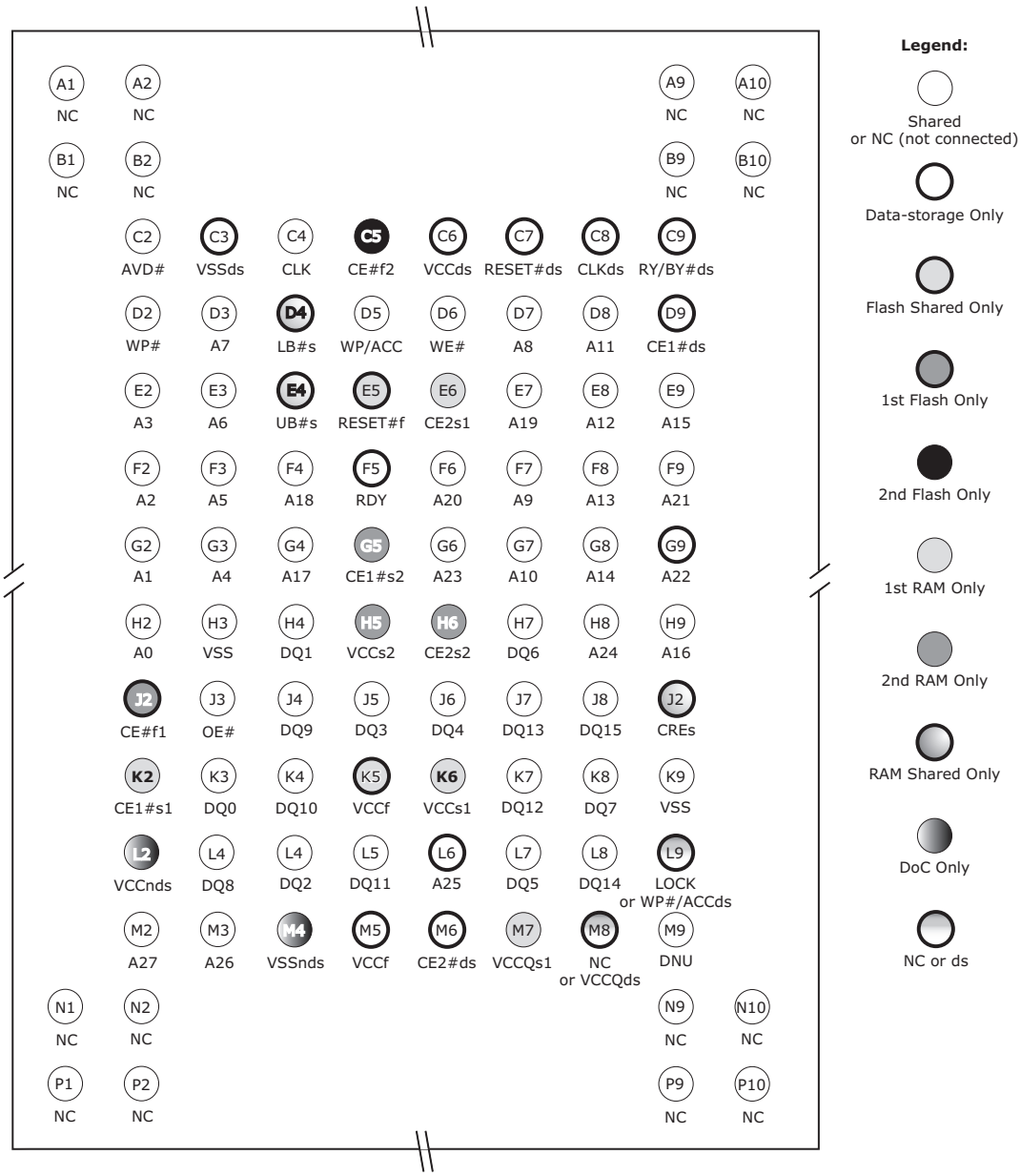


Figure 4.5. MCP Look-ahead Diagram

5 Additional Resources

Visit www.amd.com and www.fujitsu.com to obtain the following related documents:

Application Notes

- Using the Operation Status Bits in AMD Devices
- Understanding Burst Mode Flash Memory Devices
- Simultaneous Read/Write vs. Erase Suspend/Resume
- MirrorBit™ Flash Memory Write Buffer Programming and Page Buffer Read
- Design-In Scalable Wireless Solutions with Spansion Products
- Common Flash Interface Version 1.4 Vendor Specific Extensions

Specification Bulletins

Contact your local sales office for details.

Drivers and Software Support

- Spansion low-level drivers
- Enhanced Flash drivers
- Flash file system

CAD Modeling Support

- VHDL and Verilog
- IBIS
- ORCAD

Technical Support

Contact your local sales office or contact Spansion LLC directly for additional technical support:

Email
US and Canada: HW.support@amd.com
Asia Pacific: asia.support@amd.com
Europe, Middle East, and Africa
Japan: http://edevice.fujitsu.com/jp/support/tech/#b7

Frequently Asked Questions (FAQ)
http://ask.amd.com/
http://edevice.fujitsu.com/jp/support/tech/#b7

Phone
US: (408) 749-5703
Japan (03) 5322-3324

Spansion LLC Locations

915 DeGuigne Drive, P.O. Box 3453
Sunnyvale, CA 94088-3453, USA
Telephone: 408-962-2500 or
1-866-SPANSION

Spansion Japan Limited
4-33-4 Nishi Shinjuku, Shinjuku-ku
Tokyo, 160-0023
Telephone: +81-3-5302-2200
Facsimile: +81-3-5302-2674

http://www.spansion.com

6 Product Overview

The S29WSxxxN family consists of 256, 128 and 64Mbit, 1.8 volts-only, simultaneous read/write burst mode Flash device optimized for today's wireless designs that demand a large storage array, rich functionality, and low power consumption. These devices are organized in 16, 8 or 4 Mwords of 16 bits each and are capable of continuous, synchronous (burst) read or linear read (8-, 16-, or 32-word aligned group) with or without wrap around. These products also offer single word programming or a 32-word buffer for programming with program/erase and suspend functionality. Additional features include:

- Advanced Sector Protection methods for protecting sectors as required
- 256 words of secured silicon (SecSi™) area for storing customer and factory secured information. The SecSi Sector is One Time Programmable and Protectable (OTTP).

6.1 Memory Map

The S29WS256/128/064N Mbit devices consist of 16 banks organized as shown in Tables 6.1–6.3.

Table 6.1. S29WS256N Sector & Memory Address Map

Bank Size	Sector Count	Sector Size (KB)	Bank	Sector/ Sector Range	Address Range	Notes
2 MB	4	32	0	SA000	000000h–003FFFh	Contains four smaller sectors at bottom of addressable memory.
				SA001	004000h–007FFFh	
				SA002	008000h–00BFFFh	
				SA003	00C000h–00FFFFh	
	15	128		SA004 to SA018	010000h–01FFFFh to 0F0000h–0FFFFFh	All 128 KB sectors. Pattern for sector address range is xx0000h–xxFFFFh. (see note)
2 MB	16	128	1	SA019 to SA034	100000h–10FFFFh to 1F0000h–1FFFFFh	
2 MB	16	128	2	SA035 to SA050	200000h–20FFFFh to 2F0000h–2FFFFFh	
2 MB	16	128	3	SA051 to SA066	300000h–30FFFFh to 3F0000h–3FFFFFh	
2 MB	16	128	4	SA067 to SA082	400000h–40FFFFh to 4F0000h–4FFFFFh	
2 MB	16	128	5	SA083 to SA098	500000h–50FFFFh to 5F0000h–5FFFFFh	
2 MB	16	128	6	SA099 to SA114	600000h–60FFFFh to 6F0000h–6FFFFFh	
2 MB	16	128	7	SA115 to SA130	700000h–70FFFFh to 7F0000h–7FFFFFh	
2 MB	16	128	8	SA131 to SA146	800000h–80FFFFh to 8F0000h–8FFFFFh	
2 MB	16	128	9	SA147 to SA162	900000h–90FFFFh to 9F0000h–9FFFFFh	
2 MB	16	128	10	SA163 to SA178	A00000h–A0FFFFh to AF0000h–AFFFFFh	
2 MB	16	128	11	SA179 to SA194	B00000h–B0FFFFh to BF0000h–BFFFFFh	
2 MB	16	128	12	SA195 to SA210	C00000h–C0FFFFh to CF0000h–CFFFFFh	
2 MB	16	128	13	SA211 to SA226	D00000h–D0FFFFh to DF0000h–DFFFFFh	
2 MB	16	128	14	SA227 to SA242	E00000h–E0FFFFh to EF0000h–EFFFFFh	
2 MB	15	128	15	SA243 to SA257	F00000h–F0FFFFh to FE0000h–FEFFFFh	
	4	32		SA258	FF0000h–FF3FFFh	Contains four smaller sectors at top of addressable memory.
				SA259	FF4000h–FF7FFFh	
				SA260	FF8000h–FFBFFFh	
				SA261	FFC000h–FFFFFFh	

Note: *This table has been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA005–SA017) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the pattern xx00000h–xxFFFFh.*

Table 6.2. S29WSI28N Sector & Memory Address Map

Bank Size	Sector Count	Sector Size (KB)	Bank	Sector/ Sector Range	Address Range	Notes
1 MB	4	32	0	SA000	000000h–003FFFh	Contains four smaller sectors at bottom of addressable memory.
		32		SA001	004000h–007FFFh	
		32		SA002	008000h–00BFFFh	
		32		SA003	00C000h–00FFFFh	
	7	128		SA004 to SA010	010000h–01FFFFh to 070000h–07FFFFh	All 128 KB sectors. Pattern for sector address range is xx0000h–xxFFFFh. (see note)
1 MB	8	128	1	SA011 to SA018	080000h–08FFFFh to 0F0000h–0FFFFFh	
1 MB	8	128	2	SA019 to SA026	100000h–10FFFFh to 170000h–17FFFFh	
1 MB	8	128	3	SA027 to SA034	180000h–18FFFFh to 1F0000h–1FFFFFh	
1 MB	8	128	4	SA035 to SA042	200000h–20FFFFh to 270000h–27FFFFh	
1 MB	8	128	5	SA043 to SA050	280000h–28FFFFh to 2F0000h–2FFFFFh	
1 MB	8	128	6	SA051 to SA058	300000h–30FFFFh to 370000h–37FFFFh	
1 MB	8	128	7	SA059 to SA066	380000h–38FFFFh to 3F0000h–3FFFFFh	
1 MB	8	128	8	SA067 to SA074	400000h–40FFFFh to 470000h–47FFFFh	
1 MB	8	128	9	SA075 to SA082	480000h–48FFFFh to 4F0000h–4FFFFFh	
1 MB	8	128	10	SA083 to SA090	500000h–50FFFFh to 570000h–57FFFFh	
1 MB	8	128	11	SA091 to SA098	580000h–58FFFFh to 5F0000h–5FFFFFh	
1 MB	8	128	12	SA099 to SA106	600000h–60FFFFh to 670000h–67FFFFh	
1 MB	8	128	13	SA107 to SA114	680000h–68FFFFh to 6F0000h–6FFFFFh	
1 MB	8	128	14	SA115 to SA122	700000h–70FFFFh to 770000h–77FFFFh	
1 MB	7	128	15	SA123 to SA129	780000h–78FFFFh to 7E0000h–7EFFFFh	
	4	32		SA130	7F0000h–7F3FFFh	Contains four smaller sectors at top of addressable memory.
		32		SA131	7F4000h–7F7FFFh	
		32		SA132	7F8000h–7FBFFFh	
		32		SA133	7FC000h–7FFFFFh	

Note: *This table has been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA005–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the pattern xx00000h–xxFFFFh.*

Table 6.3. S29WS064N Sector & Memory Address Map

Bank Size	Sector Count	Sector Size (KB)	Bank	Sector/ Sector Range	Address Range	Notes
0.5 MB	4	32	0	SA000	000000h–003FFFh	Contains four smaller sectors at bottom of addressable memory.
				SA001	004000h–007FFFh	
				SA002	008000h–00BFFFh	
				SA003	00C000h–00FFFFh	
	3	128		SA004	010000h–01FFFFh	All 128 KB sectors. Pattern for sector address range is xx0000h–xxFFFFh. (see note)
				SA005	020000h–02FFFFh	
				SA006	030000h–03FFFFh	
0.5 MB	4	128	1	SA007–SA010	040000h–04FFFFh to 070000h–07FFFFh	
0.5 MB	4	128	2	SA011–SA014	080000h–08FFFFh to 0B0000h–0BFFFFh	
0.5 MB	4	128	3	SA015–SA018	0C0000h–0CFFFFh to 0F0000h–0FFFFFh	
0.5 MB	4	128	4	SA019–SA022	100000h–10FFFFh to 130000h–13FFFFh	
0.5 MB	4	128	5	SA023–SA026	140000h–14FFFFh to 170000h–17FFFFh	
0.5 MB	4	128	6	SA027–SA030	180000h–18FFFFh to 1B0000h–1BFFFFh	
0.5 MB	4	128	7	SA031–SA034	1C0000h–1CFFFFh to 1F0000h–1FFFFFh	
0.5 MB	4	128	8	SA035–SA038	200000h–20FFFFh to 230000h–23FFFFh	
0.5 MB	4	128	9	SA039–SA042	240000h–24FFFFh to 270000h–27FFFFh	
0.5 MB	4	128	10	SA043–SA046	280000h–28FFFFh to 2B0000h–2BFFFFh	
0.5 MB	4	128	11	SA047–SA050	2C0000h–2CFFFFh to 2F0000h–2FFFFFh	
0.5 MB	4	128	12	SA051–SA054	300000h–30FFFFh to 330000h–33FFFFh	
0.5 MB	4	128	13	SA055–SA058	340000h–34FFFFh to 370000h–37FFFFh	
0.5 MB	4	128	14	SA059–SA062	380000h–38FFFFh to 3B0000h–3BFFFFh	
0.5 MB	3	128	15	SA063	3C0000h–3CFFFFh	
				SA064	3D0000h–3DFFFFh	
				SA065	3E0000h–3EFFFFh	
	4	32		SA066	3F0000h–3F3FFFh	Contains four smaller sectors at top of addressable memory.
				SA067	3F4000h–3F7FFFh	
				SA068	3F8000h–3FBFFFh	
				SA069	3FC000h–3FFFFFh	

Note: *This table has been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the pattern xx00000h–xxFFFFh.*

7 Device Operations

This section describes the read, program, erase, simultaneous read/write operations, hand-shaking, 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 Tables 12.1 and 12.2). 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 7.4 describes the required state of each control pin for any particular operation.

Table 7.4. Device Operations

Operation	CE#	OE#	WE#	Addresses	DQ15–0	RESET#	CLK	AVD#
Asynchronous Read - Addresses Latched	L	L	H	Addr In	Data Out	H	X	↑
Asynchronous Read - Addresses Steady State	L	L	H	Addr In	Data Out	H	X	L
Asynchronous Write	L	H	L	Addr In	I/O	H	X	L
Synchronous Write	L	H	L	Addr In	I/O	H	↑	↑
Standby (CE#)	H	X	X	X	HIGH Z	H	X	X
Hardware Reset	X	X	X	X	HIGH Z	L	X	X
Burst Read Operations (Synchronous)								
Load Starting Burst Address	L	X	H	Addr In	X	H	↑	↑
Advance Burst to next address with appropriate Data presented on the Data Bus	L	L	H	X	Burst Data Out	H	↑	H
Terminate current Burst read cycle	H	X	H	X	HIGH Z	H	↑	X
Terminate current Burst read cycle via RESET#	X	X	H	X	HIGH Z	L	X	X
Terminate current Burst read cycle and start new Burst read cycle	L	X	H	Addr In	I/O	H	↑	↑

Legend: *L = Logic 0, H = Logic 1, X = Don't Care, I/O = Input/Output.*

7.2 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# should remain at V_{IH}. The rising edge of AVD# latches the address and data will appear on DQ15–DQ0 after address access time (t_{ACC}), which is equal to the delay from stable addresses to valid output data. The chip enable

access time (t_{CE}) is the delay from the stable CE# to valid data at the outputs. The output enable access time (t_{OE}) is the delay from the falling edge of OE# to valid data at the output.

7.3 Synchronous (Burst) Read Mode & 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 subsections 7.3.1 and 7.3.2, and 7.3.3.

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 will indicate 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 7.3.4, Configuration Register, and Table 12.1, Memory Array Commands for further details.

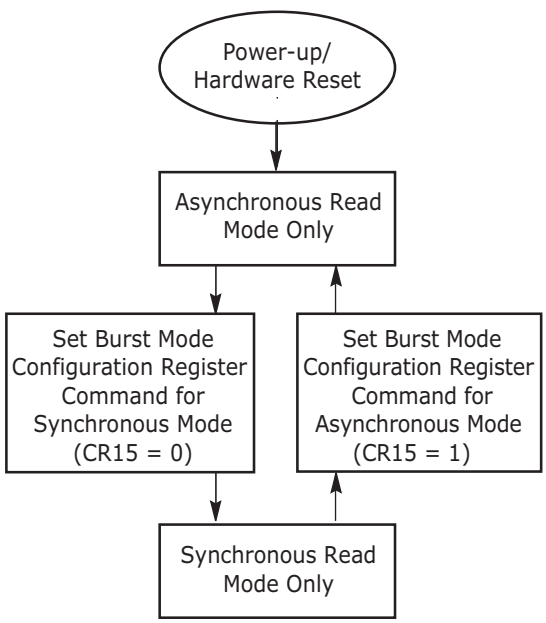


Figure 7.I. Synchronous/Asynchronous State Diagram

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 four (where A[1:0] is 00). A divisible-by-four address incurs the least number of additional wait states that occur after the initial word. The number of additional wait states required increases for burst operations in which the starting address is one, two, or three locations above the divisible-by-four address (i.e., where A[1:0] is 01, 10, or 11).
- boundary crossing: a physical aspect of the device that exists every 128 words, starting at address 00007Fh. Higher operational speeds require one additional wait state. Refer to Tables 7.10–7.13 for details. Figure 11.20 shows the effects of boundary crossings at higher frequencies.
- 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. Tables 7.7–7.13 show the effects of frequency on burst operation.

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

Table 7.5 shows the latency that occurs in the S29WS256N device when (x indicates the recommended number of wait states for various operating frequencies, as shown in Table 7.15, configuration register bits CR13-CR11).

Tables 7.7–7.9 show the effects of various combinations of the starting address, operating frequency, and wait state setting (configuration register bits CR13–CR11) for the S29WS128N and S29WS064N devices. Tables 7.10–7.13 includes the wait state that occurs when crossing the internal boundary.

