THE DATASHEET OF

S29WS256P0PBFW000

www.win-source.net 0086-755-83957316

S29WS512P
S29WS256P
S29WS128P

SUPPLEMENT

512/256/128 Mb (32/16/8 M x 16 bit), 1.8 V, Simultaneous Read/Write Flash

Features

- Single 1.8 V read/program/erase (1.70–1.95 V)
- 90 nm MirrorBit™ Technology
- Simultaneous Read/Write operation with zero latency
- Random page read access mode of 8 words with 20 ns intra page access time
- 32 Word / 64 Byte Write Buffer
- Sixteen-bank architecture consisting of 32/16/8 Mwords for 512/256/128P, respectively
- Four 16 Kword sectors at both top and bottom of memory array
- 510/254/126 64Kword sectors (WS512/256/128P)
- Programmable linear (8/16/32) with or without wrap around and continuous burst read modes
- Secured Silicon Sector region consisting of 128 words each for factory and 128 words for customer
- 20-year data retention (typical)
- Cycling Endurance: 100,000 cycles per sector (typical)
- Command set compatible with JEDEC (42.4) standard
- Hardware (WP#) protection of top and bottom sectors
- Dual boot sector configuration (top and bottom)
- Handshaking by monitoring RDY
- Offered Packages
  - WS512P/WS256P/WS128P: 84-ball FBGA (11.6 mm x 8 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)

General Description

The Cypress S29WS512/256/128P are Mirrorbit® Flash products fabricated on 90 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. These products can operate up to 104 MHz and use a single V_{CC} of 1.7 V to 1.95 V that makes them ideal for today's demanding wireless applications requiring higher density, better performance and lowered power consumption.

Performance Characteristics

Read Access Times	
Speed Option (MHz)	**104**
Max. Synch Access Time (t_{IACC})	103.8
Max. Synch. Burst Access, ns (t_{BACC})	7.6
Max OE# Access Time, ns (t_{OE})	7.6
Max. Asynch. Access Time, ns (t_{ACC})	80

Current Consumption (typical values)	
Continuous Burst Read @ 104 MHz	36 mA
Simultaneous Operation 104 MHz	40 mA
Program	20 mA
Standby Mode	20 µA

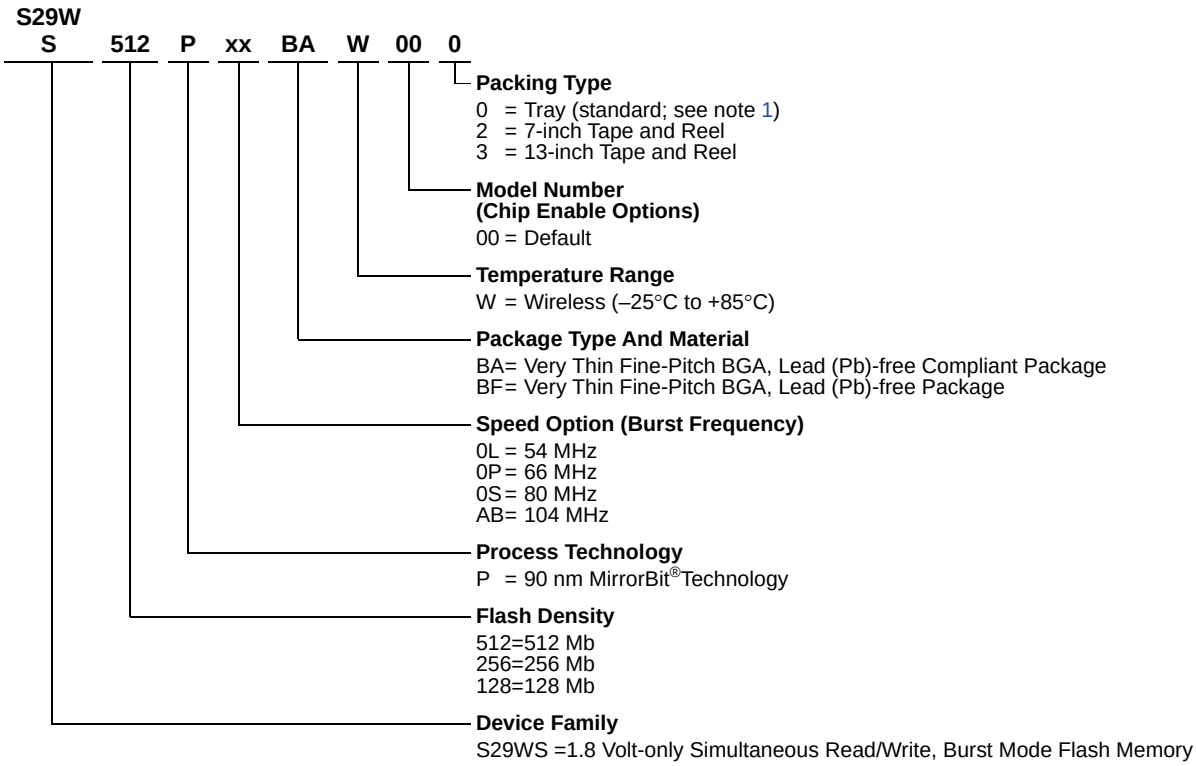
Typical Program & Erase Times	
Single Word Programming	40 µs
Effective Write Buffer Programming (V_{CC}) Per Word	9.4 µs
Effective Write Buffer Programming (V_{ACC}) Per Word	6 µs
Sector Erase (16 Kword Sector)	350 ms
Sector Erase (64 Kword Sector)	600 ms

Contents

1. Ordering Information 3
1.1 Valid Combinations 3
2. Input/Output Descriptions & Logic Symbol 4
3. Block Diagrams 5
4. Physical Dimensions/Connection Diagrams 6
4.1 Related Documents 6
4.2 Special Handling Instructions for FBGA Package 6
4.3 MCP Look-ahead Connection Diagram 7
5. Additional Resources 8
6. Product Overview 9
6.1 Memory Map 9
7. Device Operations 13
7.1 Device Operation Table 13
7.2 Asynchronous Read 14
7.3 Page Mode Read 14
7.4 Synchronous (Burst) Read Operation 15
7.5 Synchronous (Burst) Read Mode & Configuration Register 26
7.6 Autoselect 30
7.7 Program/Erase Operations 32
7.8 Simultaneous Read/Program or Erase 48
7.9 Writing Commands/Command Sequences 48
7.10 Handshaking 49
7.11 Hardware Reset 49
7.12 Software Reset 49
8. Advanced Sector Protection/Unprotection 51
8.1 Advanced Sector Protection Software Examples 51
8.2 Lock Register 52
8.3 Persistent Protection Bits 53
8.4 Dynamic Protection Bits 54
8.5 Persistent Protection Bit Lock Bit 54
8.6 Password Protection Method 54
8.7 Hardware Data Protection Methods 56
9. Power Conservation Modes 58
9.1 Standby Mode 58
9.2 Automatic Sleep Mode 58
9.3 Hardware RESET# Input Operation 58
9.4 Output Disable (OE#) 58
10. Secured Silicon Sector Flash Memory Region 59
10.1 Factory Secured Silicon Sector 59
10.2 Customer Secured Silicon Sector 60
10.3 Secured Silicon Sector Entry/Exit Command Sequences 60
11. Electrical Specifications 62
11.1 Absolute Maximum Ratings 62
11.2 Operating Ranges 62
11.3 DC Characteristics 63
11.4 Test Conditions 64
11.5 Key to Switching Waveforms 65
11.6 Switching Waveforms 65
11.7 Power-up/Initialization 65
11.8 CLK Characterization 66
11.9 AC Characteristics 67
11.10 Erase and Programming Performance 80
12. Appendix 81
12.1 Common Flash Memory Interface 85
13. Revision History 90

1. Ordering Information

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



1.1 Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult your local sales office to confirm availability of specific valid combinations and to check on newly released combinations.

S29WS512P Valid Combinations (Notes 1, 2)					Model Numbers	Package Type (Note 2)
Base Ordering Part Number	**Product Status**	**Speed Option**	**Package Type, Material, & Temperature Range**	**Packing Type**		
S29WS512P	Advance	0L, 0P, 0S, AB	BAW (Lead (Pb)-free Compliant), BFW (Lead (Pb)-free)	0, 2, 3 (Note 1)	00	11.6 mm x 8 mm 84-ball MCP-Compatible
S29WS256P						
S29WS128P						11.6 mm x 8 mm 84-ball MCP-Compatible