Table 7.5. Address Latency for x Wait States (≤ 80 MHz, WS256N only)

Word	Wait States	Cycle								
0	x ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	x ws	D1	D2	D3	1 ws	D4	D5	D6	D7	D8
2	x ws	D2	D3	1 ws	1 ws	D4	D5	D6	D7	D8
3	x ws	D3	1 ws	1 ws	1 ws	D4	D5	D6	D7	D8

Table 7.6. Address Latency for 6 Wait States (≤ 80 MHz)

Word	Wait States	Cycle								
0	6 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	6 ws	D1	D2	D3	1 ws	D4	D5	D6	D7	D8
2	6 ws	D2	D3	1 ws	1 ws	D4	D5	D6	D7	D8
3	6 ws	D3	1 ws	1 ws	1 ws	D4	D5	D6	D7	D8

Table 7.7. Address Latency for 5 Wait States (≤ 68 MHz)

Word	Wait States	Cycle								
0	5 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	5 ws	D1	D2	D3	D4	D5	D6	D7	D8	D9
2	5 ws	D2	D3	1 ws	D4	D5	D6	D7	D8	D9
3	5 ws	D3	1 ws	1 ws	D4	D5	D6	D7	D8	D9

Table 7.8. Address Latency for 4 Wait States (≤ 54 MHz)

Word	Wait States	Cycle								
0	4 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	4 ws	D1	D2	D3	D4	D5	D6	D7	D8	D9
2	4 ws	D2	D3	D4	D5	D6	D7	D8	D9	D10
3	4 ws	D3	1 ws	D4	D5	D6	D7	D8	D9	D10

Table 7.9. Address Latency for 3 Wait States (≤ 40 MHz)

Word	Wait States	Cycle								
0	3 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	3 ws	D1	D2	D3	D4	D5	D6	D7	D8	D9
2	3 ws	D2	D3	D4	D5	D6	D7	D8	D9	D10
3	3 ws	D3	D4	D5	D6	D7	D8	D9	D10	D11

Table 7.10. Address/Boundary Crossing Latency for 6 Wait States (≤ 80 MHz)

Word	Wait States	Cycle								
0	6 ws	D0	D1	D2	D3	1 ws	D4	D5	D6	D7
1	6 ws	D1	D2	D3	1 ws	1 ws	D4	D5	D6	D7
2	6 ws	D2	D3	1 ws	1 ws	1 ws	D4	D5	D6	D7
3	6 ws	D3	1 ws	1 ws	1 ws	1 ws	D4	D5	D6	D7

Table 7.11. Address/Boundary Crossing Latency for 5 Wait States (≤ 68 MHz)

Word	Wait States	Cycle								
0	5 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	5 ws	D1	D2	D3	1 ws	D4	D5	D6	D7	D8
2	5 ws	D2	D3	1 ws	1 ws	D4	D5	D6	D7	D8
3	5 ws	D3	1 ws	1 ws	1 ws	D4	D5	D6	D7	D8

Table 7.12. Address/Boundary Crossing Latency for 4 Wait States (≤ 54 MHz)

Word	Wait States	Cycle								
0	4 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	4 ws	D1	D2	D3	D4	D5	D6	D7	D8	D9
2	4 ws	D2	D3	1 ws	D4	D5	D6	D7	D8	D9
3	4 ws	D3	1 ws	1 ws	D4	D5	D6	D7	D8	D9

Table 7.13. Address/Boundary Crossing Latency for 3 Wait States (≤ 40 MHz)

Word	Wait States	Cycle								
0	3 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1	3 ws	D1	D2	D3	D4	D5	D6	D7	D8	D9
2	3 ws	D2	D3	D4	D5	D6	D7	D8	D9	D10
3	3 ws	D3	1 ws	D4	D5	D6	D7	D8	D9	D10

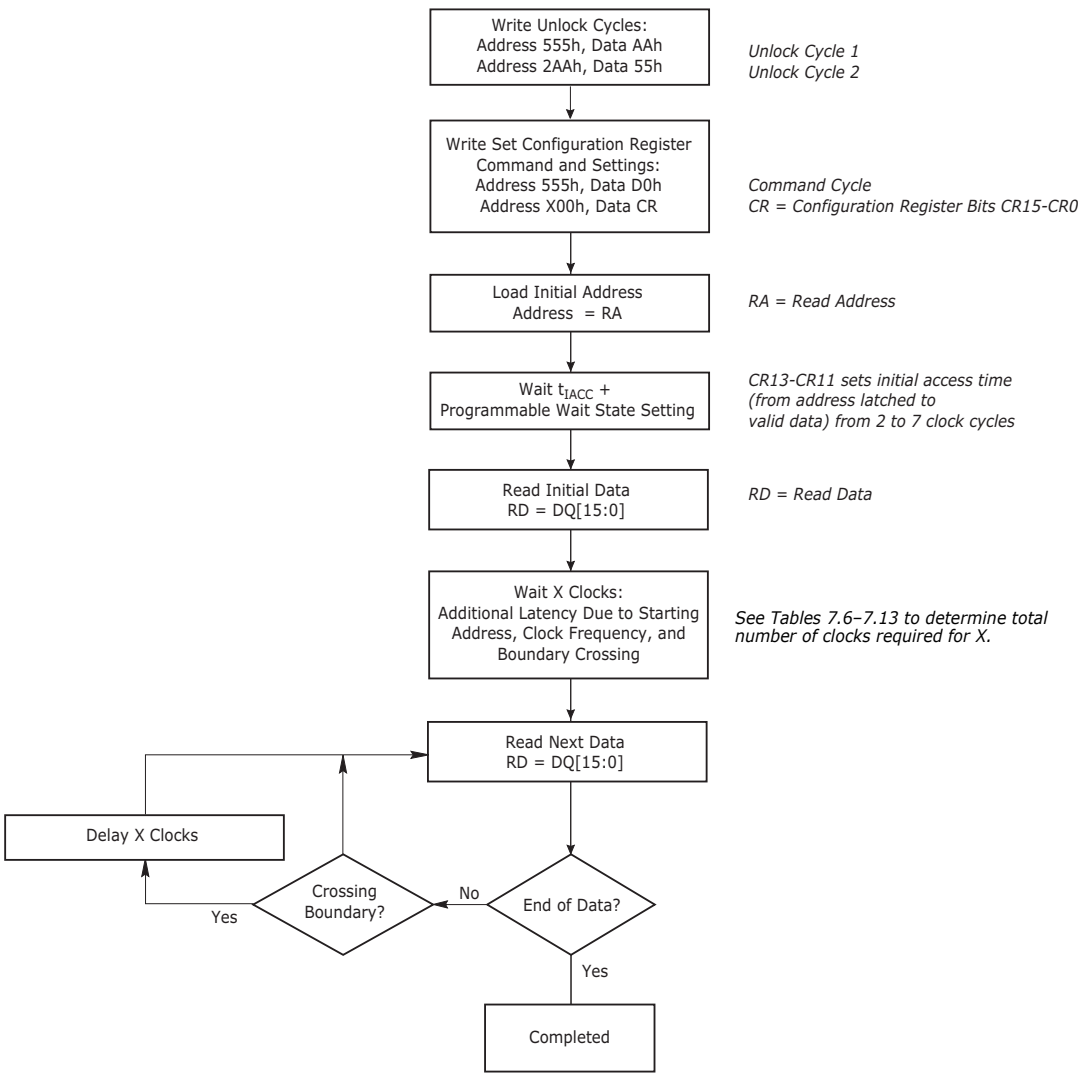


Figure 7.2. Synchronous Read

7.3.1 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 will continue until the system drives CE# high, RESET# low, or AVD# low in conjunction with a new address.

If the address being read crosses a 128-word line boundary and the subsequent word line is not programming or erasing, additional latency cycles are required as shown in Tables 7.10–7.13.

If the address crosses a bank boundary while the subsequent bank is programming or erasing, the device will provide read status information and the clock will be 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.

7.3.2 8-, 16-, 32-Word Linear Burst Read with Wrap Around

In a linear burst read operation, a fixed number of words (8, 16, or 32 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 7.14).

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 and 32-word Linear Wrap modes begin their burst sequence on the starting address written to the device, then wrap back to the first address in the selected address group.

Note that in this mode the address pointer does not cross the boundary that occurs every 128 words; thus, no wait states are inserted (except during the initial access).

Table 7.14. Burst Address Groups

Mode	Group Size	Group Address Ranges
8-word	8 words	0-7h, 8-Fh, 10-17h,...
16-word	16 words	0-Fh, 10-1Fh, 20-2Fh,...
32-word	32 words	00-1Fh, 20-3Fh, 40-5Fh,...

7.3.3 8-, 16-, 32-Word Linear Burst without Wrap Around

If wrap around is not enabled for linear burst read operations, the 8-word, 16-word, or 32-word burst will execute up to the maximum memory address of the selected number of words. The burst will stop after 8, 16, or 32 addresses and will not wrap around to the first address of the selected group.

For example, if the starting address in the 8- word mode is 3Ch, the address range to be read would be 39-40h, and the burst sequence would be 3C-3D-3E-3F-40-41-42-43h if wrap around is not enabled. The next address to be read will require a new address and AVD# pulse. Note that in this burst read mode, the address pointer may cross the boundary that occurs every 128 words.

7.3.4 Configuration Register

The configuration register sets various operational features, most of which are 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 deasserting CE#. The Configuration Register can also be read using a command sequence (see Table 12.1). The following list describes the register settings.

Table 7.15. 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	Boundary Crossing	0 = No extra boundary crossing latency 1 = With extra boundary crossing latency (default) Must be set to "1" at higher operating frequencies. See Tables 7.10–7.13.
CR13 CR12 CR11	Programmable Wait State	000 = Data valid on 2nd active CLK edge after addresses latched 001 = Data valid on 3rd active CLK edge after addresses latched 010 = Data valid on 4th active CLK edge after addresses latched (recommended for 54 MHz) 011 = Data valid on 5th active CLK edge after addresses latched (recommended for 66 MHz) 100 = Data valid on 6th active CLK edge after addresses latched (recommended for 80 MHz) 101 = Data valid on 7th active CLK edge after addresses latched (default) 110 = Reserved 111 = Reserved Inserts wait states before initial data is available. Setting greater number of wait states before initial data reduces latency after initial data. See Tables 7.6–7.13.
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 will be active with data regardless of CR8 setting.
CR7	Reserved	1 = default
CR6	Reserved	1 = default
CR5	Reserved	0 = default
CR4	Reserved	0 = default
CR3	Burst Wrap Around	0 = No Wrap Around Burst 1 = Wrap Around Burst (default)
CR2 CR1 CR0	Burst Length	000 = Continuous (default) 010 = 8-Word Linear Burst 011 = 16-Word Linear Burst 100 = 32-Word Linear Burst (All other bit settings are reserved)

Note: *Configuration Register will be in the default state upon power-up or hardware reset.*

Reading the Configuration Table. The configuration register can be read with a four-cycle command sequence. See Table 12.1 for sequence details. Once the data has been read from the configuration register, a software reset command is required to set the device into the correct state.

7.4 Autoselect

The Autoselect mode provides manufacturer and device identification, and sector protection verification, through identifier codes output from the internal register (separate from the memory array) on DQ15-DQ0. This mode is primarily intended for programming equipment to automatically match a device to be programmed with its corresponding programming algorithm. The Autoselect codes can also be accessed in-system. When verifying sector protection, the sector address must appear on the appropriate highest order address bits (see Tables 7.17 to 7.16). The remaining address bits are don't care. When all necessary bits have been set as required, the programming equipment may then read the corresponding identifier code on DQ15-DQ0. The Autoselect codes can also be accessed in-system through the command register. Note that if a Bank Address (BA) on the four uppermost address bits is asserted during the third write cycle of the Autoselect command, the host system can read Autoselect data from that bank and then immediately read array data from the other bank, without exiting the Autoselect mode.

- To access the Autoselect codes, the host system must issue the Autoselect command.
- The Autoselect command sequence may be written to an address within a bank that is either in the read or erase-suspend-read mode.
- The Autoselect command may not be written while the device is actively programming or erasing in the other bank. Autoselect does not support simultaneous operations or burst mode.
- The 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 for command sequence details.

Table 7.16. Autoselect Addresses

Description	Address	Read Data
Manufacturer ID	(BA) + 00h	0001h
Device ID, Word 1	(BA) + 01h	227Eh
Device ID, Word 2	(BA) + 0Eh	2230 (WS256N) 2231 (WS128N) 2232 (WS064N)
Device ID, Word 3	(BA) + 0Fh	2200
Indicator Bits (See Note)	(BA) + 03h	DQ15 - DQ8 = Reserved DQ7 (Factory Lock Bit): 1 = Locked, 0 = Not Locked DQ6 (Customer Lock Bit): 1 = Locked, 0 = Not Locked DQ5 (Handshake Bit): 1 = Reserved, 0 = Standard Handshake DQ4, DQ3 (WP# Protection Boot Code): 00 = WP# Protects both Top Boot and Bottom Boot Sectors. 01, 10, 11 = Reserved DQ2 = Reserved DQ1 (DYB Power up State [Lock Register DQ4]): 1 = Unlocked (user option), 0 = Locked (default) DQ0 (PPB Eraseability [Lock Register DQ3]): 1 = Erase allowed, 0 = Erase disabled
Sector Block Lock/ Unlock	(SA) + 02h	0001h = Locked, 0000h = Unlocked

Note: *For WS128N and WS064, DQ1 and DQ0 will be reserved.*

Software Functions and Sample Code

Table 7.17. Autoselect Entry

(LLD Function = lld_AutoselectEntryCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	BAxAAAh	BAx555h	0x00AAh
Unlock Cycle 2	Write	BAx555h	BAx2AAh	0x0055h
Autoselect Command	Write	BAxAAAh	BAx555h	0x0090h

Table 7.18. Autoselect Exit

(LLD Function = lld_AutoselectExitCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	base + XXXh	base + XXXh	0x00F0h

Notes:

1. *Any offset within the device will work.*
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.amd.com and www.fujitsu.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) */
```

7.5 Program/Erase Operations

These devices are capable of several modes of programming and or erase operations which are described in details during the following sections. However, prior to any programming and or erase operation, devices must be setup appropriately as outlined in Table 6.4.

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#.

Note the following:

- When the Embedded Program algorithm is complete, the device then returns to the read mode.
- The system can determine the status of the program operation by using DQ7 or DQ6. Refer to the Write Operation Status section for information on these status bits.
- A "0" cannot be programmed back to a "1." Attempting to do so will cause the device to set DQ5 = 1 (halting any further operation and requiring a reset command). A succeeding read will show 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.
- SecSi 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.
- Programming to the same word address multiple times without intervening erases is limited. For such application requirements, please contact your local Spansion representative.

7.5.1. 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-, 16- or 32-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 for the required bus cycles and Figure 7.19 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 the Write Operation Status section for information on these status bits.

- During programming, any command (except the Suspend Program command) is ignored.
- The SecSi 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.
- Programming to the same address multiple times continuously (for example, "walking" a bit within a word) for an extended period is not recommended. For more information, contact your local sales office.

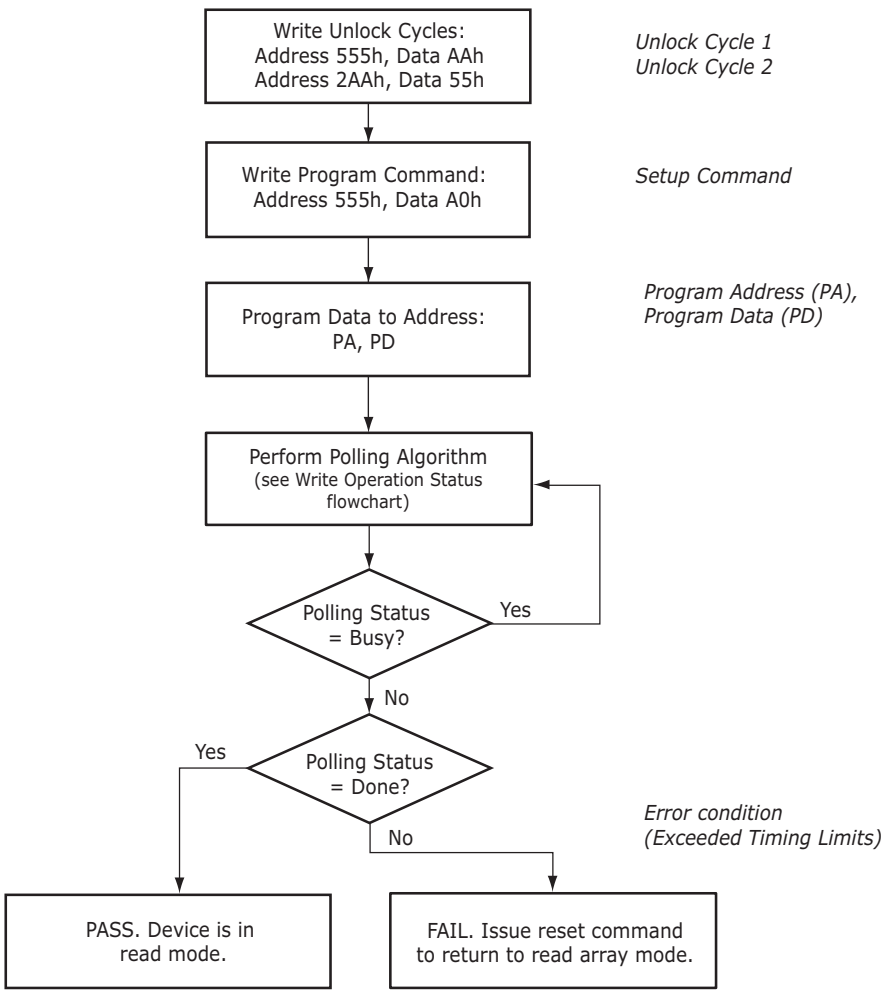


Figure 7.19. Single Word Program

Software Functions and Sample Code

Table 7.20. Single Word Program

(LLD Function = lld_ProgramCmd)

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
Program Setup	Write	Base + AAAh	Base + 555h	00A0h
Program	Write	Word Address	Word Address	Data Word

Note: *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.amd.com and www.fujitsu.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 */
```

7.5.2 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 will occur. At this point, the system writes the number of "word locations minus 1" that will be loaded into the page buffer at the Sector Address in which programming will occur. This tells the device how many write buffer addresses will be loaded with data and therefore when to expect the "Program Buffer to Flash" confirm command. The number of locations to program cannot exceed the size of the write buffer or the operation will abort. (Number loaded = the number of locations to program minus 1. For example, if the system will program 6 address locations, then 05h should be written to the device.)

The system then writes the starting address/data combination. This starting address is the first address/data pair to be programmed, and selects the "write-buffer-page" address. All subsequent address/data pairs must fall within the 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 will ABORT.)

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 will be decremented for every data load operation. Also, the last data loaded at a location before the "Program Buffer to Flash" confirm command will be programmed into the

device. It is the software's responsibility to comprehend ramifications of loading a write-buffer location more than once. The counter decrements for each data load operation, NOT for each unique write-buffer-address location. Once the specified number of write buffer locations have been loaded, the system must then write the "Program Buffer to Flash" command at the Sector Address. Any other address/data write combinations will abort the Write Buffer Programming operation. The device will then "go busy." The Data Bar polling techniques should be used while monitoring the last address location loaded into the write buffer. This eliminates the need to store an address in memory because the system can load the last address location, issue the program confirm command at the last loaded address location, and then data bar poll at that same address. DQ7, DQ6, DQ5, DQ2, and DQ1 should be monitored to determine the device status during Write Buffer Programming.

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

The Write Buffer Programming Sequence 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. A "Write-to-Buffer-Abort reset" command sequence is required when using the write buffer Programming features in Unlock Bypass mode. Note that the SecSI™ 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. However, programming the same word address multiple times without intervening erases requires a modified programming method. Please contact your local Spansion™ representative for details.