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.*

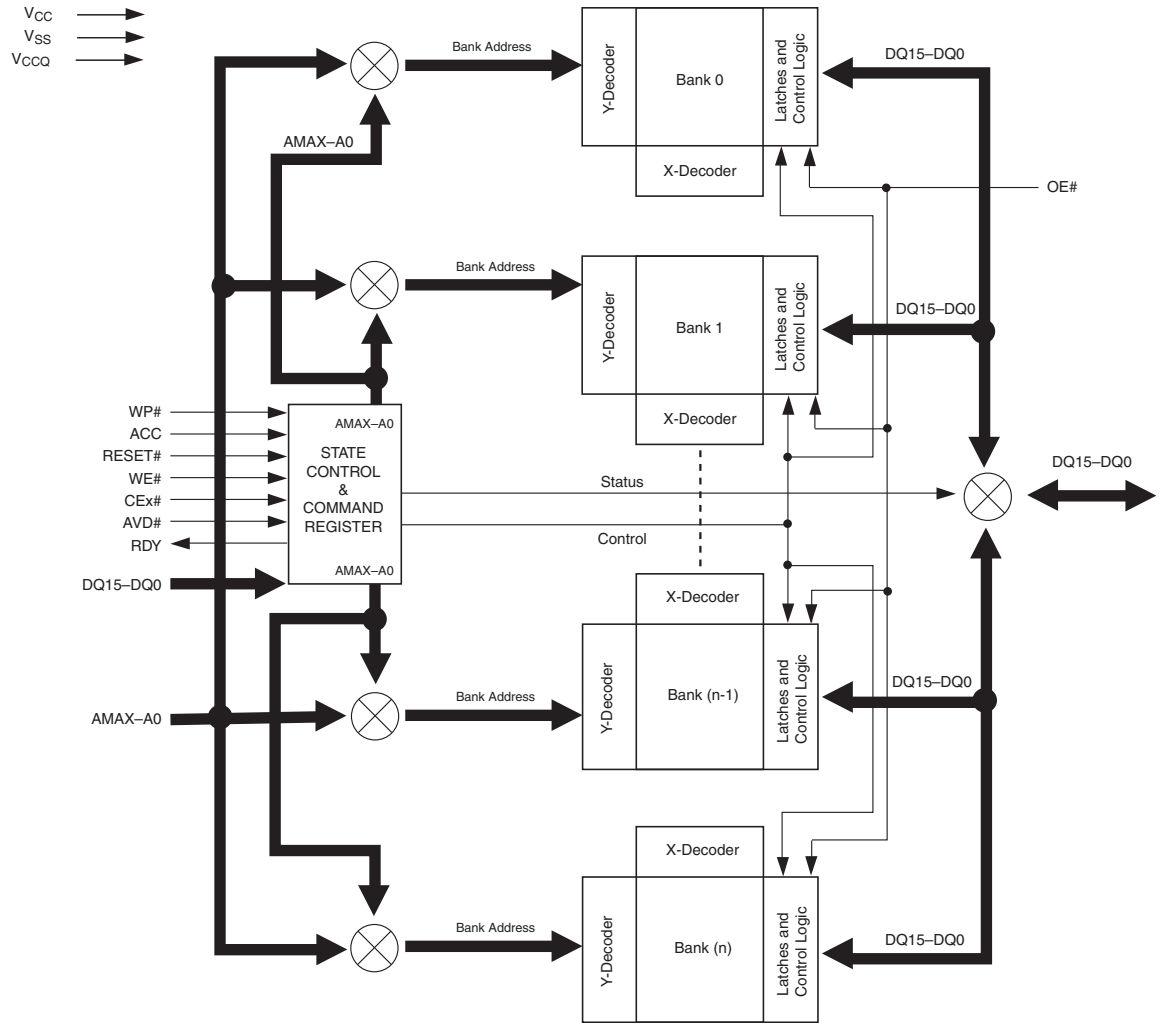
2. Input/Output Descriptions & Logic Symbol

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

Input/Output Descriptions

Symbol	Type	Description
A_{MAX}–A0	Input	Address lines (Amax = 24 for WS512P 1CE# option, 23 for WS512P 2CE# option, 23 for WS256P, and 22 for WS128P)
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_{CCQ}	Supply	Device Input/Output Power Supply (Must be ramped simultaneously with V_{CC})
V_{SS}	Supply	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 Diagrams



Notes:

1. *AMAX-A0 = A24-A0 for the WS512P, A23-A0 for the WS256P, and A22-A0 for the WS128P.*
2. *n = 15 for WS512P / WS256P / WS128P.*

4. Physical Dimensions/Connection Diagrams

This section shows the I/O designations and package specifications for the S29WS-P.

4.1 Related Documents

The following documents contain information relating to the S29WS-P devices. Click on the title or go to www.cypress.com to download the PDF file, or request a copy from your sales office.

- Considerations for X-ray Inspection of Surface-Mounted Flash Integrated Circuits

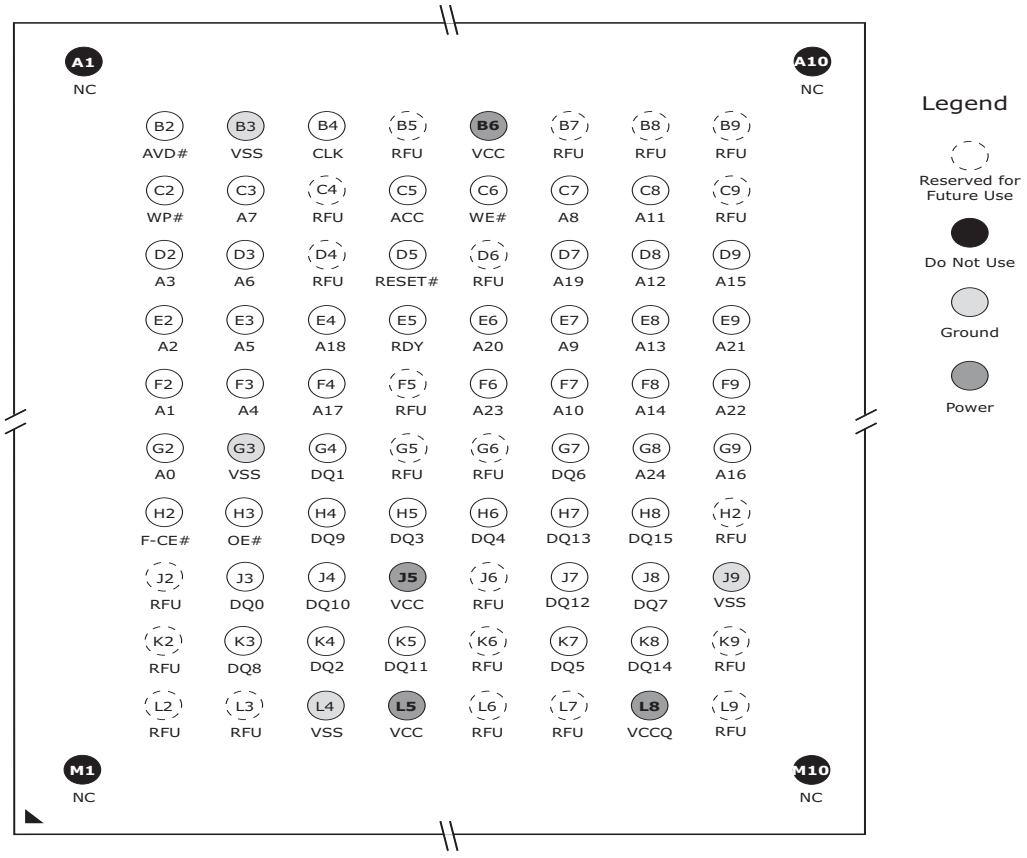
4.2 Special Handling Instructions for FBGA Package

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.

Figure 4.1 84-Ball Fine-Pitch Ball Grid Array, 512, 256 & 128 Mb

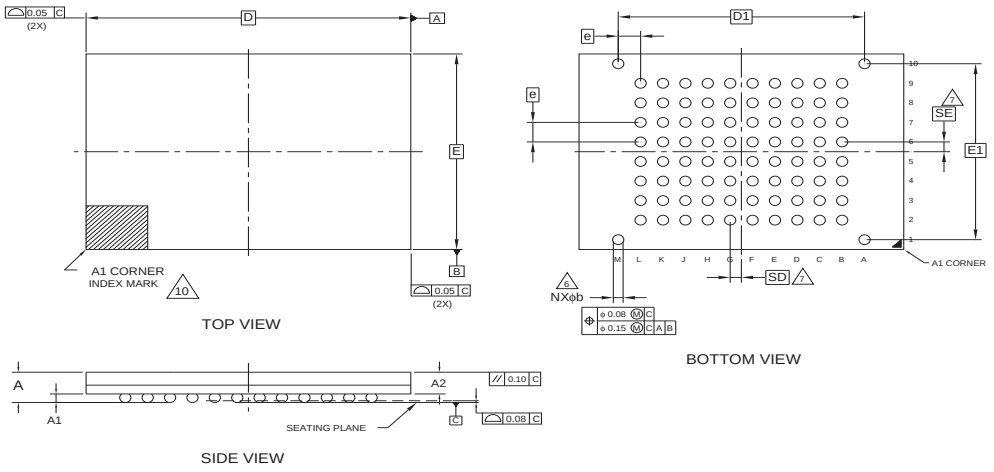
(Top View, Balls Facing Down, MCP Compatible)