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 7.2l. 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.amd.com and www.fujitsu.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        */
```

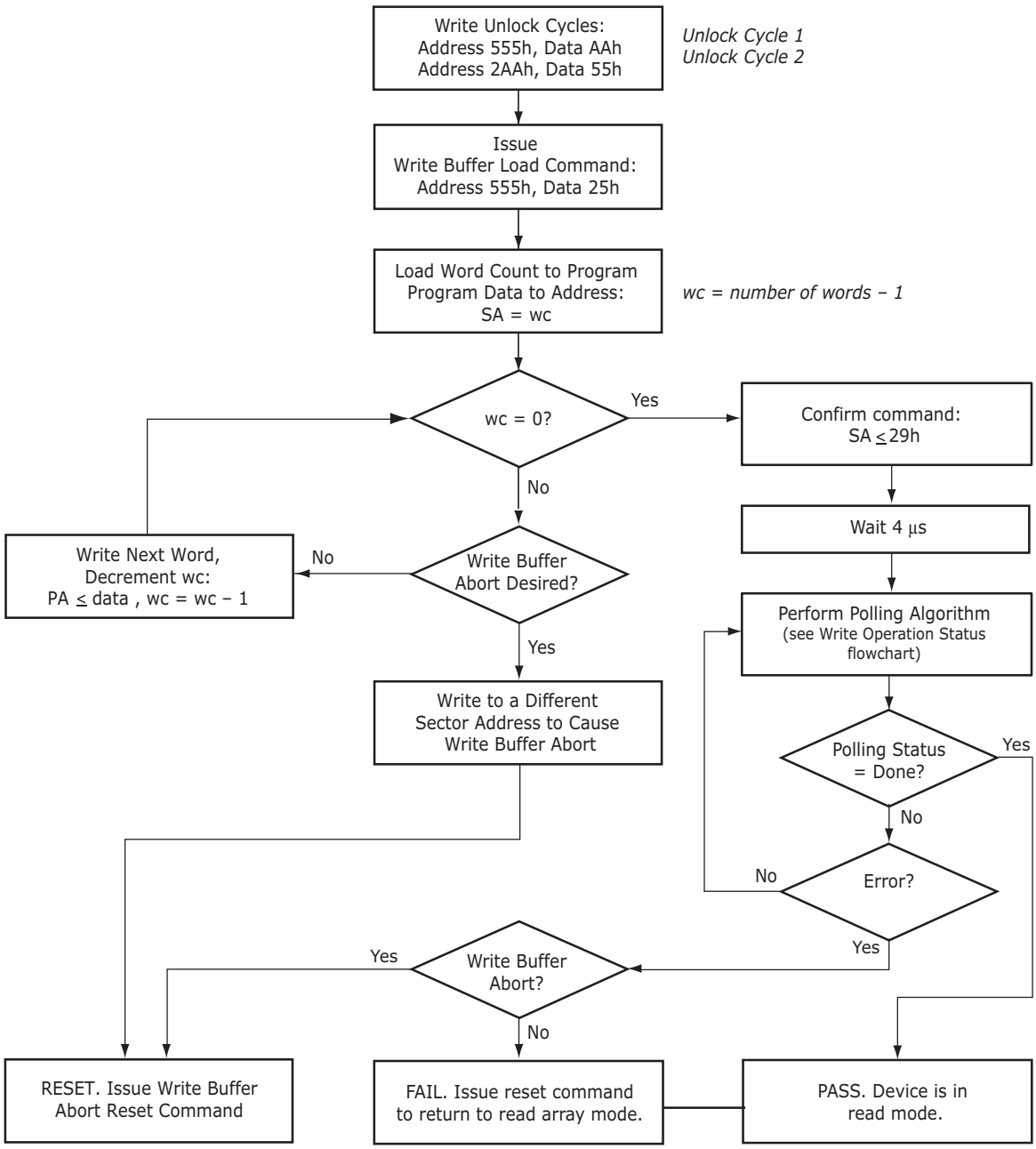


Figure 7.22. Write Buffer Programming Operation

7.5.3 Sector Erase

The sector erase function erases one or more sectors in the memory array. (See Table 12.1, Memory Array Commands; and Figure 7.24, Sector Erase Operation.) The device does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically programs and verifies the entire memory for an all zero data pattern prior to electrical erase. The system is not required to provide any controls or timings during these operations.

After the command sequence is written, a sector erase time-out of no less than t_{SEA} occurs. During the time-out period, additional sector addresses and sector erase commands may be written. Loading the sector erase buffer may be done in any sequence, and the number of

sectors may be from one sector to all sectors. The time between these additional cycles must be less than t_{SEA}. Any sector erase address and command following the exceeded time-out (t_{SEA}) may or may not be accepted. Any command other than Sector Erase or Erase Suspend during the time-out period resets that bank to the read mode. The system can monitor DQ3 to determine if the sector erase timer has timed out (See the "DQ3: Sector Erase Timer" section.) The time-out begins from the rising edge of the final WE# pulse in the command sequence.

When the Embedded Erase algorithm is complete, the bank returns to reading array data and addresses are no longer latched. Note that while the Embedded Erase operation is in progress, the system can read data from the non-erasing banks. The system can determine the status of the erase operation by reading DQ7 or DQ6/DQ2 in the erasing bank. Refer to "Write Operation Status" 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 7.24 illustrates the algorithm for the erase operation. Refer to the "Erase/Program Operations" section for parameters and timing diagrams.

Software Functions and Sample Code

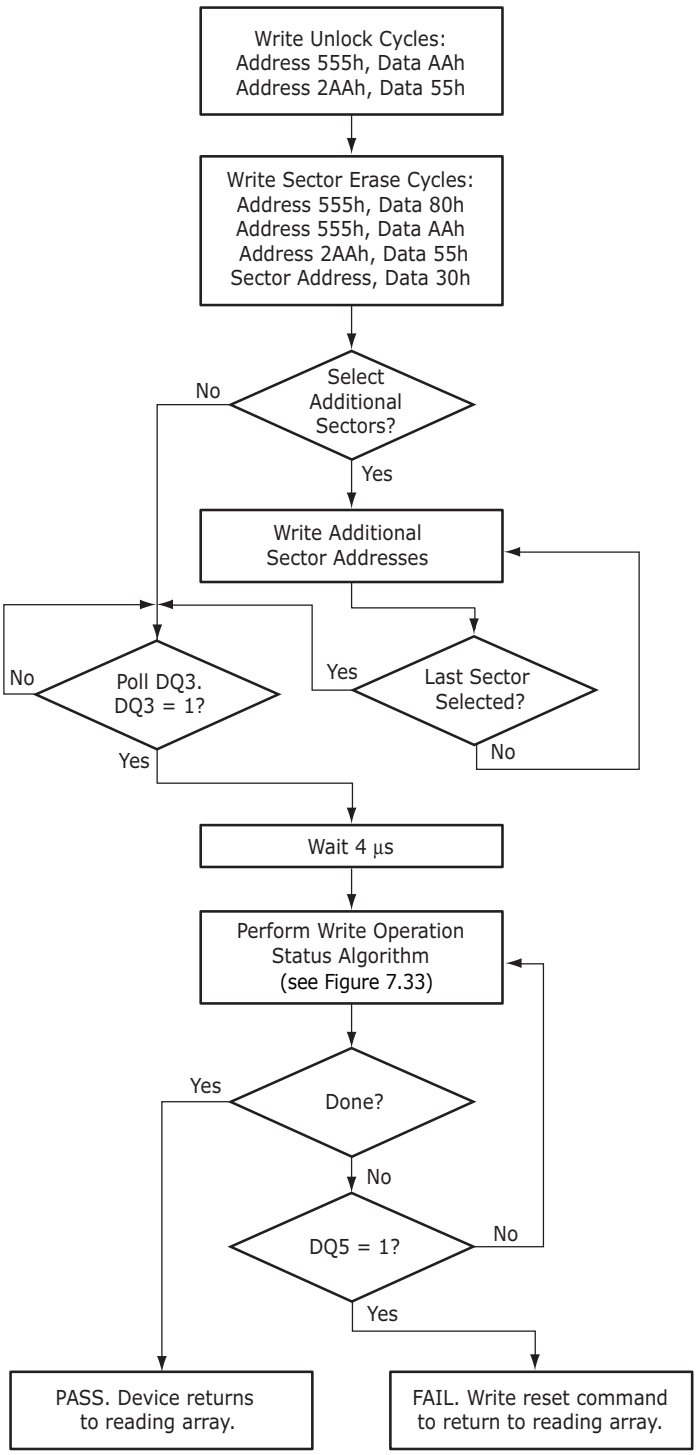
Table 7.23. Sector Erase

(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
Unlimited additional sectors may be selected for erase; command(s) must be written within t_{SEA}.					

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.amd.com and www.fujitsu.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      */
```



Unlock Cycle 1
Unlock Cycle 2

Command Cycle 1
Command Cycle 2
Command Cycle 3
Specify first sector for erasure

- *Each additional cycle must be written within t_{SEA} timeout*
- *Timeout resets after each additional cycle is written*
- *The host system may monitor DQ3 or wait t_{SEA} to ensure acceptance of erase commands*
- *No limit on number of sectors*
- *Commands other than Erase Suspend or selecting additional sectors for erasure during timeout reset device to reading array data*

Status may be obtained by reading DQ7, DQ6 and/or DQ2.

Error condition (Exceeded Timing Limits)

Notes:
1. *See Table 12.1 for erase command sequence.*
2. *See the section on DQ3 for information on the sector erase timeout.*

Figure 7.24. Sector Erase Operation

7.5.4 Chip Erase Command Sequence

Chip erase is a six-bus cycle operation as indicated by Table 12.1. 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. The system is not required to provide any controls or timings during these operations. The "Command Definition" section 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 "Write Operation Status" 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 7.25. 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.amd.com and www.fujitsu.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        */
```

7.5.5 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, including the minimum t_{SEA} time-out period during the sector erase command sequence. 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. How-

ever, when the Erase Suspend command is written during the sector erase time-out, the device immediately terminates the time-out period and suspends the erase operation.

After the erase operation has been suspended, the bank enters the erase-suspend-read mode. The system can read data from or program data to any sector not selected for erasure. (The device "erase suspends" all sectors selected for erasure.) Reading at any address within erase-suspended sectors produces status information on DQ7-DQ0. The system can use DQ7, or DQ6, and DQ2 together, to determine if a sector is actively erasing or is erase-suspended. Refer to Table 7.35 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 the "Write Buffer Programming Operation" section and the "Autoselect Command Sequence" section 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 7.26. Erase Suspend

(LLD Function = lld_EraseSuspendCmd)

Cycle	Operation	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.amd.com and www.fujitsu.com) for general information on Spansion Flash memory software development guidelines.

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

Table 7.27. Erase Resume

(LLD Function = lld_EraseResumeCmd)

Cycle	Operation	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.amd.com and www.fujitsu.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 */
```

7.5.6 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 SecSi 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 "Autoselect Command Sequence" for more information.

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

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

Software Functions and Sample Code

Table 7.28. Program Suspend

(LLD Function = lld_ProgramSuspendCmd)

Cycle	Operation	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.amd.com and www.fujitsu.com) for general information on Spansion Flash memory software development guidelines.

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

Table 7.29. Program Resume

(LLD Function = lld_ProgramResumeCmd)

Cycle	Operation	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.amd.com and www.fujitsu.com) for general information on Spansion Flash memory software development guidelines.

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

7.5.7 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 automatically enters the aforementioned Unlock Bypass mode and uses the higher voltage on the input to reduce the time required for program and erase operations. The system can then use the Write Buffer Load command sequence provided by the Unlock Bypass mode. Note that if a "Write-to-Buffer-Abort Reset" is required while in Unlock Bypass mode, the full 3-cycle RESET command sequence must be used to reset the device. 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.
- t_{ACC} locks all sector if set to V_{IL}; t_{ACC} should be set to V_{IH} for all other conditions.

7.5.8 Unlock Bypass

The device features an Unlock Bypass mode to facilitate faster word programming. Once the device enters the Unlock Bypass mode, only two write cycles are required to program data, instead of the normal four cycles.

This mode dispenses with the initial two unlock cycles required in the standard program command sequence, resulting in faster total programming time. The "Command Definition Summary" section shows the requirements for the unlock bypass command sequences.

During the unlock bypass mode, only the Read, Unlock Bypass Program and Unlock Bypass Reset commands are valid. To exit the unlock bypass mode, the system must issue the two-cycle unlock bypass reset command sequence. The first cycle must contain the bank address and the data 90h. The second cycle need only contain the data 00h. The bank then returns to the read mode.

Software Functions and Sample Code

The following are C source code examples of using the unlock bypass entry, program, and exit functions. Refer to the *Spansion Low Level Driver User's Guide* (available soon on www.amd.com and www.fujitsu.com) for general information on Spansion Flash memory software development guidelines.

Table 7.30. Unlock Bypass Entry

(LLD Function = lld_UnlockBypassEntryCmd)

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	Entry Command	Write	Base + AAAh	Base + 555h	0020h

```
/* Example: Unlock Bypass Entry Command     */
  *( (UINT16 *)bank_addr + 0x555 ) = 0x00AA;   /* write unlock cycle 1            */
  *( (UINT16 *)bank_addr + 0x2AA ) = 0x0055;   /* write unlock cycle 2            */
  *( (UINT16 *)bank_addr + 0x555 ) = 0x0020;   /* write unlock bypass command     */
  /* At this point, programming only takes two write cycles.   */
  /* Once you enter Unlock Bypass Mode, do a series of like    */
  /* operations (programming or sector erase) and then exit    */
  /* Unlock Bypass Mode before beginning a different type of   */
  /* operations.                                               */
```

Table 7.31. Unlock Bypass Program

(LLD Function = lld_UnlockBypassProgramCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Program Setup Command	Write	Base + xxxh	Base +xxxh	00A0h
2	Program Command	Write	Program Address	Program Address	Program Data

```
/* Example: Unlock Bypass Program Command  */
/* Do while in Unlock Bypass Entry Mode!   */
  *( (UINT16 *)bank_addr + 0x555 ) = 0x00A0;   /* write program setup command     */
  *( (UINT16 *)pa )                = data;     /* write data to be programmed     */
  /* Poll until done or error.    */
  /* If done and more to program, */
  /* do above two cycles again.   */
```

Table 7.32. Unlock Bypass Reset

(LLD Function = lld_UnlockBypassResetCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Reset Cycle 1	Write	Base + xxxh	Base +xxxh	0090h
2	Reset Cycle 2	Write	Base + xxxh	Base +xxxh	0000h

```
/* Example: Unlock Bypass Exit Command */
  *( (UINT16 *)base_addr + 0x000 ) = 0x0090;
  *( (UINT16 *)base_addr + 0x000 ) = 0x0000;
```

7.5.9 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, DQ3, 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 will return false status information.

During the Embedded Program algorithm, the device outputs on DQ7 the complement of the datum programmed to DQ7. This DQ7 status also applies to programming during Erase Suspend. When the Embedded Program algorithm is complete, the device outputs the datum programmed to DQ7. The system must provide the program address to read valid status information on DQ7. If a program address falls within a protected sector, Data# polling on DQ7 is active for approximately t_{PSP}, then that bank returns to the read mode.

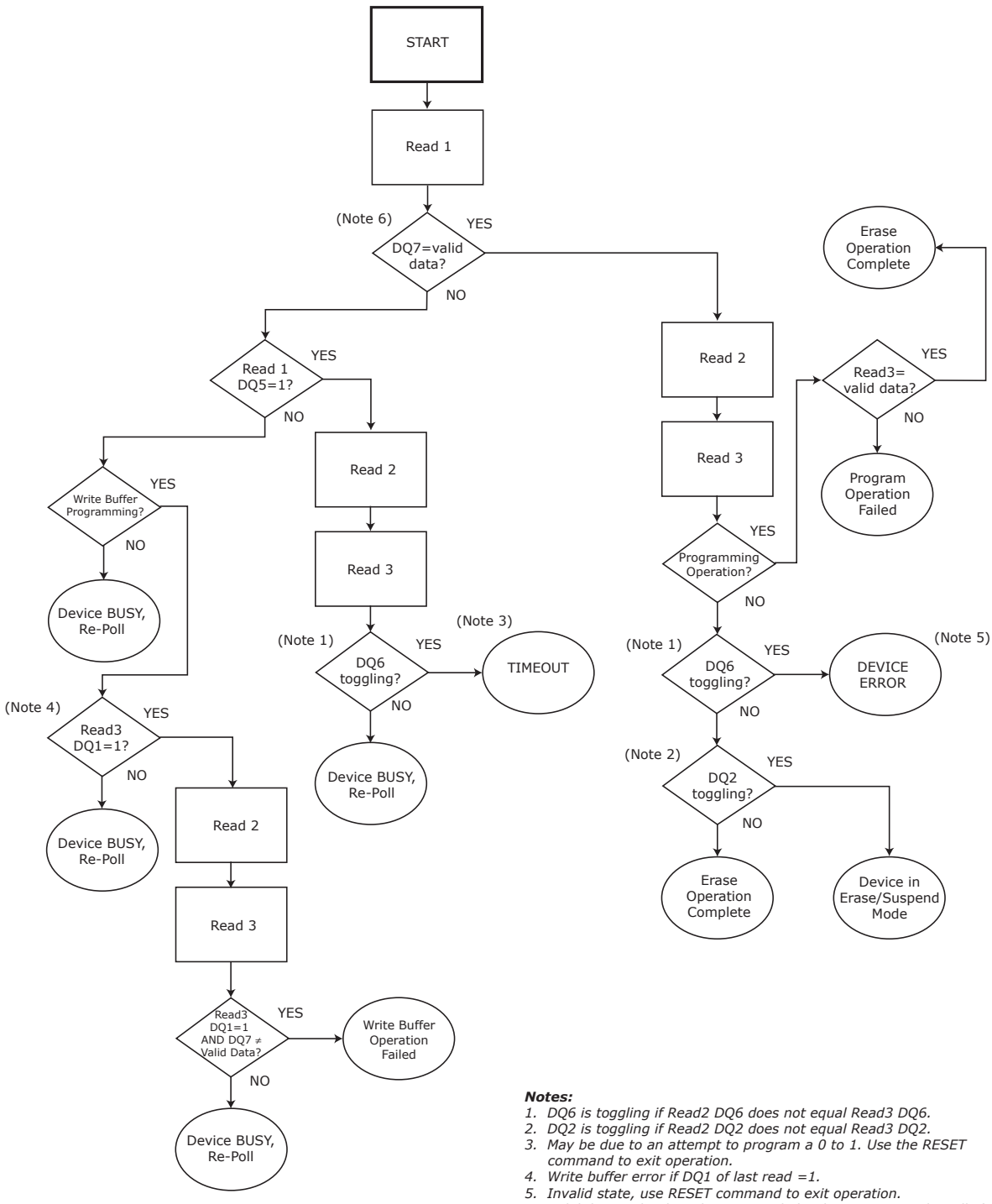
During the Embedded Erase Algorithm, Data# polling produces a "0" on DQ7. When the Embedded Erase algorithm is complete, or if the bank enters the Erase Suspend mode, Data# Polling produces a "1" on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status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

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

Just prior to the completion of an Embedded Program or Erase operation, DQ7 may change asynchronously with DQ6-DQ0 while Output Enable (OE#) is asserted low. That is, the device may change from providing status information to valid data on DQ7. Depending on when the

system samples the DQ7 output, it may read the status or valid data. Even if the device has completed the program or erase operation and DQ7 has valid data, the data outputs on DQ6-DQ0 may be still invalid. Valid data on DQ7-D00 will appear on successive read cycles.

See the following for more information: Table 7.35, Write Operation Status, shows the outputs for Data# Polling on DQ7. Figure 7.33, Write Operation Status Flowchart, shows the Data# Polling algorithm; and Figure 11.16, Data# Polling Timings (During Embedded Algorithm), shows the Data# Polling timing diagram.



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.*

Figure 7.33. Write Operation Status Flowchart

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

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

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

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

If a program address falls within a protected sector, DQ6 toggles for approximately t_{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 7.33, Write Operation Status Flowchart; Figure 11.17, Toggle Bit Timings (During Embedded Algorithm), and Tables 7.34 and 7.35.

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 those sectors that have 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 sectors are selected for erasure. Thus, both status bits are required for sector and mode information. Refer to Table 7.34 to compare outputs for DQ2 and DQ6. See the following for additional information: Figure 7.33, the "DQ6: Toggle Bit I" section, and Figures 11.16–11.19.

Table 7.34. 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 erases 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 7.33 for more details.

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).

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

After the sector erase command is written, the system should read the status of DQ7 (Data# Polling) or DQ6 (Toggle Bit I) to ensure that the device has accepted the command sequence, and then read DQ3. If DQ3 is "1," the Embedded Erase algorithm has begun; all further commands (except Erase Suspend) are ignored until the erase operation is complete. If DQ3 is "0," the device will accept additional sector erase commands. To ensure the command has been accepted, the system software should check the status of DQ3 prior to and following each sub-sequent sector erase command. If DQ3 is high on the second status check, the last command might not have been accepted. Table 7.35 shows the status of DQ3 relative to the other status bits.

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

Table 7.35. Write Operation Status

Status			DQ7 (Note 2)	DQ6	DQ5 (Note I)	DQ3	DQ2 (Note 2)	DQI (Note 4)
Standard Mode	Embedded Program Algorithm		DQ7#	Toggle	0	N/A	No toggle	0
	Embedded Erase Algorithm		0	Toggle	0	1	Toggle	N/A
Program Suspend Mode (Note 3)	Reading within Program Suspended Sector		INVALID *(Not Allowed)*	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	Data
Erase Suspend Mode	Erase-Suspend-Read	Erase Suspended Sector	1	No toggle	0	N/A	Toggle	N/A
		Non-Erase Suspended Sector	Data	Data	Data	Data	Data	Data
	Erase-Suspend-Program		DQ7#	Toggle	0	N/A	N/A	N/A
Write to Buffer (Note 5)	BUSY State		DQ7#	Toggle	0	N/A	N/A	0
	Exceeded Timing Limits		DQ7#	Toggle	1	N/A	N/A	0
	ABORT State		DQ7#	Toggle	0	N/A	N/A	1

Notes:

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

7.6 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 11.23, Back-to-Back Read/Write Cycle Timings, shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to the DC Characteristics (CMOS Compatible) table for read-while-program and read-while-erase current specification.

7.7 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, multiple sectors, or the entire device. Tables 6.1–6.3 indicate the address space that each sector occupies. The device address space is divided into sixteen banks: Banks 1 through 14 contain only 64 Kword sectors, while Banks 0 and 15 contain both 16 Kword boot sectors in addition to 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 "DC Characteristics" represents the active current specification for the write mode. "AC Characteristics-Synchronous" and "AC Characteristics-Asynchronous" contain timing specification tables and timing diagrams for write operations.

7.8 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 7.15, Configuration Register).

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.

7.9 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 will be 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 Figures 11.5 and 11.12 for timing diagrams.