Notes:

1. *Balls F6 and G8 are RFU on the WS128P.*
2. *Ball G8 is RFU on the WS256P.*
3. *V_{CC} pins must ramp simultaneously.*

Figure 4.2 VBH084—84-ball Fine-Pitch Ball Grid Array, 11.6 x 8 mm MCP Compatible Package



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

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. BALL POSITION DESIGNATION PER JESD 95-1, SPP-010 (EXCEPT AS NOTED).
4. e REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.
   N IS THE TOTAL NUMBER OF SOLDER BALLS.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = e/2
8. NOT USED.
9. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
10. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

3339 \ 16-038.25b

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

4.3 MCP Look-ahead Connection Diagram

Cypress Inc. 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 physical package outline may vary between connection diagrams and densities. The connection diagram for any MCP, however, is a subset of the pinout.

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 Cypress Products* application note (publication number: Design_Scalable_Wireless_AN), available on the web or through a Cypress sales office.

5. Additional Resources

Visit www.cypress.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 Cypress Products
- Common Flash Interface Version 1.4 Vendor Specific Extensions

Specification Bulletins

Contact your local sales office for details.

Drivers and Software Support

- Cypress low-level drivers
- True Flash File System

CAD Modeling Support

- VHDL and Verilog
- IBIS
- ORCAD® Schematic Symbols

Technical Support

Contact your local sales office or contact Cypress Inc. directly for additional technical support:
http://www.cypress.com/flash_memory_products/support/ses/index.html

6. Product Overview

The S29WS-P family consists of 512, 256, and 128 Mbit, 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 32, 16, or 8 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 area for storing customer and factory secured information. The Secured Silicon Sector is One Time Programmable.

6.1 Memory Map

The S29WS512/256/128P Mbit devices consist of 16 banks organized as shown in Tables –.

S29WS512P Sector & Memory Address Map

Bank Size	Sector Count	Sector Size (KB)	Bank	Sector/ Sector Range	Address Range	Notes
4 MB	4	32	0	SA000	000000h–003FFFh	Sector Starting Address – Sector Ending Address
		32		SA001	004000h–007FFFh	
		32		SA002	008000h–00BFFFh	
		32		SA003	00C000h–00FFFFh	
	31	128		SA004	010000h–01FFFFh	Sector Starting Address – Sector Ending Address (see note)
		⋮		⋮	⋮	
		128		SA034	1F0000h–1FFFFFh	
4 MB	32	128	1	SA035–SA066	200000h–3FFFFFh	First Sector, Starting Address – Last Sector, Ending Address (see note)
4 MB	32	128	2	SA067–SA098		
4 MB	32	128	3	SA099–SA130	⋮	
4 MB	32	128	4	SA131–SA162	⋮	
4 MB	32	128	5	SA163–SA194	⋮	
4 MB	32	128	6	SA195–SA226	⋮	
4 MB	32	128	7	SA227–SA258	E00000h–FFFFFFh	
4 MB	32	128	8	SA259–SA290	1000000-11FFFFF	
4 MB	32	128	9	SA291–SA322	⋮	
4 MB	32	128	10	SA323–SA354	⋮	
4 MB	32	128	11	SA355–SA386	⋮	
4 MB	32	128	12	SA387–SA418	⋮	
4 MB	32	128	13	SA419–SA450	⋮	
4 MB	32	128	14	SA451–SA482	1C00000h-1DFFFFFh	

S29WS512P Sector & Memory Address Map

Bank Size	Sector Count	Sector Size (KB)	Bank	Sector/ Sector Range	Address Range	Notes
4 MB	31	128	15	SA483	1E00000h-1E0FFFFh	Sector Starting Address – Sector Ending Address (see note)
				⋮		
				SA513	1FE0000h-1FEFFFFh	
	4	32		SA514	1FF0000h-1FF3FFFh	Sector Starting Address – Sector Ending Address
				SA515	1FF4000h-1FF7FFFh	
				SA516	1FF8000h-1FFBFFFh	
				SA517	1FFC000h-1FFFFFFh	

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–SA033) 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.