7.10 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:

1. to exit Autoselect mode
2. when DQ5 goes high during write status operation that indicates program or erase cycle was not successfully completed
3. exit sector lock/unlock operation.
4. to return to erase-suspend-read mode if the device was previously in Erase Suspend mode.
5. after any aborted operations

Software Functions and Sample Code

Table 7.36. Reset

(LLD Function = lld_ResetCmd)

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

Note: *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.amd.com and www.fujitsu.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 will not work during this condition.
- To exit the unlock bypass mode, the system must issue a two-cycle unlock bypass reset command sequence [see command table for details].

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.

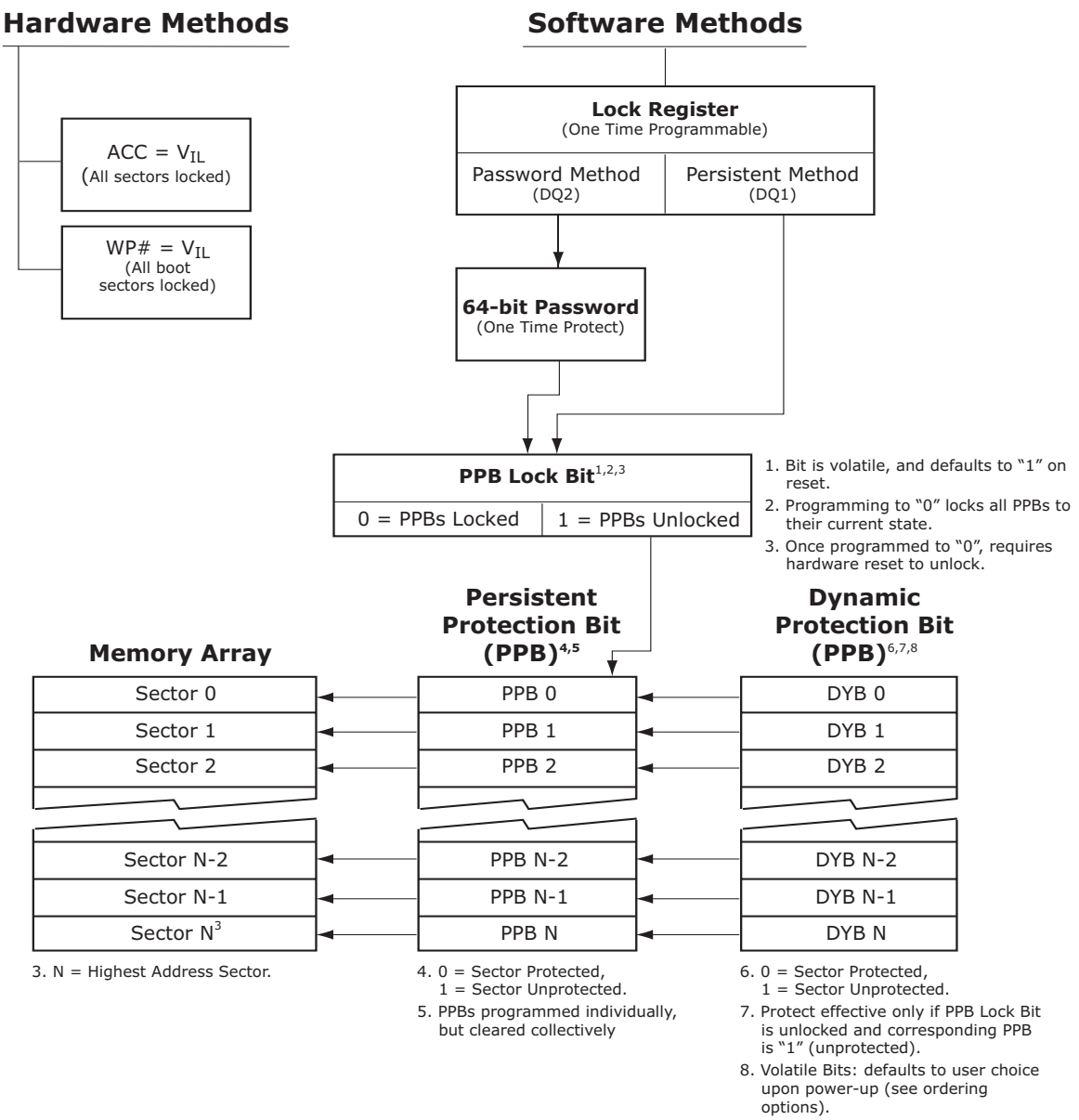


Figure 8.I. Advanced Sector Protection/Unprotection

8.1 Lock Register

As shipped from the factory, all devices default to the persistent mode when power is applied, and all sectors are unprotected, unless otherwise chosen through the DYB ordering option (see Ordering Information). The device programmer or host system must then choose which sector protection method to use. Programming (setting to "0") any one of the following two one-time programmable, non-volatile bits locks the part permanently in that mode:

- Lock Register Persistent Protection Mode Lock Bit (DQ1)
- Lock Register Password Protection Mode Lock Bit (DQ2)

Table 8.1. Lock Register

Device	DQI5-05	DQ4	DQ3	DQ2	DQI	DQ0
S29WS256N	1	1	1	Password Protection Mode Lock Bit	Persistent Protection Mode Lock Bit	Customer SecSi Sector Protection Bit
S29WS128N/ S29WS064N	Undefined	DYB Lock Boot Bit 0 = sectors power up protected 1 = sectors power up unprotected	PPB One-Time Programmable Bit 0 = All PPB erase command disabled 1 = All PPB Erase command enabled	Password Protection Mode Lock Bit	Persistent Protection Mode Lock Bit	SecSi Sector Protection Bit

For programming lock register bits refer to Table 12.2.

Notes

1. If the password mode is chosen, the password must be programmed before setting the corresponding lock register bit.
2. After the Lock Register Bits Command Set Entry command sequence is written, reads and writes for Bank 0 are disabled, while reads from other banks are allowed until exiting this mode.
3. If both lock bits are selected to be programmed (to zeros) at the same time, the operation will abort.
4. Once the Password Mode Lock Bit is programmed, the Persistent Mode Lock Bit is permanently disabled, and no changes to the protection scheme are allowed. Similarly, if the Persistent Mode Lock Bit is programmed, the Password Mode is permanently disabled.

After selecting a sector protection method, each sector can operate in any of the following three states:

1. *Constantly locked.* The selected sectors are protected and can not be reprogrammed unless PPB lock bit is cleared via a password, hardware reset, or power cycle.
2. *Dynamically locked.* The selected sectors are protected and can be altered via software commands.
3. *Unlocked.* The sectors are unprotected and can be erased and/or programmed.

These states are controlled by the bit types described in Sections 8.2–8.6.

8.2 Persistent Protection Bits

The Persistent Protection Bits are unique and nonvolatile for each sector and have the same endurances as the Flash memory. Preprogramming and verification prior to erasure are handled by the device, and therefore do not require system monitoring.

Notes

1. Each PPB is individually programmed and all are erased in parallel.
2. Entry command disables reads and writes for the bank selected.
3. Reads within that bank will return the PPB status for that sector.
4. Reads from other banks are allowed while writes are not allowed.

5. All Reads must be performed using the Asynchronous mode.
6. The specific sector address (A23-A14 WS256N, A22-A14 WS128N, A21-A14 WS064N) are written at the same time as the program command.
7. If the PPB Lock Bit is set, the PPB Program or erase command will not execute and will time-out without programming or erasing the PPB.
8. There are no means for individually erasing a specific PPB and no specific sector address is required for this operation.
9. Exit command must be issued after the execution which resets the device to read mode and re-enables reads and writes for Bank 0
10. The programming state of the PPB for a given sector can be verified by writing a PPB Status Read Command to the device as described by the flow chart below.

8.3 Dynamic Protection Bits

Dynamic Protection Bits are volatile and unique for each sector and can be individually modified. DYBs only control the protection scheme for unprotected sectors that have their PPBs cleared (erased to "1"). By issuing the DYB Set or Clear command sequences, the DYBs will be set (programmed to "0") or cleared (erased to "1"), thus placing each sector in the protected or unprotected state respectively. 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.

When the parts are first shipped, the PPBs are cleared (erased to "1") and upon power up or reset, the DYBs can be set or cleared depending upon the ordering option chosen.

2. If the option to clear the DYBs after power up is chosen, (erased to "1"), then the sectors may be modified depending upon the PPB state of that sector (see Table 8.2).
3. The sectors would be in the protected state If the option to set the DYBs after power up is chosen (programmed to "0").
4. It is possible to have sectors that are persistently locked with sectors that are left in the dynamic state.
5. The DYB Set or Clear commands for the dynamic sectors signify protected or unprotectedstate of the sectors respectively. However, if there is a need to change the status of the persistently locked sectors, a few more steps are required. First, the PPB Lock Bit must be cleared by either putting the device through a power-cycle, or hardware reset. The PPBs can then be changed to reflect the desired settings. Setting the PPB Lock Bit once again will lock the PPBs, and the device operates normally again.
6. To achieve the best protection, it is recommended to execute the PPB Lock Bit Set command early in the boot code and protect the boot code by holding WP# = V_{IL}. Note that the PPB and DYB bits have the same function when ACC = V_{HH} as they do when ACC =V_{IH}.

8.4 Persistent Protection Bit Lock Bit

The Persistent Protection Bit Lock Bit is a global volatile bit for all sectors. When set (programmed to "0"), this bit locks all PPB and when cleared (programmed to "1"), unlocks each sector. There is only one PPB Lock Bit per device.

Notes

1. No software command sequence unlocks this bit unless the device is in the password protection mode; only a hardware reset or a power-up clears this bit.
2. The PPB Lock Bit must be set (programmed to "0") only after all PPBs are configured to the desired settings.

8.5 Password Protection Method

The Password Protection Method allows an even higher level of security than the Persistent Sector Protection Mode by requiring a 64 bit password for unlocking the device PPB Lock Bit. In addition to this password requirement, after power up and reset, the PPB Lock Bit is set "0" to maintain the password mode of operation. Successful execution of the Password Unlock command by entering the entire password clears the PPB Lock Bit, allowing for sector PPBs modifications.

Notes

1. There is no special addressing order required for programming the password. Once the Password is written and verified, the Password Mode Locking Bit must be set in order to prevent access.
2. The Password Program Command is only capable of programming "0"s. Programming a "1" after a cell is programmed as a "0" results in a time-out with the cell as a "0".
3. The password is all "1"s when shipped from the factory.
4. All 64-bit password combinations are valid as a password.
5. There is no means to verify what the password is after it is set.
6. The Password Mode Lock Bit, once set, prevents reading the 64-bit password on the data bus and further password programming.
7. The Password Mode Lock Bit is not erasable.
8. The lower two address bits (A1–A0) are valid during the Password Read, Password Program, and Password Unlock.
9. The exact password must be entered in order for the unlocking function to occur.
10. The Password Unlock command cannot be issued any faster than 1 µs at a time to prevent a hacker from running through all the 64-bit combinations in an attempt to correctly match a password.
11. Approximately 1 µs is required for unlocking the device after the valid 64-bit password is given to the device.
12. Password verification is only allowed during the password programming operation.
13. All further commands to the password region are disabled and all operations are ignored.
14. If the password is lost after setting the Password Mode Lock Bit, there is no way to clear the PPB Lock Bit.
15. Entry command sequence must be issued prior to any of any operation and it disables reads and writes for Bank 0. Reads and writes for other banks excluding Bank 0 are allowed.
16. If the user attempts to program or erase a protected sector, the device ignores the command and returns to read mode.
17. A program or erase command to a protected sector enables status polling and returns to read mode without having modified the contents of the protected sector.
18. The programming of the DYB, PPB, and PPB Lock for a given sector can be verified by writing individual status read commands DYB Status, PPB Status, and PPB Lock Status to the device.

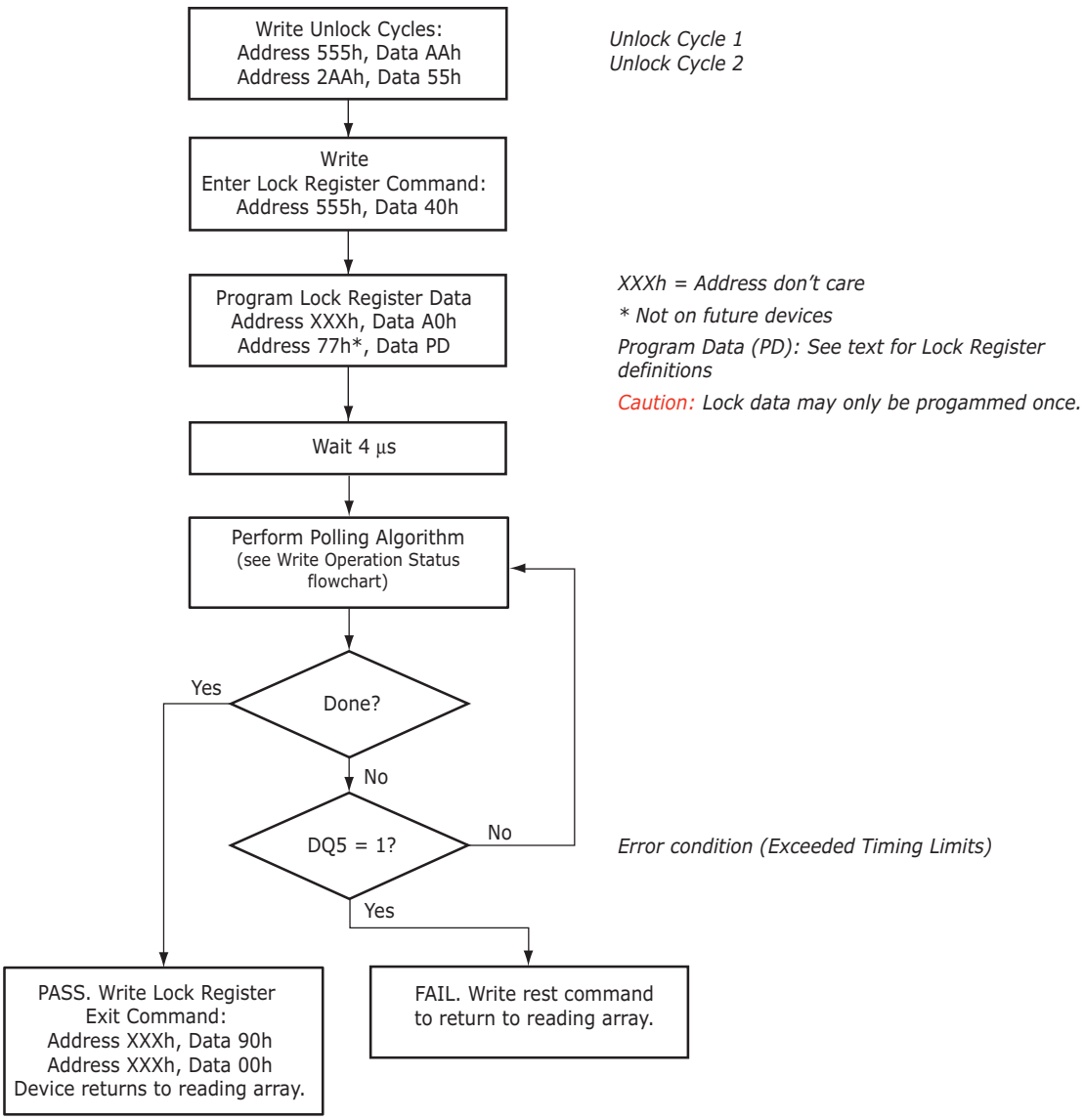


Figure 8.2. Lock Register Program Algorithm

8.6 Advanced Sector Protection Software Examples

Table 8.2. Sector Protection Schemes

Unique Device PPB Lock Bit 0 = locked I = unlocked		Sector PPB 0 = protected I = unprotected	Sector DYB 0 = protected I = unprotected	Sector Protection Status
Any Sector	0	0	x	Protected through PPB
Any Sector	0	0	x	Protected through PPB
Any Sector	0	1	1	Unprotected
Any Sector	0	1	0	Protected through DYB
Any Sector	1	0	x	Protected through PPB
Any Sector	1	0	x	Protected through PPB
Any Sector	1	1	0	Protected through DYB
Any Sector	1	1	1	Unprotected

Table 8.2 contains all possible combinations of the DYB, PPB, and PPB Lock Bit relating to the status of the sector. In summary, if the PPB Lock Bit is locked (set to "0"), no changes to the PPBs are allowed. The PPB Lock Bit can only be unlocked (reset to "1") through a hardware reset or power cycle. See also Figure 8.1 for an overview of the Advanced Sector Protection feature.

8.7 Hardware Data Protection Methods

The device offers two main types of data protection at the sector level via hardware control:

- When WP# is at V_{IL}, the four outermost sectors are locked (device specific).
- 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 describes these methods:

8.7.1. WP# Method

The Write Protect feature provides a hardware method of protecting the four outermost sectors. This function is provided by the WP# pin and overrides the previously discussed Sector Protection/Unprotection method.

If the system asserts V_{IL} on the WP# pin, the device disables program and erase functions in the "outermost" boot sectors. The outermost boot sectors are the sectors containing both the lower and upper set of sectors in a dual-boot-configured device.

If the system asserts V_{IH} on the WP# pin, the device reverts to whether the boot sectors were last set to be protected or unprotected. That is, sector protection or unprotection for these sectors depends on whether they were last protected or unprotected.

Note that the WP# pin must not be left floating or unconnected as inconsistent behavior of the device may result.

The WP# pin must be held stable during a command sequence execution

8.7.2 ACC Method

This method is similar to above, except it protects all sectors. Once ACC input is set to V_{IL}, all program and erase functions are disabled and hence all sectors are protected.

8.7.3 Low V_{CC} Write Inhibit

When V_{CC} is less than V_{LKO}, the device does not accept any write cycles. This protects data during V_{CC} power-up and power-down.

The command register and all internal program/erase circuits are disabled, and the device resets to 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}.

8.7.4 Write Pulse "Glitch Protection"

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

8.7.5 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} ± 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 "DC Characteristics" 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" 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} ± 0.2 V, the device draws CMOS standby current (I_{CC4}). If RESET# is held at V_{IL} but not within V_{SS} ± 0.2 V, the standby current will be 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 SecSi™ (Secured Silicon) Sector Flash Memory Region

The SecSi (Secured Silicon) Sector provides an extra Flash memory region that enables permanent part identification through an Electronic Serial Number (ESN). The SecSi Sector is 256 words in length that consists of 128 words for factory data and 128 words for customer-secured areas. All SecSi reads outside of the 256-word address range will return invalid data. The Factory Indicator Bit, DQ7, (at Autoselect address 03h) is used to indicate whether or not the Factory SecSi Sector is locked when shipped from the factory. The Customer Indicator Bit (DQ6) is used to indicate whether or not the Customer SecSi Sector is locked when shipped from the factory.

Please note the following general conditions:

- While SecSi 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 SecSi Sector will wrap from address FFh back to address 00h.
- Reads outside of sector 0 will return memory array data.
- Continuous burst read past the maximum address is undefined.
- Sector 0 is remapped from memory array to SecSi Sector array.
- Once the SecSi Sector Entry Command is issued, the SecSi Sector Exit command must be issued to exit SecSi Sector Mode.
- The SecSi Sector is not accessible when the device is executing an Embedded Program or Embedded Erase algorithm.

Table 10.1. SecSi™ Sector Addresses

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

10.1 Factory SecSi™ Sector

The Factory SecSi 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.

These devices are available pre programmed with one of the following:

- A random, 8 Word secure ESN only within the Factory SecSi Sector
- Customer code within the Customer SecSi 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 SecSi Sector and Customer SecSi Sector permanently locked. Contact your local representative for details on using Spansion programming services.

10.2 Customer SecSi™ Sector

The Customer SecSi 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 SecSi Sector can be treated as an additional Flash memory space.

Please note the following:

- Once the Customer SecSi Sector area is protected, the Customer Indicator Bit will be permanently set to "1."
- The Customer SecSi Sector can be read any number of times, but can be programmed and locked only once. The Customer SecSi Sector lock must be used with caution as once locked, there is no procedure available for unlocking the Customer SecSi Sector area and none of the bits in the Customer SecSi Sector memory space can be modified in any way.
- The accelerated programming (ACC) and unlock bypass functions are not available when programming the Customer SecSi Sector, but reading in Banks 1 through 15 is available.
- Once the Customer SecSi Sector is locked and verified, the system must write the Exit SecSi Sector Region command sequence which return the device to the memory array at sector 0.

10.3 SecSi™ Sector Entry and SecSi Sector Exit Command Sequences

The system can access the SecSi Sector region by issuing the three-cycle Enter SecSi Sector command sequence. The device continues to access the SecSi Sector region until the system issues the four-cycle Exit SecSi Sector command sequence.

See Command Definition Table [SecSi™ Sector Command Table, Appendix Table 12.1 for address and data requirements for both command sequences.

The SecSi Sector Entry Command allows the following commands to be executed

- Read customer and factory SecSi areas
- Program the customer SecSi Sector

After the system has written the Enter SecSi Sector command sequence, it may read the SecSi 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 SecSi Sector command sequence, or until power is removed from the device.

Software Functions and Sample Code

The following are C functions and source code examples of using the SecSi Sector Entry, Program, and exit commands. Refer to the *Spansion Low Level Driver User's Guide* (available soon on www.amd.com and www.fujitsu.com) for general information on Spansion Flash memory software development guidelines.

Table l0.2. SecSi 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: *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 I0.3. SecSi Sector Program

(LLD Function = lld_ProgramCmd)

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
Program Setup	Write	Base + AAAh	Base + 555h	00A0h
Program	Write	Word Address	Word Address	Data Word

Note: *Base = Base Address.*

```
/* Once in the SecSi Sector mode, you program */
/* words using the programming algorithm.     */
```