S29WS256P 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.

S29WS128P 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.

7. Device Operations

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

Operations are initiated by writing specific commands or a sequence with specific address and data patterns into the command registers (see Table on page 81 and Table on page 83). 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 describes the required state of each control pin for any particular operation.

Device Operations

Operation	CE#	OE#	WE#	CLK	AVD#	Amax–A0	DQ15–0	RDY	RESET#
Asynchronous Read - Addresses Latched	L	L	H	X	↑	Addr In	Output Valid	H	H
Asynchronous Read AVD# Steady State	L	L	H	X	L	Addr In	Output Valid	H	H
Asynchronous Write	L	H	↑	X	L	Addr In	Input Valid	H	H
Synchronous Write	L	H	L	↑	↑	Addr In	I/O	H	H
Standby (CE#)	H	X	X	X	X	X	HIGH Z	HIGH Z	H
Hardware Reset	X	X	X	X	X	X	HIGH Z	HIGH Z	↓
Burst Read Operations									
Latch Starting Burst Address by CLK	L	X	H	↑	L	Addr In	Output Invalid	X	H
Advance Burst read to next address	L	L	H	↑	H	X	Output Valid	H	H
Terminate current Burst read cycle	H	X	H	X	X	X	HIGH Z	HIGH Z	H
Terminate current Burst read cycle via RESET#	X	X	H	X	X	X	HIGH Z	HIGH Z	L
Terminate current Burst read cycle and start new Burst read cycle	L	X	H	↑	↓	Addr In	Output Invalid	X	H

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

Note:
Address is latched on the rising edge of clock.

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# must remain at V_{IH}. The rising edge of AVD# latches the address, preventing changes to the address lines from effecting the address being accessed.. Data is output on DQ15-DQ0 pins after the access time (t_{ACC}) has elapsed from the falling edge of AVD#, or the last time the address lines changed while AVD# was low.

7.3 Page Mode Read

The device is capable of fast page mode read. This mode provides fast (t_{PACC}) random read access speed for locations within a page. Address bits Amax–A3 select an 8 word page, and address bits A2–A0 select a specific word within that page. This is an asynchronous operation with the microprocessor supplying the specific word location. It does not matter if AVD# stays low or toggles. However, the address input must be always valid and stable if AVD# is low during the page read.

The random or initial page access is t_{ACC} or t_{CE} (depending on how the device was accessed) and subsequent page read accesses (as long as the locations specified by the microprocessor falls within that page) is equivalent to t_{PACC}. When CE# is deasserted (=V_{IH}), the reassertion of CE# for subsequent access has access time of t_{CE}. Here again, CE# selects the device and OE# is the output control and should be used to gate data to the output inputs if the device is selected. Fast page mode accesses are obtained by keeping Amax–A3 constant and changing A2–A0 to select the specific word within that page.

Page Select

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

7.4 Synchronous (Burst) Read Operation

The device is capable of continuous sequential burst operation and linear burst operation of a preset length. When the device first powers up, it is enabled for asynchronous read operations and can be automatically enabled for burst mode. To enter into synchronous mode, the configuration register will need to be set.

Prior to entering burst mode, the system should determine how many wait states are desired for the initial word (t_{IACC}) of each burst access, what mode of burst operation is desired and how the RDY signal will transition with valid data. The system would then write the configuration register command sequence.

Once the system has written the *Set Configuration Register* command sequence, the device is enabled for synchronous reads only.

The data is output t_{IACC} after the **rising edge** of the first CLK. Subsequent words are output t_{BACC} after the rising edge of each successive clock cycle, which automatically increments the internal address counter. Note that data is output only at the rising edge of the clock. RDY indicates the initial latency.