Table I0.4. SecSi Sector Entry

(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: *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 */
```

II Electrical Specifications

11.1 Absolute Maximum Ratings

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) . . . . . –0.5 V to V_{IO} + 0.5 V
V_{CC} (Note 1). . . . . –0.5 V to +2.5 V
V_{IO} . . . . . –0.5 V to +2.5 V
ACC (Note 2) . . . . . –0.5 V to +9.5 V
Output Short Circuit Current (Note 3) . . . . . 100 mA

Notes:

1. *Minimum DC voltage on input or I/Os is –0.5 V. During voltage transitions, inputs or I/Os may undershoot V_{SS} to –2.0 V for periods of up to 20 ns. See Figure 11.1. Maximum DC voltage on input or I/Os is V_{CC} + 0.5 V. During voltage transitions outputs may overshoot to V_{CC} + 2.0 V for periods up to 20 ns. See Figure 11.2.*
2. *Minimum DC input voltage on pin ACC is -0.5V. During voltage transitions, ACC may overshoot V_{SS} to –2.0 V for periods of up to 20 ns. See Figure 11.1. Maximum DC voltage on pin ACC is +9.5 V, which may overshoot to 10.5 V for periods up to 20 ns.*
3. *No more than one output may be shorted to ground at a time. Duration of the short circuit should not be greater than one second.*
4. *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this data sheet is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.*

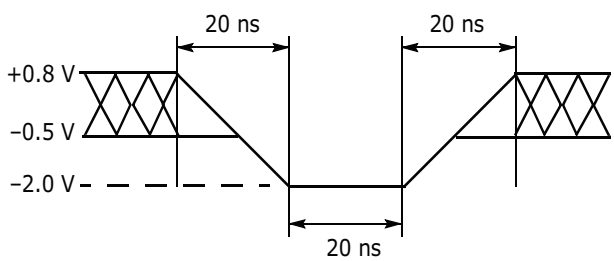


Figure II.1. Maximum Negative Overshoot Waveform

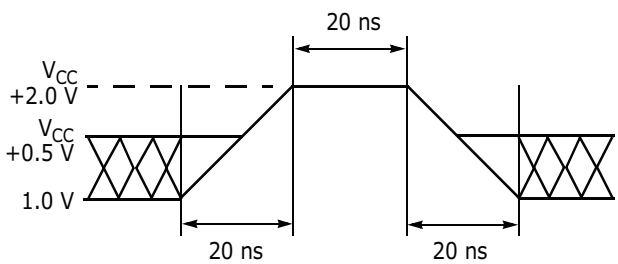


Figure II.2. Maximum Positive Overshoot Waveform

11.2 Operating Ranges

Wireless (W) Devices

Ambient Temperature (T_A) . . . . . –25°C to +85°C

Industrial (I) Devices

Ambient Temperature (T_A) . . . . . –40°C to +85°C

Supply Voltages

V_{CC} Supply Voltages . . . . . +1.70 V to +1.95 V

V_{IO} Supply Voltages: . . . . . +1.70 V to +1.95 V
(Contact local sales office for V_{IO} = 1.35 to +1.70 V.)

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

11.3 Test Conditions

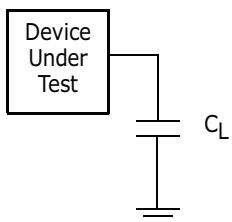


Figure 11.3. Test Setup

Table 11.1. Test Specifications

Test Condition	All Speed Options	Unit
Output Load Capacitance, C_L (including jig capacitance)	30	pF
Input Rise and Fall Times	3.0 @ 54, 66 MHz 2.5 @ 80 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)

11.5 Switching Waveforms

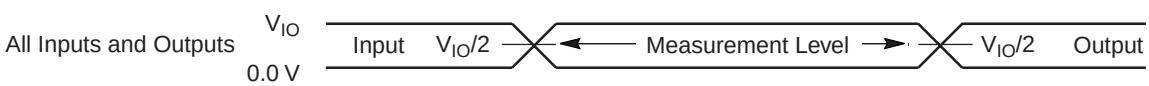


Figure 11.4. Input Waveforms and Measurement Levels

11.6 V_{CC} Power-up

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

Notes:

1. *V_{CC} >= V_{IO} - 100mV and V_{CC} ramp rate is > 1V / 100µs*
2. *V_{CC} ramp rate <1V / 100µs, a Hardware Reset will be required.*

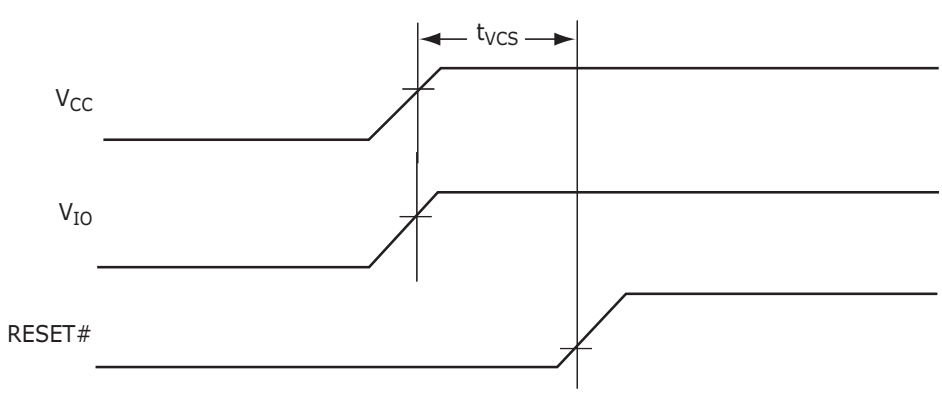


Figure 11.5. V_{CC} Power-up Diagram

11.7 DC Characteristics (CMOS Compatible)

Parameter	Description (Notes)	Test Conditions (Notes 1, 2, 9)		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 (3)	$V_{OUT} = V_{SS}$ 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	54 MHz		27	54	mA
			66 MHz		28	60	mA
			80 MHz		30	66	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 16	54 MHz		28	48	mA
			66 MHz		30	54	mA
			80 MHz		32	60	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 32	54 MHz		29	42	mA
			66 MHz		32	48	mA
			80 MHz		34	54	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = Continuous	54 MHz		32	36	mA
			66 MHz		35	42	mA
			80 MHz		38	48	mA
I_{IO1}	V_{IO} Non-active Output	OE# = V_{IH}			20	30	µA
I_{CC1}	V_{CC} Active Asynchronous Read Current (4)	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}	10 MHz		27	36	mA
			5 MHz		13	18	mA
			1 MHz		3	4	mA
I_{CC2}	V_{CC} Active Write Current (5)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}	V_{ACC}		1	5	µA
			V_{CC}		19	52.5	mA
I_{CC3}	V_{CC} Standby Current (6, 7)	CE# = RESET# = V_{CC} ± 0.2 V	V_{ACC}		1	5	µA
			V_{CC}		20	40	µA
I_{CC4}	V_{CC} Reset Current (7)	RESET# = V_{IL}, CLK = V_{IL}			70	150	µA
I_{CC5}	V_{CC} Active Current (Read While Write) (7)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}			50	60	mA
I_{CC6}	V_{CC} Sleep Current (7)	CE# = V_{IL}, OE# = V_{IH}			2	40	µA
I_{ACC}	Accelerated Program Current (8)	CE# = V_{IL}, OE# = V_{IH}, V_{ACC} = 9.5 V	V_{ACC}		6	20	mA
			V_{CC}		14	20	mA
V_{IL}	Input Low Voltage	V_{IO} = 1.8 V		–0.5		0.4	V
V_{IH}	Input High Voltage	V_{IO} = 1.8 V		V_{IO} – 0.4		V_{IO} + 0.4	V
V_{OL}	Output Low Voltage	I_{OL} = 100 µA, $V_{CC} = V_{CC\ min} = V_{IO}$				0.1	V
V_{OH}	Output High Voltage	I_{OH} = –100 µA, $V_{CC} = V_{CC\ min} = V_{IO}$		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		1.4	V

Notes:

1. *Maximum I_{CC} specifications are tested with $V_{CC} = V_{CC}$max.*
2. *$V_{CC} = V_{IO}$.*
3. *CE# must be set high when measuring the RDY pin.*
4. *The I_{CC} current listed is typically less than 3 mA/MHz, with OE# at V_{IH}.*
5. *I_{CC} active while Embedded Erase or Embedded Program is in progress.*
6. *Device enters automatic sleep mode when addresses are stable for t_{ACC} + 20 ns. Typical sleep mode current is equal to I_{CC3}.*
7. *$V_{IH} = V_{CC}$ ± 0.2 V and V_{IL} > –0.1 V.*
8. *Total current during accelerated programming is the sum of V_{ACC} and V_{CC} currents.*
9. *$V_{ACC} = V_{HH}$ on ACC input.*

11.8 AC Characteristics

11.8.1. CLK Characterization

Parameter	Description		54 MHz	66 MHz	80 MHz	Unit
f_{CLK}	CLK Frequency	Max	54	66	80	MHz
t_{CLK}	CLK Period	Min	18.5	15.1	12.5	ns
t_{CH}	CLK High Time	Min	7.4	6.1	5.0	ns
t_{CL}	CLK Low Time					
t_{CR}	CLK Rise Time	Max	3	3	2.5	ns
t_{CF}	CLK Fall Time					

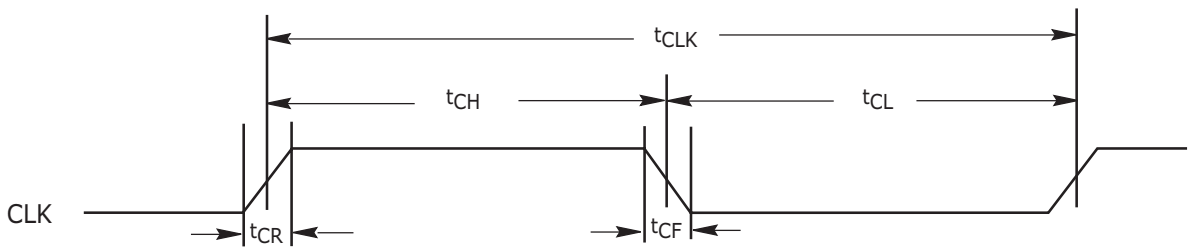


Figure 11.6. CLK Characterization

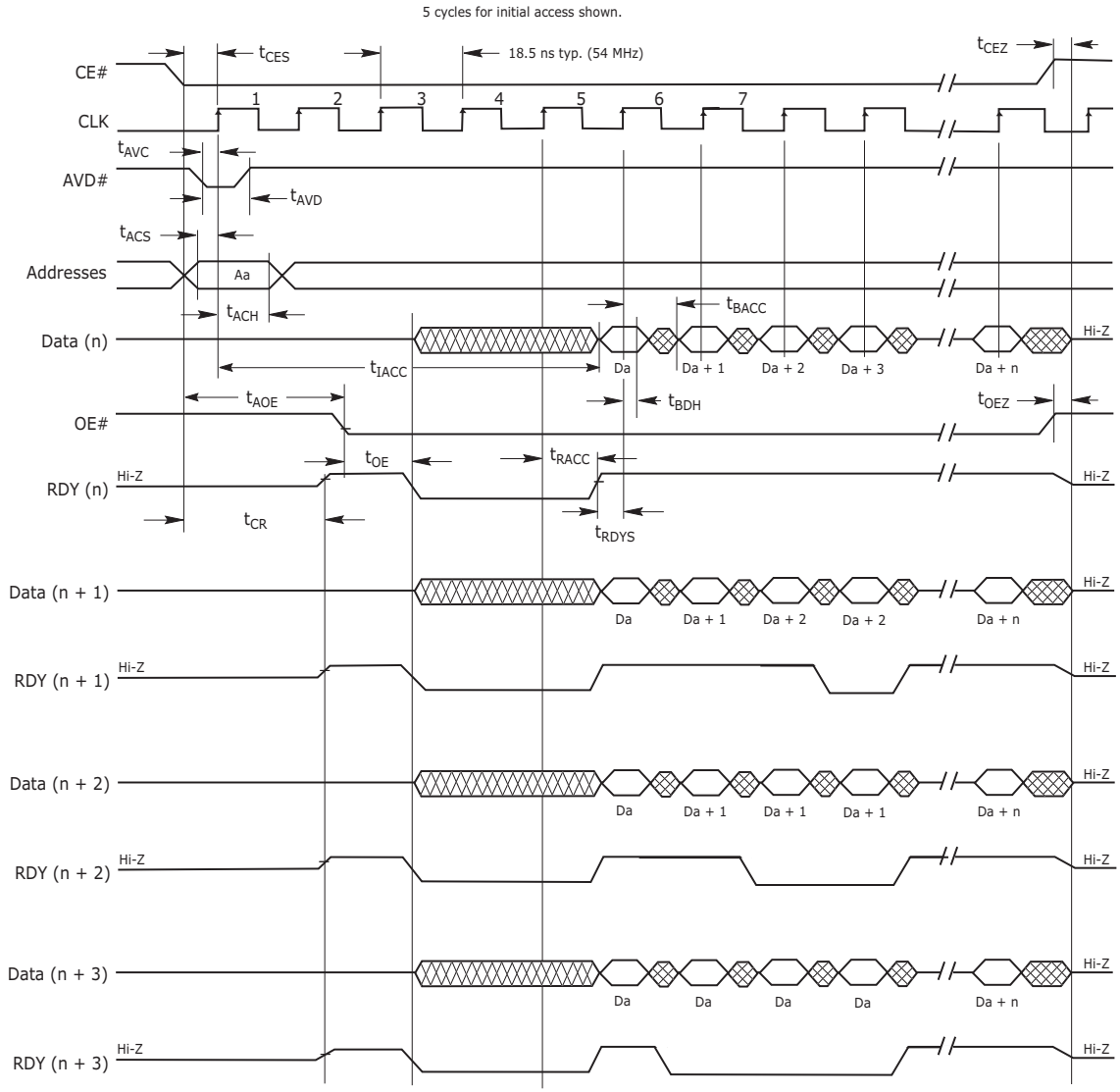
11.8.2 Synchronous/Burst Read

Parameter		Description		54 MHz	66 MHz	80 MHz	Unit
JEDEC	Standard						
	t_{IACC}	Latency	Max	69			ns
	t_{BACC}	Burst Access Time Valid Clock to Output Delay	Max	13.5	11.2	9	ns
	t_{ACS}	Address Setup Time to CLK (Note 1)	Min	5	4		ns
	t_{ACH}	Address Hold Time from CLK (Note 1)	Min	7	6		ns
	t_{BDH}	Data Hold Time from Next Clock Cycle	Min	4	3		ns
	t_{CR}	Chip Enable to RDY Valid	Max	13.5	11.2	9	ns
	t_{OE}	Output Enable to Output Valid	Max	13.5	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			ns
	t_{RDYS}	RDY Setup Time to CLK	Min	5	4	3.5	ns
	t_{RACC}	Ready Access Time from CLK	Max	13.5	11.2	9	ns
	t_{CAS}	CE# Setup Time to AVD#	Min	0			ns
	t_{AVC}	AVD# Low to CLK	Min	4			ns
	t_{AVD}	AVD# Pulse	Min	8			ns
	t_{AOE}	AVD Low to OE# Low	Max	38.4			ns

Notes:

1. *Addresses are latched on the first rising edge of CLK.*
2. *Not 100% tested.*

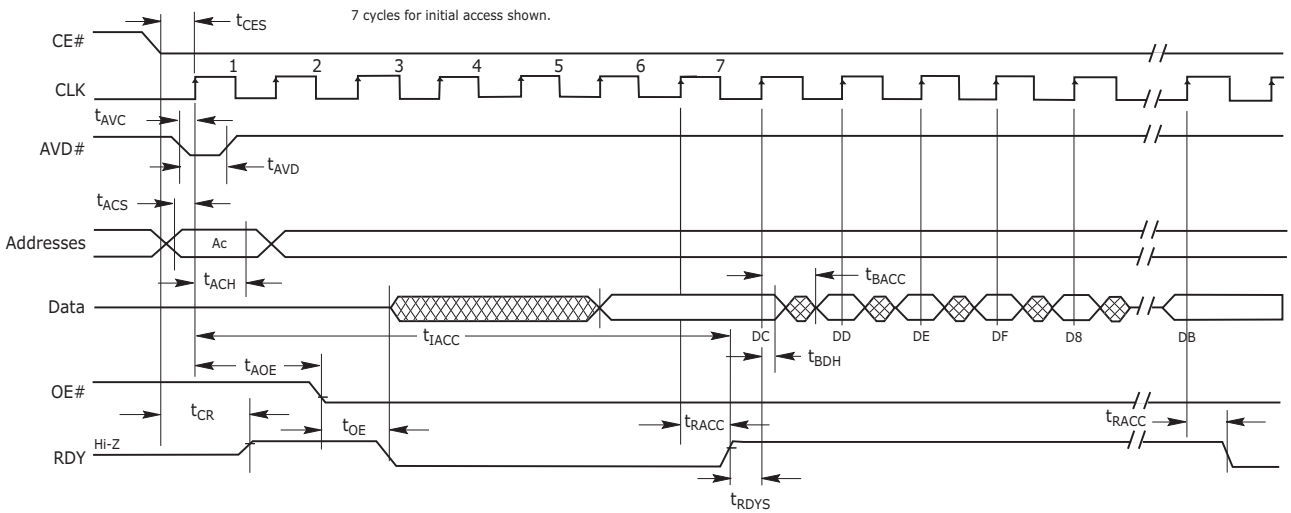
11.8.3 Timing Diagrams



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.*

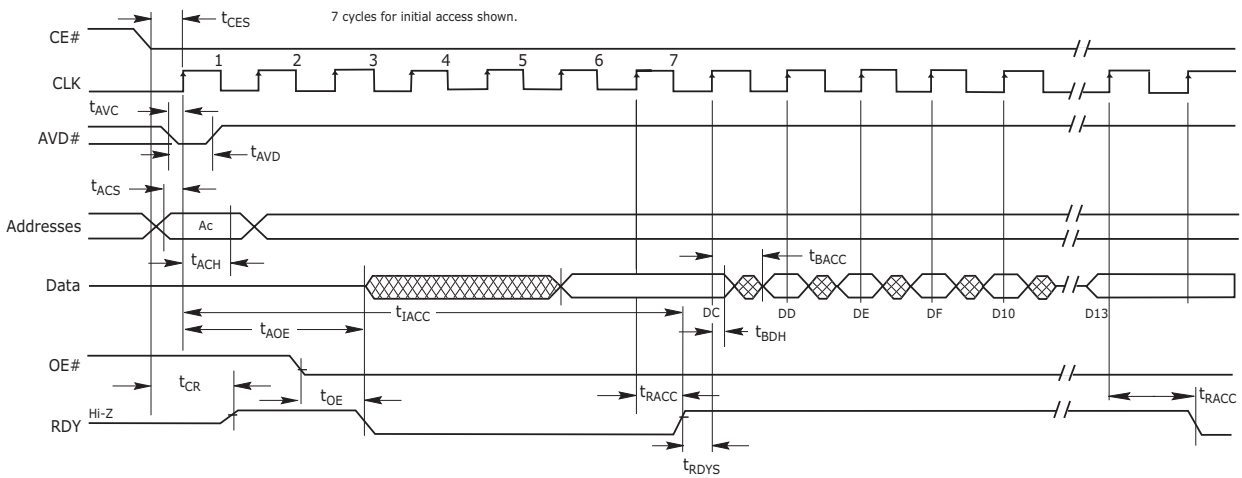
Figure 11.7. CLK Synchronous Burst Mode Read



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).*

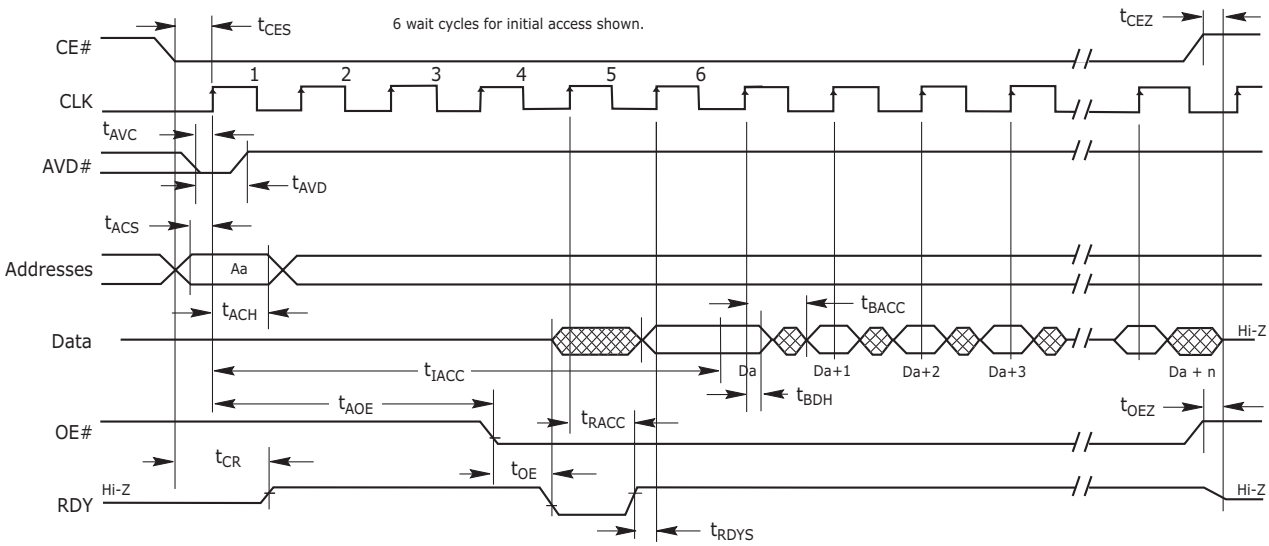
Figure 11.8. 8-word Linear Burst with Wrap Around



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. Clock is set for active rising edge.*
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 asynchronous mode with out wrap around.*
4. *DC–D13 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 1st address in range (c-13).*

Figure 11.9. 8-word Linear Burst without Wrap Around



Notes:

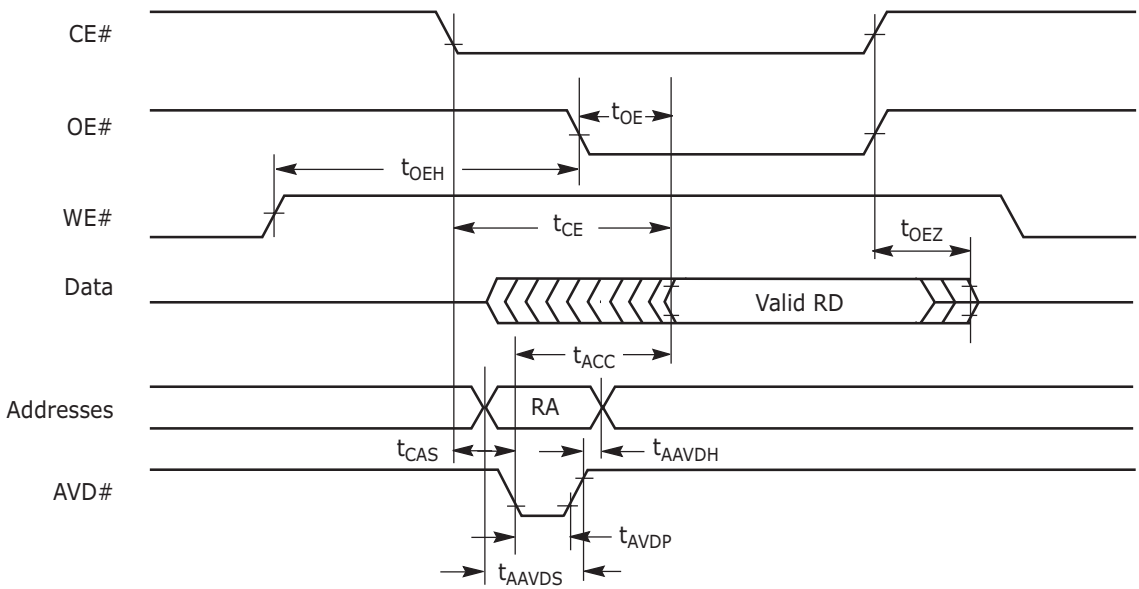
1. *Figure assumes 6 wait states for initial access and synchronous read.*
2. *The Set Configuration Register command sequence has been written with CR8=0; device will output RDY one cycle before valid data.*

Figure 11.10. Linear Burst with RDY Set One Cycle Before Data

11.8.4 AC Characteristics—Asynchronous Read

Parameter		Description			54 MHz	66 MHz	80 MHz	Unit
JEDEC	Standard							
	t_{CE}	Access Time from CE# Low		Max	70			ns
	t_{ACC}	Asynchronous Access Time		Max	70			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	7	6		ns
	t_{OE}	Output Enable to Output Valid		Max	13.5	11.2		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 (see Note)		Max	10			ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0			ns

Note: *Not 100% tested.*



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

Figure 11.11. Asynchronous Mode Read

11.8.5 Hardware Reset (RESET#)

Parameter		Description		All Speed Options	Unit
JEDEC	Std.				
	t_{RP}	RESET# Pulse Width	Min	30	µs
	t_{RH}	Reset High Time Before Read (See Note)	Min	200	ns

Note: *Not 100% tested.*

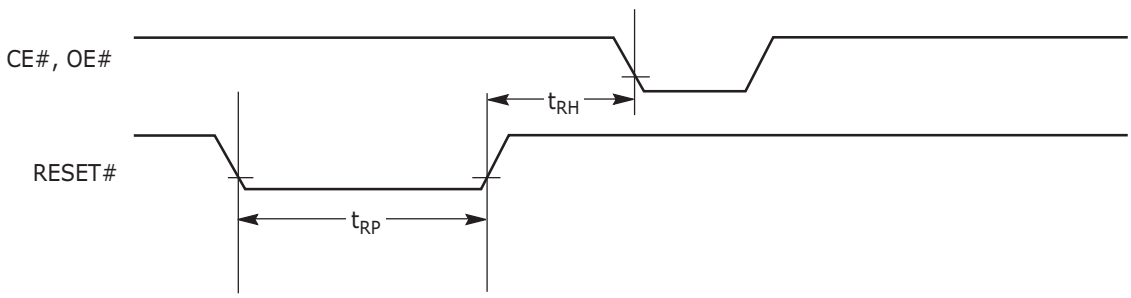


Figure 11.12. Reset Timings

11.8.6 Erase/Program Timing

Parameter		Description			54 MHz	66 MHz	80 MHz	Unit
JEDEC	Standard							
t_{AVAV}	t_{WC}	Write Cycle Time (Note 1)		Min	70			ns
t_{AVWL}	t_{AS}	Address Setup Time (Notes 2, 3)	Synchronous	Min	5			ns
			Asynchronous		0			ns
t_{WLAX}	t_{AH}	Address Hold Time (Notes 2, 3)	Synchronous	Min	9			ns
			Asynchronous		20			
	t_{AVDP}	AVD# Low Time		Min	8			ns
t_{DVWH}	t_{DS}	Data Setup Time		Min	45	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	30			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_{VID}	V_{ACC} Rise and Fall Time		Min	500			ns
	t_{VIDS}	V_{ACC} Setup Time (During Accelerated Programming)		Min	1			µs
	t_{VCS}	V_{CC} Setup Time		Min	50			µs
t_{ELWL}	t_{CS}	CE# Setup Time to WE#		Min	5			ns
	t_{AVSW}	AVD# Setup Time to WE#		Min	5			ns
	t_{AVHW}	AVD# Hold Time to WE#		Min	5			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_{SEA}	Sector Erase Accept Time-out		Max	50			µs
	t_{ESL}	Erase Suspend Latency		Max	20			µs
	t_{PSL}	Program Suspend Latency		Max	20			µs
	t_{ASP}	Toggle Time During Sector Protection		Typ	100			µs
	t_{PSP}	Toggle Time During Programming Within a Protected Sector		Typ	1			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 falling edge of WE#. In synchronous program operation timing, addresses are latched on the rising edge of CLK.*
4. *See the "Erase and Programming Performance" section for more information.*
5. *Does not include the preprogramming time.*

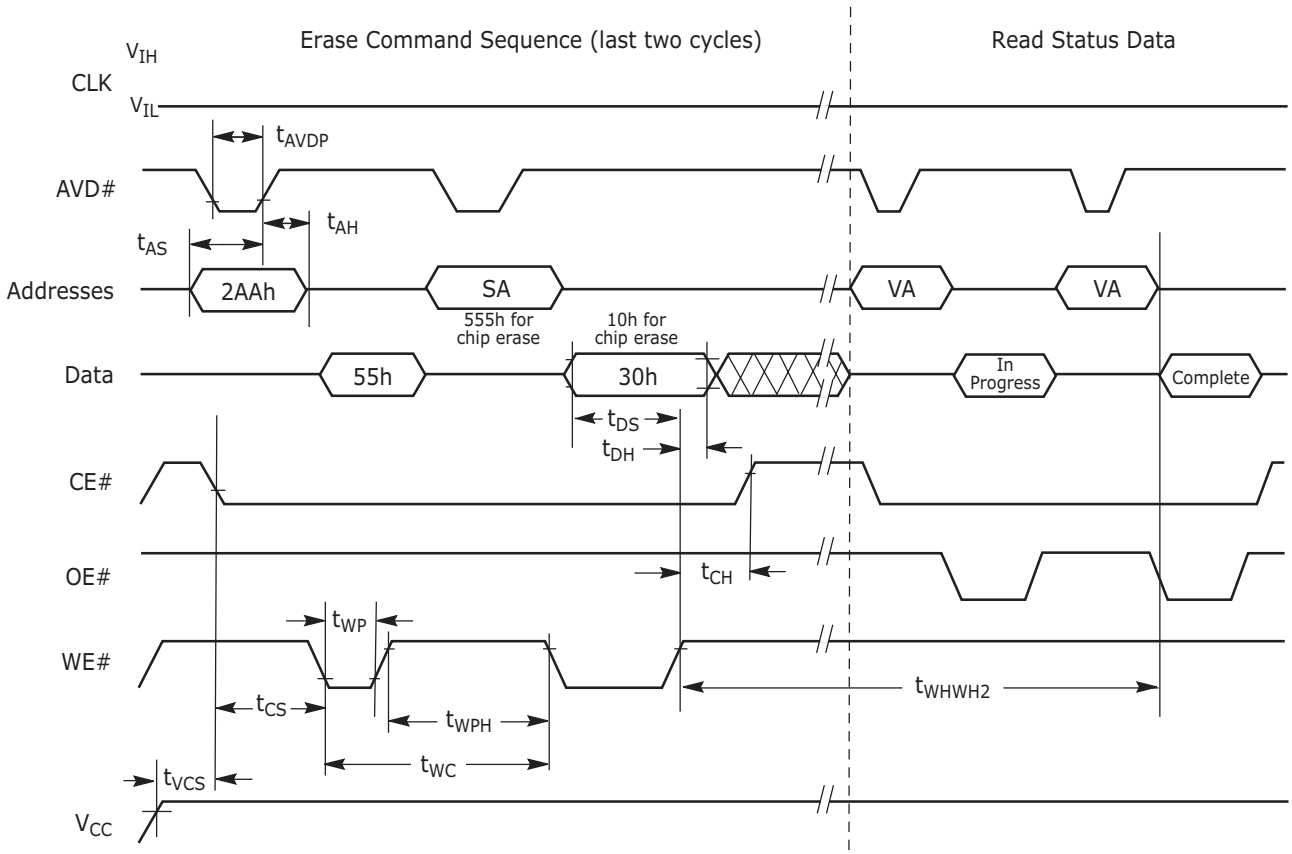
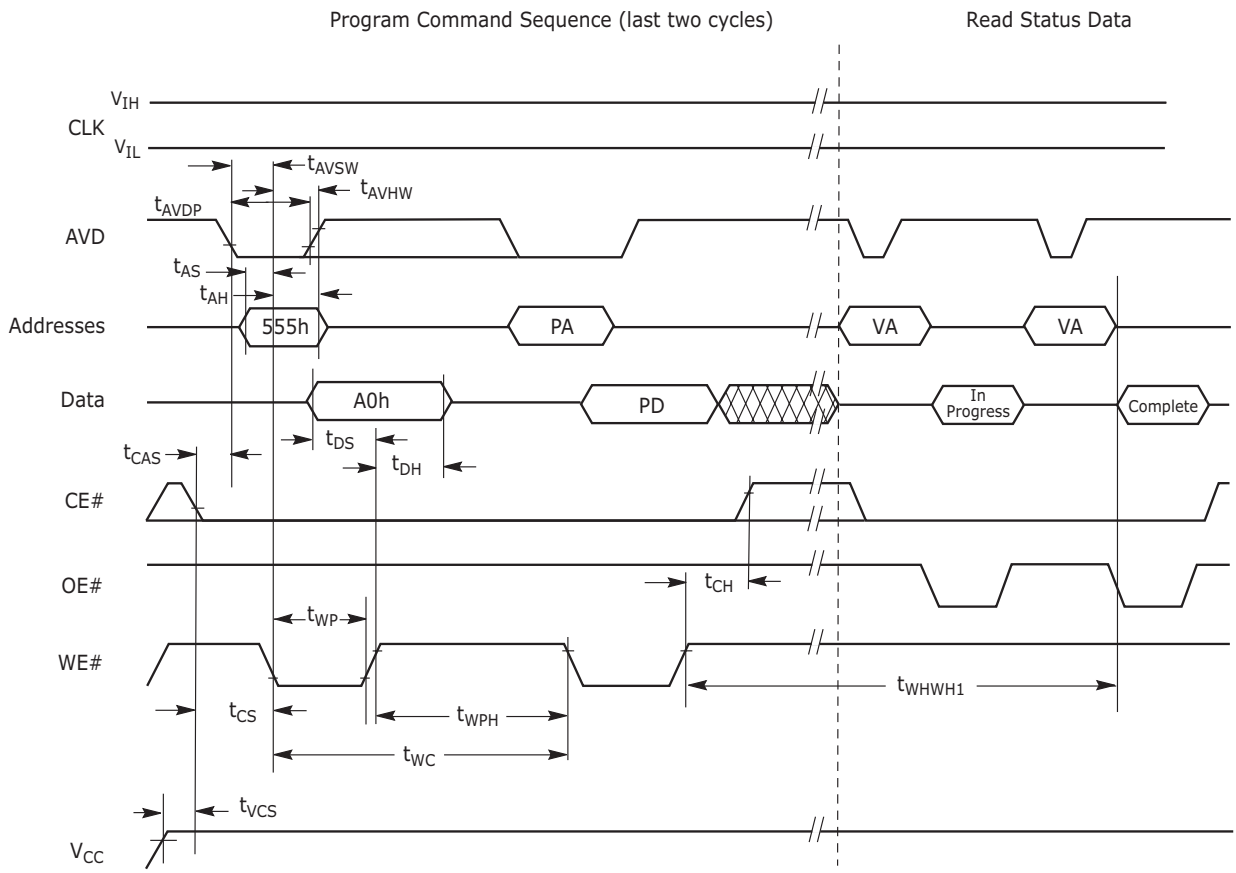


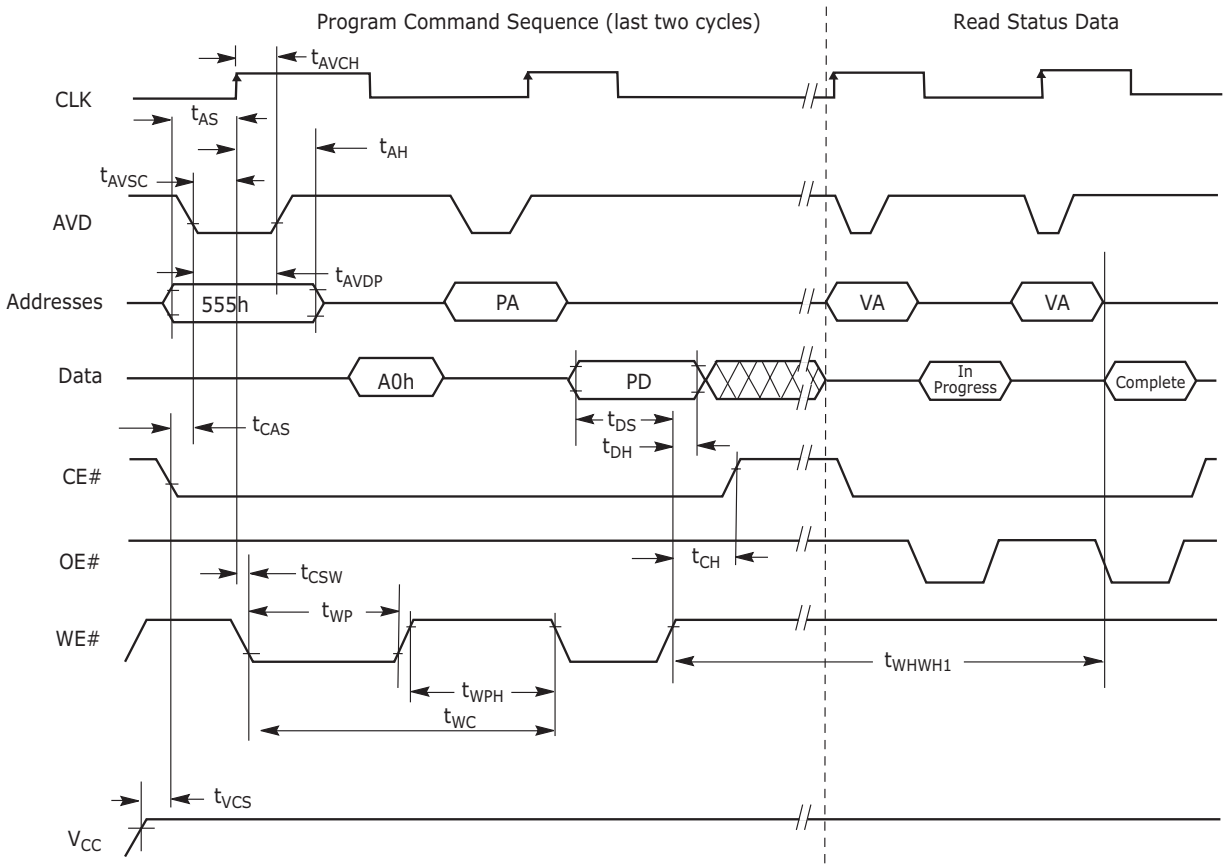
Figure 11.2. Chip/Sector Erase Operation Timings: WE# Latched Addresses



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. *A23–A14 for the WS256N (A22–A14 for the WS128N, A21–A14 for the WS064N) 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.*

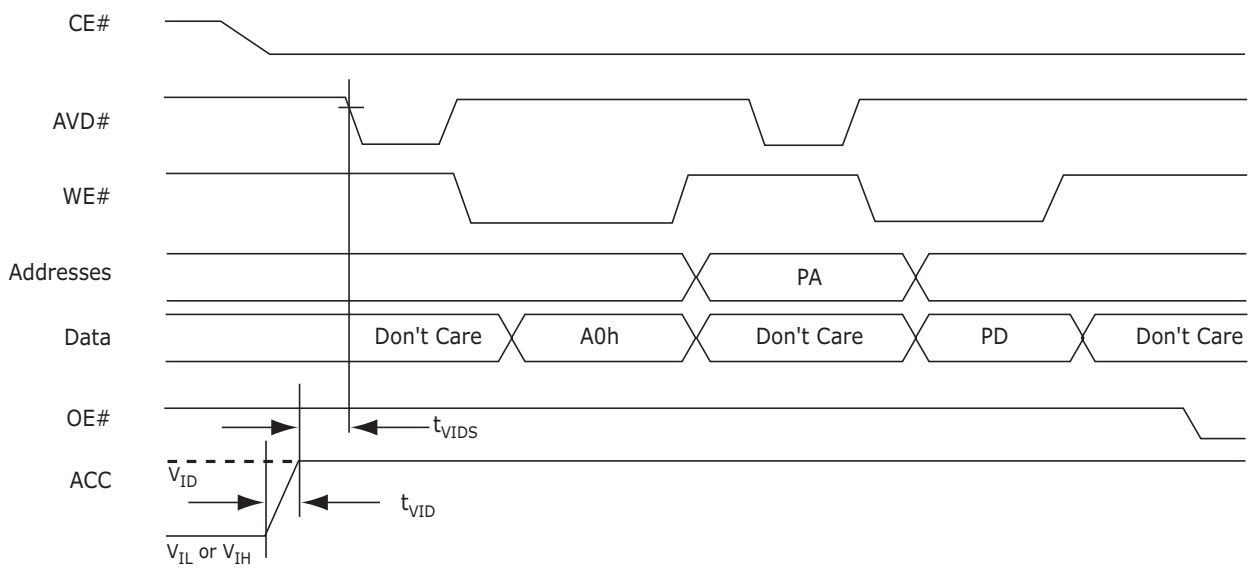
Figure 11.13. Asynchronous Program Operation Timings: WE# Latched Addresses



Notes:

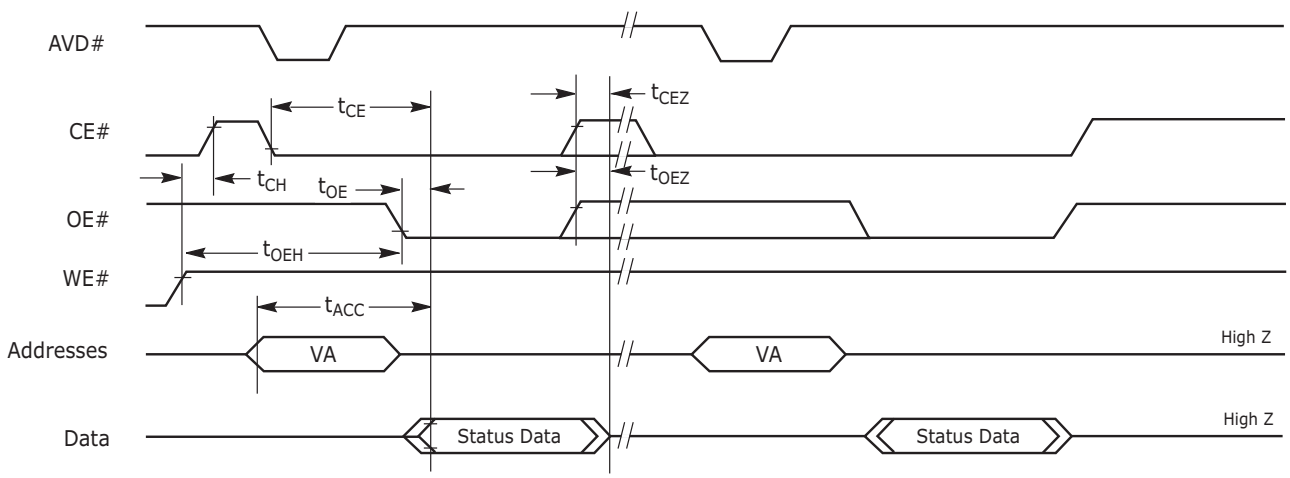
1. *PA = Program Address, PD = Program Data, VA = Valid Address for reading status bits.*
2. *"In progress" and "complete" refer to status of program operation.*
3. *A23–A14 for the WS256N (A22–A14 for the WS128N, A21–A14 for the WS064N) are don't care during command sequence unlock cycles.*
4. *Addresses 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 11.14. Synchronous Program Operation Timings: CLK Latched Addresses



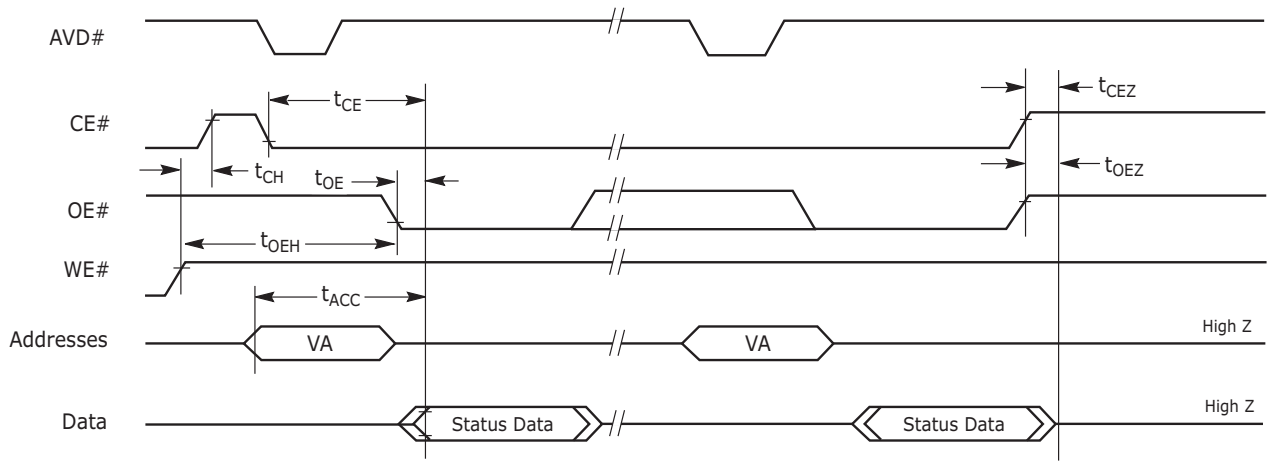
Note: *Use setup and hold times from conventional program operation.*

Figure 11.15. Accelerated Unlock Bypass Programming Timing



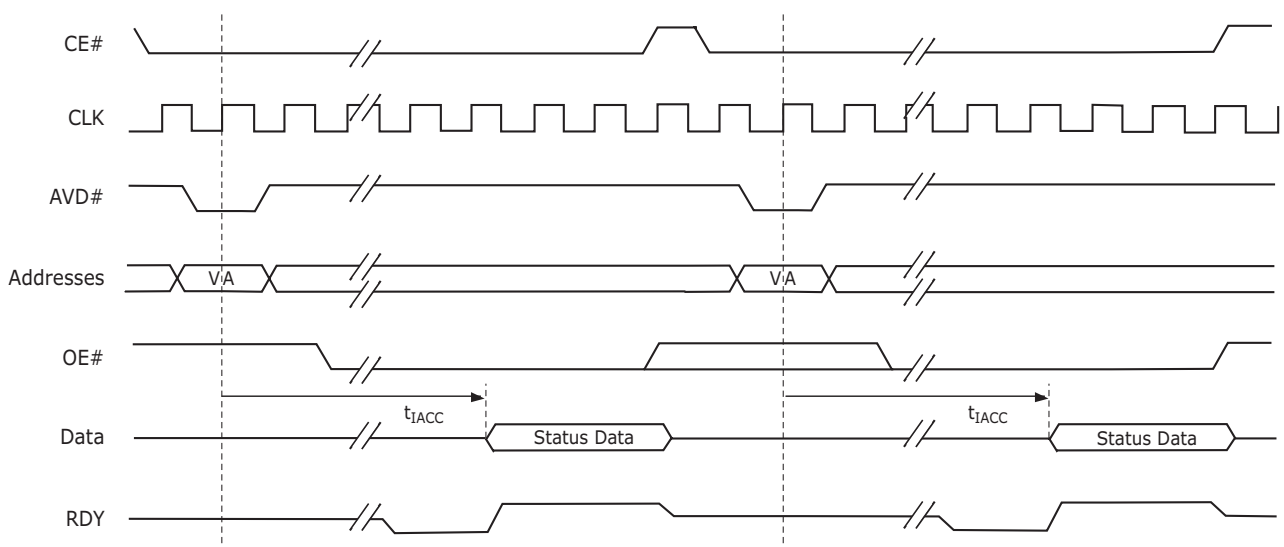
Notes:
1. *Status reads in figure are shown as asynchronous.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, and Data# Polling will output true data.*

Figure 11.16. Data# Polling Timings (During Embedded Algorithm)



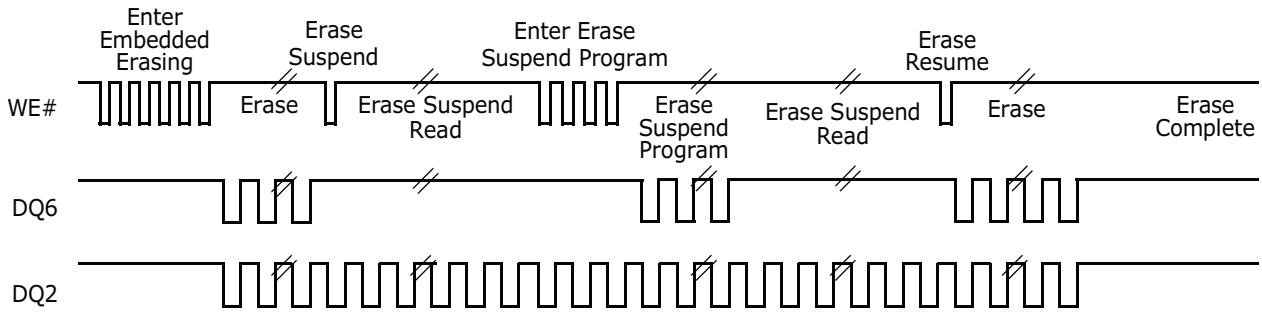
Notes:
1. *Status reads in figure are shown as asynchronous.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, the toggle bits will stop toggling.*

Figure 11.17. Toggle Bit Timings (During Embedded Algorithm)



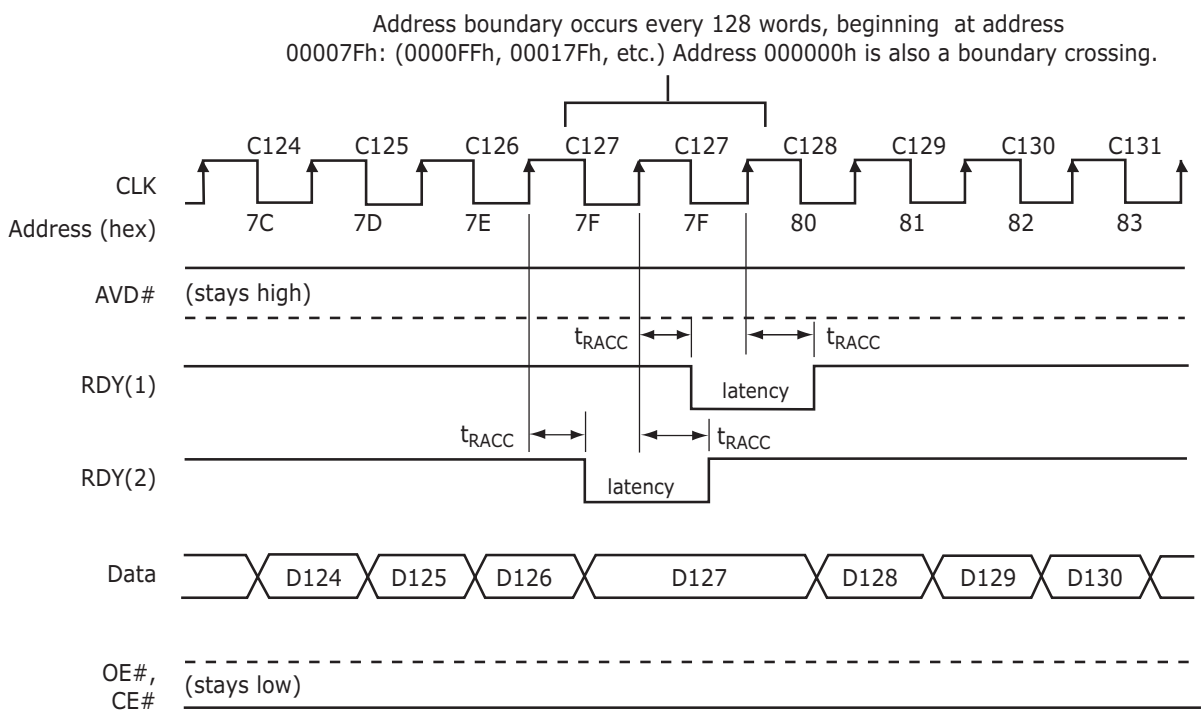
Notes:
1. *The timings are similar to synchronous read timings.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, the toggle bits will stop toggling.*
3. *RDY is active with data (D8 = 1 in the Configuration Register). When D8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

Figure 11.18. Synchronous Data Polling Timings/Toggle Bit Timings



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.*

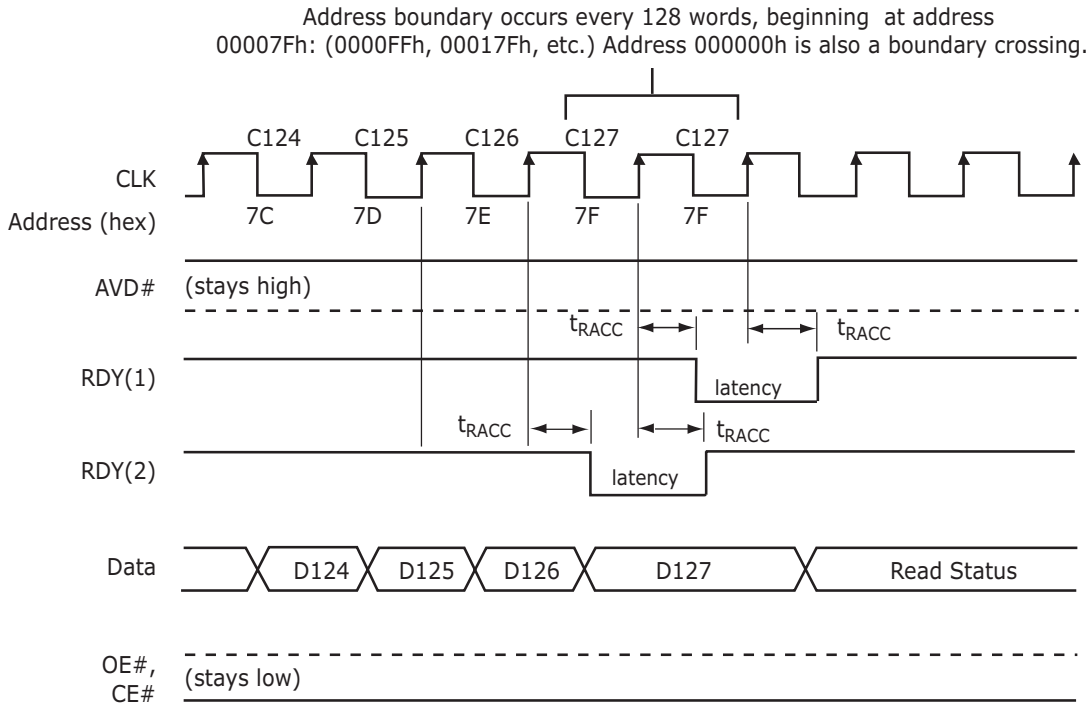
Figure 11.19. DQ2 vs. DQ6



Notes:

1. *RDY(1) active with data (D8 = 1 in the Configuration Register).*
2. *RDY(2) active one clock cycle before data (D8 = 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.*
5. *RDY will not go low and no additional wait states will be required if the Burst frequency is <=66 MHz and the Boundary Crossing bit (D14) in the Configuration Register is set to 0*

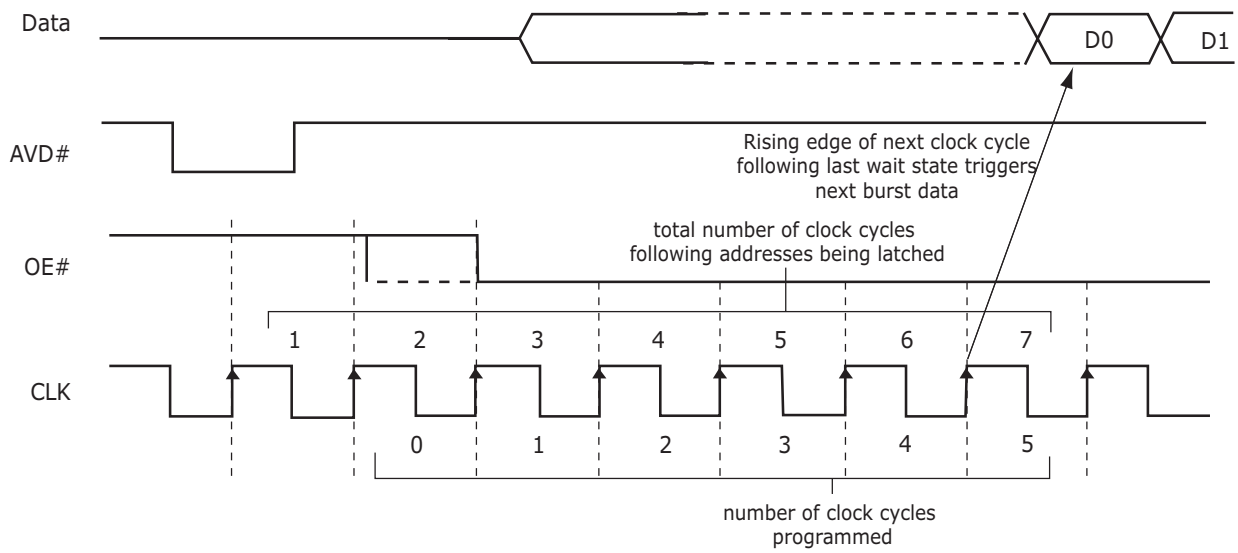
Figure 11.20. Latency with Boundary Crossing when Frequency > 66 MHz



Notes:

1. *RDY(1) active with data (D8 = 1 in the Configuration Register).*
2. *RDY(2) active one clock cycle before data (D8 = 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.*
5. *RDY will not go low and no additional wait states will be required if the Burst frequency is ≤ 66 MHz and the Boundary Crossing bit (D14) in the Configuration Register is set to 0.*

Figure 11.2I. Latency with Boundary Crossing into Program/Erase Bank

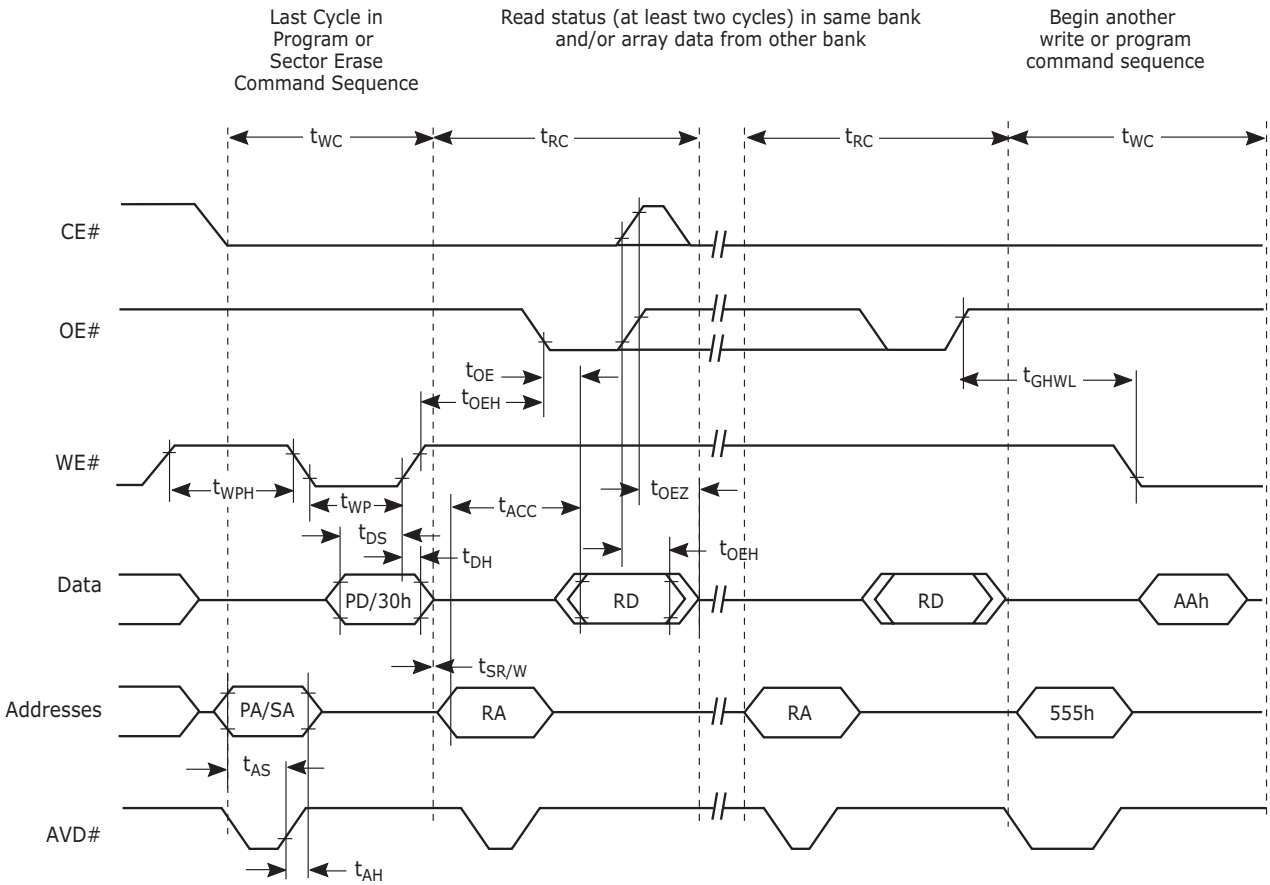


Wait State Configuration Register Setup:

D13, D12, D11 = "111" ⇒ Reserved
D13, D12, D11 = "110" ⇒ Reserved
D13, D12, D11 = "101" ⇒ 5 programmed, 7 total
D13, D12, D11 = "100" ⇒ 4 programmed, 6 total
D13, D12, D11 = "011" ⇒ 3 programmed, 5 total
D13, D12, D11 = "010" ⇒ 2 programmed, 4 total
D13, D12, D11 = "001" ⇒ 1 programmed, 3 total
D13, D12, D11 = "000" ⇒ 0 programmed, 2 total

Note: *Figure assumes address D0 is not at an address boundary, and wait state is set to "101".*

Figure 11.22. Example of Wait States Insertion



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.*

Figure 11.23. Back-to-Back Read/Write Cycle Timings

11.8.7 Erase and Programming Performance

Parameter			Typ (Note l)	Max (Note 2)	Unit	Comments
Sector Erase Time	64 Kword	V_{CC}	0.6	3.5	s	Excludes 00h programming prior to erasure (Note 4)
	16 Kword	V_{CC}	<0.15	2		
Chip Erase Time		V_{CC}	153.6 (WS256N) 77.4 (WS128N) 39.3 (WS064N)	308 (WS256N) 154 (WS128N) 78 (WS064N)	s	
		ACC	130.6 (WS256N) 65.8 (WS128N) 33.4 (WS064N)	262 (WS256N) 132 (WS128N) 66 (WS064N)		
Single Word Programming Time (Note 8)		V_{CC}	40	400	µs	
		ACC	24	240		
Effective Word Programming Time utilizing Program Write Buffer		V_{CC}	9.4	94	µs	
		ACC	6	60		
Total 32-Word Buffer Programming Time		V_{CC}	300	3000	µs	
		ACC	192	1920		
Chip Programming Time (Note 3)		V_{CC}	157.3 (WS256N) 78.6 (WS128N) 39.3 (WS064N)	314.6 (WS256N) 157.3 (WS128N) 78.6 (WS064N)	s	Excludes system level overhead (Note 5)
		ACC	100.7 (WS256N) 50.3 (WS128N) 25.2 (WS064N)	201.3 (WS256N) 100.7 (WS128N) 50.3 (WS064N)		

Notes:

1. *Typical program and erase times assume the following conditions: 25°C, 1.8 V V_{CC}, 10,000 cycles; checkerboard data pattern.*
2. *Under worst case conditions of 90°C, V_{CC} = 1.70 V, 100,000 cycles.*
3. *Typical chip programming time is considerably less than the maximum chip programming time listed, and is based on single word programming.*
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 the Appendix for further information on command definitions.*
6. *Contact the local sales office for minimum cycling endurance values in specific applications and operating conditions.*
7. *Refer to Application Note "Erase Suspend/Resume Timing" for more details.*
8. *Word programming specification is based upon a single word programming operation not utilizing the write buffer.*

11.8.8 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.*

I2 Appendix

This section contains information relating to software control or interfacing with the Flash device. For additional information and assistance regarding software, see the Additional Resources section on page 15, or explore the Web at www.amd.com and www.fujitsu.com.

Table 12.1. Memory Array Commands

Command Sequence (Notes)		Cycles	Bus Cycles (Notes 1–5)											
			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				
	Device ID (9)	6	555	AA	2AA	55	[BA]555	90	[BA]X01	227E	BA+X0E	Data	BA+X0F	2200
	Indicator Bits (10)	4	555	AA	2AA	55	[BA]555	90	[BA]X03	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 (18)		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	[BA]555	98										
Unlock Bypass Mode	Entry	3	555	AA	2AA	55	555	20						
	Program (16)	2	XXX	A0	PA	PD								
	CFI (16)	1	XXX	98										
	Reset	2	XXX	90	XXX	00								
SecSi Sector	Entry	3	555	AA	2AA	55	555	88						
	Program (17)	4	555	AA	2AA	55	555	A0	PA	PD				
	Read (17)	1	00	Data										
	Exit (17)	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. WS256N = A23–A14; WS128N = A22–A14; WS064N = A21–A14.

BA = Bank Address. WS256N = A23–A20; WS128N = A22–A20; WS064N = A21–A18.

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 6.4 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 Autoselect section for more information.*
9. *Data in cycle 5 is 2230 (WS256N), 2232 (WS064N), or 2231 (WS128N).*
10. *See Table 7.16 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. The number of cycles in the command sequence is 37 for full page programming (32 words). Less than 32 word programming is not recommended.*
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. Address will equal 55h on all future devices, but 555h for WS256N/128N/064N.*
16. *Requires Entry command sequence prior to execution. Unlock Bypass Reset command is required to return to reading array data.*
17. *Requires Entry command sequence prior to execution. SecSi Sector Exit Reset command is required to exit this mode; device may otherwise be placed in an unknown state.*
18. *Requires reset command to configure the Configuration Register.*

Table I2.2. Sector Protection Commands

Command Sequence (Notes)		Cycles	Bus Cycles (Notes 1–4)													
			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 (6)	2	XX	A0	77	data										
	Read (6)	1	77	data												
	Command Set Exit (7)	2	XX	90	XX	00										
Password Protection	Command Set Entry (5)	3	555	AA	2AA	55	555	60								
	Program [0-3] (8)	2	XX	A0	00	PWD[0-3]										
	Read (9)	4	0...00	PWD0	0...01	PWD1	0...02	PWD2	0...03	PWD3						
	Unlock	7	00	25	00	03	00	PWD0	01	PWD1	02	PWD2	03	PWD3	00	29
	Command Set Exit (7)	2	XX	90	XX	00										
Non-Volatile Sector Protection (PPB)	Command Set Entry (5)	3	555	AA	2AA	55	[BA]555	C0								
	PPB Program (10)	2	XX	A0	SA	00										
	All PPB Erase (10, 11)	2	XX	80	00	30										
	PPB Status Read	1	SA	RD(0)												
	Command Set Exit (7)	2	XX	90	XX	00										
Global Volatile Sector Protection Freeze (PPB Lock)	Command Set Entry (5)	3	555	AA	2AA	55	[BA]555	50								
	PPB Lock Bit Set	2	XX	A0	XX	00										
	PPB Lock Bit Status Read	1	BA	RD(0)												
	Command Set Exit (7)	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 (7)	2	XX	90	XX	00										

Legend:

X = Don't care.

RA = Address of the memory location to be read.

PD(0) = SecSi Sector Lock Bit. PD(0), or bit[0].

PD(1) = Persistent Protection Mode Lock Bit. PD(1), or bit[1], must be set to '0' for protection while PD(2), bit[2] must be left as '1'.

PD(2) = Password Protection Mode Lock Bit. PD(2), or bit[2], must be set to '0' for protection while PD(1), bit[1] must be left as '1'.

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

SA = Sector Address. WS256N = A23–A14; WS128N = A22–A14; WS064N = A21–A14.

BA = Bank Address. WS256N = A23–A20; WS128N = A22–A20; WS064N = A21–A18.

PWD3–PWD0 = Password Data. PD3–PD0 present four 16 bit combinations that represent the 64-bit Password

PWA = Password Address. Address bits A1 and A0 are used to select each 16-bit portion of the 64-bit entity.

PWD = Password Data.

RD(0), RD(1), RD(2) = DQ0, DQ1, or DQ2 protection indicator bit. If protected, DQ0, DQ1, or DQ2 = 0. If unprotected, DQ0, DQ1, DQ2 = 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. *If both the Persistent Protection Mode Locking Bit and the Password Protection Mode Locking Bit are set at the same time, the command operation will abort and return the device to the default Persistent Sector Protection Mode during 2nd bus cycle. Note that on all future devices, addresses will equal 00h, but are currently 77h for WS256N, WS128N, and WS064N. See Tables 8.1 and 8.2 for explanation of lock bits.*
7. *Exit command must be issued to reset the device into read mode; device may otherwise be placed in an unknown state.*
8. *Entire two bus-cycle sequence must be entered for each portion of the password.*
9. *Full address range is required for reading password.*
10. *See Figure 8.2 for details.*
11. *"All PPB Erase" command will pre-program all PPBs before erasure to prevent over-erasure.*

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 (BA)555h any time the device is ready to read array data. The system can read CFI information at the addresses given in Tables 12.3–12.6) within that bank. All reads outside of the CFI address range, within the bank, will return 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.amd.com and www.fujitsu.com) for general information on Spansion Flash memory software development guidelines.