7.4.1 Latency Tables for Variable Wait State

The following tables show the latency for variable wait state in a continuous Burst operation

Address Latency for 11 Wait States

Word	Initial Wait																	
0	11 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
1		D1	D2	D3	D4	D5	D6	D7	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
2		D2	D3	D4	D5	D6	D7	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
3		D3	D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
4		D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
5		D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**2 ws**	D0

Address Latency for 10 Wait States

Word	Initial Wait																	
0	10 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
1		D1	D2	D3	D4	D5	D6	D7	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
2		D2	D3	D4	D5	D6	D7	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
3		D3	D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
4		D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
5		D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	**1 ws**	D0

Address Latency for 09 Wait States

Word	Initial Wait																	
0	9 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	…	D124	D125	D126	D127	D0
1		D1	D2	D3	D4	D5	D6	D7	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0
2		D2	D3	D4	D5	D6	D7	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0
3		D3	D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0
4		D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0
5		D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	…	D124	D125	D126	D127	D0

Address Latency for 8 Wait States

Word	Initial Wait									
0	8 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	**1 ws**	D8	D9
3		D3	D4	D5	D6	D7	**1 ws**	**1 ws**	D8	D9
4		D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	D8	D9
5		D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9

Address Latency for 7 Wait States

Word	Initial Wait									
0	7 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	D8	D9	D10
3		D3	D4	D5	D6	D7	1 ws	D8	D9	D10
4		D4	D5	D6	D7	**1 ws**	**1 ws**	D8	D9	D10
5		D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10

Address Latency for 6 Wait States

Word	Initial Wait									
0	6 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	D8	D9	D10
3		D3	D4	D5	D6	D7	D8	D9	D10	D11
4		D4	D5	D6	D7	**1 ws**	D8	D9	D10	D11
5		D5	D6	D7	**1 ws**	**1 ws**	D8	D9	D10	D11
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	D11
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	D11

Address Latency for 5 Wait States

Word	Initial Wait									
0	5 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	D8	D9	D10
3		D3	D4	D5	D6	D7	D8	D9	D10	D11
4		D4	D5	D6	D7	D8	D9	D10	D11	D12
5		D5	D6	D7	**1 ws**	D8	D9	D10	D11	D12
6		D6	D7	**1 ws**	**1 ws**	D8	D9	D10	D11	D12
7		D7	**1 ws**	**1 ws**	**1 ws**	D8	D9	D10	D11	D12

Address Latency for 4 Wait States

Word	Initial Wait									
0	4 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	D8	D9	D10
3		D3	D4	D5	D6	D7	D8	D9	D10	D11
4		D4	D5	D6	D7	D8	D9	D10	D11	D12
5		D5	D6	D7	D8	D9	D10	D11	D12	D13
6		D6	D7	**1 ws**	D8	D9	D10	D11	D12	D13
7		D7	**1 ws**	**1 ws**	D8	D9	D10	D11	D12	D13

Address Latency for 3 Wait States

Word	Initial Wait									
0	3 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	D8	D9	D10
3		D3	D4	D5	D6	D7	D8	D9	D10	D11
4		D4	D5	D6	D7	D8	D9	D10	D11	D12
5		D5	D6	D7	D8	D9	D10	D11	D12	D13
6		D6	D7	D8	D9	D10	D11	D12	D13	D14
7		D7	**1 ws**	D8	D9	D10	D11	D12	D13	D14

7.4.2 Latency for Boundary Crossing during First Read

The following tables show the latency at End of Word Line for boundary corssing during First Read in continuous burst operation

Address Latency for 11 Wait States

Word	Initial Wait										
0	11 ws	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
1		D121	D122	D123	D124	D125	D126	D127	**1 ws**	**2 ws**	D0
2		D122	D123	D124	D125	D126	D127	**1 ws**	**1 ws**	**2 ws**	D0
3		D123	D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**2 ws**	D0
4		D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**2 ws**	D0
5		D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**2 ws**	D0
6		D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**2 ws**	D0
7		D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**2 ws**	D0

Address Latency for 10 Wait States

Word	Initial Wait										
0	10 ws	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
1		D121	D122	D123	D124	D125	D126	D127	**1 ws**	**1 ws**	D0
2		D122	D123	D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	D0
3		D123	D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
4		D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
5		D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
6		D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
7		D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0

Address Latency for 9 Wait States

Word	Initial Wait									
0	9 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
2		D122	D123	D124	D125	D126	D127	**1 ws**	**1 ws**	D0
3		D123	D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	D0
4		D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
5		D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
6		D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0
7		D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0

Address Latency for 8 Wait States

Word	Initial Wait									
0	8 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	D0	D1
2		D122	D123	D124	D125	D126	D127	**1 ws**	D0	D1
3		D123	D124	D125	D126	D127	**1 ws**	**1 ws**	D0	D1
4		D124	D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	D0	D1
5		D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0	D1
6		D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0	D1
7		D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0	D1