```
/* Example: CFI Entry command */
  *( (UINT16 *)bank_addr + 0x555 ) = 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. 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 I2.4. 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)
1Eh	0000h	V_{PP} Max. voltage (00h = no V_{PP} pin present)
1Fh	0006h	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	0000h	Typical timeout for full chip erase 2^N ms (00h = not supported)
23h	0004h	Max. timeout for byte/word write 2^N times typical
24h	0004h	Max. timeout for buffer write 2^N times typical
25h	0003h	Max. timeout per individual block erase 2^N times typical
26h	0000h	Max. timeout for full chip erase 2^N times typical (00h = not supported)

Table I2.5. Device Geometry Definition

Addresses	Data	Description
27h	0019h (WS256N) 0018h (WS128N) 0017h (WS064N)	Device Size = 2^N byte
28h 29h	0001h 0000h	Flash Device Interface description (refer to CFI publication 100)
2Ah 2Bh	0006h 0000h	Max. number of bytes in multi-byte write = 2^N (00h = not supported)
2Ch	0003h	Number of Erase Block Regions within device
2Dh 2Eh 2Fh 30h	0003h 0000h 0080h 0000h	Erase Block Region 1 Information (refer to the CFI specification or CFI publication 100)
31h	00FDh (WS256N) 007Dh (WS128N) 003Dh (WS064N)	Erase Block Region 2 Information
32h 33h 34h	0000h 0000h 0002h	
35h 36h 37h 38h	0003h 0000h 0080h 0000h	Erase Block Region 3 Information
39h 3Ah 3Bh 3Ch	0000h 0000h 0000h 0000h	Erase Block Region 4 Information

Table I2.6. Primary Vendor-Specific Extended Query

Addresses	Data	Description
40h 41h 42h	0050h 0052h 0049h	Query-unique ASCII string "PRI"
43h	0031h	Major version number, ASCII
44h	0034h	Minor version number, ASCII
45h	0100h	Address Sensitive Unlock (Bits 1-0) 0 = Required, 1 = Not Required Silicon Technology (Bits 5-2) 0100 = 0.11 µm
46h	0002h	Erase Suspend 0 = Not Supported, 1 = To Read Only, 2 = To Read & 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	00F3h (WS256N) 006Fh (WS128N) 0037h (WS064N)	Simultaneous Operation Number of Sectors in all banks except boot bank
4Bh	0001h	Burst Mode Type 00 = Not Supported, 01 = Supported
4Ch	0000h	Page Mode Type 00 = Not Supported, 01 = 4 Word Page, 02 = 8 Word Page, 04 = 16 Word Page
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	0001h	Top/Bottom Boot Sector Flag 0001h = Dual Boot Device
50h	0001h	Program Suspend. 00h = not supported
51h	0001h	Unlock Bypass 00 = Not Supported, 01=Supported
52h	0007h	SecSi Sector (Customer OTP Area) Size 2^N bytes
53h	0014h	Hardware Reset Low Time-out during an embedded algorithm to read mode Maximum 2^N ns
54h	0014h	Hardware Reset Low Time-out not during an embedded algorithm to read 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	0010h	Bank Organization: X = Number of banks
58h	0013h (WS256N) 000Bh (WS128N) 0007h (WS064N)	Bank 0 Region Information. X = Number of sectors in bank

Table 12.6. Primary Vendor-Specific Extended Query (Continued)

Addresses	Data	Description
59h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 1 Region Information. X = Number of sectors in bank
5Ah	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 2 Region Information. X = Number of sectors in bank
5Bh	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 3 Region Information. X = Number of sectors in bank
5Ch	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 4 Region Information. X = Number of sectors in bank
5Dh	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 5 Region Information. X = Number of sectors in bank
5Eh	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 6 Region Information. X = Number of sectors in bank
5Fh	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 7 Region Information. X = Number of sectors in bank
60h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 8 Region Information. X = Number of sectors in bank
61h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 9 Region Information. X = Number of sectors in bank
62h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 10 Region Information. X = Number of sectors in bank
63h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 11 Region Information. X = Number of sectors in bank
64h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 12 Region Information. X = Number of sectors in bank
65h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 13 Region Information. X = Number of sectors in bank
66h	0010h (WS256N) 0008h (WS128N) 0004h (WS064N)	Bank 14 Region Information. X = Number of sectors in bank
67h	0013h (WS256N) 000Bh (WS128N) 0007h (WS064N)	Bank 15 Region Information. X = Number of sectors in bank

13 Commonly Used Terms

Term	Definition
ACC	ACCelerate. A special purpose input signal which allows for faster programming or erase operation when raised to a specified voltage above V_{CC}. In some devices ACC may protect all sectors when at a low voltage.
A_{max}	Most significant bit of the address input [A23 for 256Mbit, A22 for128Mbit, A21 for 64Mbit]
A_{min}	Least significant bit of the address input signals (A0 for all devices in this document).
Asynchronous	Operation where signal relationships are based only on propagation delays and are unrelated to synchronous control (clock) signal.
Autoselect	Read mode for obtaining manufacturer and device information as well as sector protection status.
Bank	Section of the memory array consisting of multiple consecutive sectors. A read operation in one bank, can be independent of a program or erase operation in a different bank for devices that offer simultaneous read and write feature.
Boot sector	Smaller size sectors located at the top and or bottom of Flash device address space. The smaller sector size allows for finer granularity control of erase and protection for code or parameters used to initiate system operation after power-on or reset.
Boundary	Location at the beginning or end of series of memory locations.
Burst Read	*See synchronous read*.
Byte	8 bits
CFI	Common Flash Interface. A Flash memory industry standard specification [JEDEC 137-A and JESD68.01] designed to allow a system to interrogate the Flash to determine its size, type and other performance parameters.
Clear	Zero (Logic Low Level)
Configuration Register	Special purpose register which must be programmed to enable synchronous read mode
Continuous Read	Synchronous method of burst read whereby the device will read continuously until it is stopped by the host, or it has reached the highest address of the memory array, after which the read address wraps around to the lowest memory array address
Erase	Returns bits of a Flash memory array to their default state of a logical One (High Level).
Erase Suspend/Erase Resume	Halts an erase operation to allow reading or programming in any sector that is not selected for erasure
BGA	Ball Grid Array package. Spansion LLC offers two variations: Fortified Ball Grid Array and Fine-pitch Ball Grid Array. See the specific package drawing or connection diagram for further details.
Linear Read	Synchronous (burst) read operation in which 8, 16, or 32 words of sequential data with or without wraparound before requiring a new initial address**.**
MCP	Multi-Chip Package. A method of combining integrated circuits in a single package by "stacking" multiple die of the same or different devices.
Memory Array	The programmable area of the product available for data storage.
MirrorBit™ Technology	Spansion™ trademarked technology for storing multiple bits of data in the same transistor.

Term	Definition
Page	Group of words that may be accessed more rapidly as a group than if the words were accessed individually.
Page Read	Asynchronous read operation of several words in which the first word of the group takes a longer initial access time and subsequent words in the group take less "page" access time to be read. Different words in the group are accessed by changing only the least significant address lines.
Password Protection	Sector protection method which uses a programmable password, in addition to the Persistent Protection method, for protection of sectors in the Flash memory device**.**
Persistent Protection	Sector protection method that uses commands and only the standard core voltage supply to control protection of sectors in the Flash memory device. This method replaces a prior technique of requiring a 12V supply to control the protection method.
Program	Stores data into a Flash memory by selectively clearing bits of the memory array in order to leave a data pattern of "ones" and "zeros".
Program Suspend/Program Resume	Halts a programming operation to read data from any location that is not selected for programming or erase.
Read	Host bus cycle that causes the Flash to output data onto the data bus.
Registers	Dynamic storage bits for holding device control information or tracking the status of an operation.
SecSi™	Secured Silicon. An area consisting of 256 bytes in which any word may be programmed once, and the entire area may be protected once from any future programming. Information in this area may be programmed at the factory or by the user. Once programmed and protected there is no way to change the secured information. This area is often used to store a software readable identification such as a serial number.
Sector Protection	Use of one or more control bits per sector to indicate whether each sector may be programmed or erased. If the Protection bit for a sector is set the embedded algorithms for program or erase will ignore program or erase commands related to that sector.
Sector	An Area of the memory array in which all bits must be erased together by an erase operation.
Simultaneous Operation	Mode of operation in which a host system may issue a program or erase command to one bank, that embedded algorithm operation may then proceed while the host immediately follows the embedded algorithm command with reading from another bank. Reading may continue concurrently in any bank other than the one executing the embedded algorithm operation.
Synchronous Operation	Operation that progresses only when a timing signal, known as a clock, transitions between logic levels (that is, at a clock edge).
VersatileIO™ (V_{IO})	Separate power supply or voltage reference signal that 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.
Unlock Bypass	Mode that facilitates faster program times by reducing the number of command bus cycles required to issue a write operation command. In this mode the initial two "Unlock" write cycles, of the usual 4 cycle Program command, are not required – reducing all Program commands to two bus cycles while in this mode.
Word	Two contiguous bytes (16 bits) located at an even byte boundary. A double word is two contiguous words located on a two word boundary. A quad word is four contiguous words located on a four word boundary.

Term	Definition
Wraparound	Special burst read mode where the read address "wraps" or returns back to the lowest address boundary in the selected range of words, after reading the last Byte or Word in the range, e.g. for a 4 word range of 0 to 3, a read beginning at word 2 would read words in the sequence 2, 3, 0, 1.
Write	Interchangeable term for a program/erase operation where the content of a register and or memory location is being altered. The term write is often associated with "writing command cycles" to enter or exit a particular mode of operation.
Write Buffer	Multi-word area in which multiple words may be programmed as a single operation**.** A Write Buffer may be 16 to 32 words long and is located on a 16 or 32 word boundary respectively.
Write Buffer Programming	Method of writing multiple words, up to the maximum size of the Write Buffer, in one operation. Using Write Buffer Programming will result in ≥ 8 times faster programming time than by using single word at a time programming commands.
Write Operation Status	Allows the host system to determine the status of a program or erase operation by reading several special purpose register bits**.**

I4 Revisions

Revision F (October 29, 2004)

Data sheet completely revised. Changed arrangement of sections; edited explanatory text, added flowcharts. This document supersedes Revision E+1, issued August 9, 2004; only the changes specified for Revision F in this section affect the document or device. All other device specifications remain the same as presented in Revision E+1.

Deleted product selector guide.

11.8.2, Synchronous/Burst Read

Deleted t_{AAS} and t_{AAH} from table. Modified Note 1.

Table 12.4, System Interface String

Changed data at address 23h from 0003h to 0004h.