Address Latency for 7 Wait States

Word	Initial Wait									
0	7 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	D0	D1
2		D122	D123	D124	D125	D126	D127	D0	D1	D2
3		D123	D124	D125	D126	D127	**1 ws**	D0	D1	D2
4		D124	D125	D126	D127	**1 ws**	**1 ws**	D0	D1	D2
5		D125	D126	D127	**1 ws**	**1 ws**	**1 ws**	D0	D1	D2
6		D126	D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0	D1	D2
7		D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0	D1	D2

Address Latency for 6 Wait States

Word	Initial Wait									
0	6 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	D0	D1
2		D122	D123	D124	D125	D126	D127	D0	D1	D2
3		D123	D124	D125	D126	D127	D0	D1	D2	D3
4		D124	D125	D126	D127	**1 ws**	D0	D1	D2	D3
5		D125	D126	D127	**1 ws**	**1 ws**	D0	D1	D2	D3
6		D126	D127	**1 ws**	**1 ws**	**1 ws**	D0	D1	D2	D3
7		D127	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D0	D1	D2	D3

Address Latency for 5 Wait States

Word	Initial Wait									
0	5 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	D0	D1
2		D122	D123	D124	D125	D126	D127	D0	D1	D2
3		D123	D124	D125	D126	D127	D0	D1	D2	D3
4		D124	D125	D126	D127	D0	D1	D2	D3	D4
5		D125	D126	D127	**1 ws**	D0	D1	D2	D3	D4
6		D126	D127	**1 ws**	**1 ws**	D0	D1	D2	D3	D4
7		D127	**1 ws**	**1 ws**	**1 ws**	D0	D1	D2	D3	D4

Address Latency for 4 Wait States

Word	Initial Wait									
0	4 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	D0	D1
2		D122	D123	D124	D125	D126	D127	D0	D1	D2
3		D123	D124	D125	D126	D127	D0	D1	D2	D3
4		D124	D125	D126	D127	D0	D1	D2	D3	D4
5		D125	D126	D127	D0	D1	D2	D3	D12	D5
6		D126	D127	**1 ws**	D0	D1	D2	D3	D12	D5
7		D127	**1 ws**	**1 ws**	D0	D1	D2	D3	D12	D5

Address Latency for 3 Wait States

Word	Initial Wait									
0	3 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D121	D122	D123	D124	D125	D126	D127	D0	D1
2		D122	D123	D124	D125	D126	D127	D0	D1	D2
3		D123	D124	D125	D126	D127	D0	D1	D2	D3
4		D124	D125	D126	D127	D0	D1	D2	D3	D4
5		D125	D126	D127	D0	D1	D2	D3	D4	D5
6		D126	D127	D0	D1	D2	D3	D4	D5	D6
7		D127	**1 ws**	D0	D1	D2	D3	D4	D5	D6

7.4.3 Latency at End of Word Line for Boundary Crossing After Second Read in Continuous Burst Operation

The following tables show the latency for boundary crossing after Second Read in a continuous Burst operation.

Address Latency for 11 Wait States

Word	Initial Wait																		
0	11 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
1		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
2		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
3		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
4		D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
5		D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
6		D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0
7		D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**2 ws**	D0

Address Latency for 10 Wait States

Word	Initial Wait																		
0	10 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
1		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
2		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
3		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
4		D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
5		D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
6		D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0
7		D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	**1 ws**	D0

Address Latency for 9 Wait States

Word	Initial Wait																	
0	9 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0

Address Latency for 8 Wait States

Word	Initial Wait																	
0	8 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0

Address Latency for 7 Wait States

Word	Initial Wait																	
0	7 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0

Address Latency for 6 Wait States

Word	Initial Wait																	
0	6 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0

Address Latency for 5 Wait States

Word	Initial Wait																	
0	5 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D113	D114	D115	D116	D117	D118	D119	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D114	D115	D116	D117	D118	D119	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D115	D116	D117	D118	D119	**1 ws**	**1 ws**	**1 ws**	D120	D121	D122	D123	D124	D125	D126	D127	D0

Address Latency for 4 Wait States

Word	Initial Wait																	
0	4 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D113	D114	D115	D116	D117	D118	D119	1 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D114	D115	D116	D117	D118	D119	1 ws	1 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0

Address Latency for 3 Wait States

Word	Initial Wait																	
0	4 ws	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
1		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
2		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
3		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
4		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
5		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
6		D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D0
7		D113	D114	D115	D116	D117	D118	D119	1 ws	D120	D121	D122	D123	D124	D125	D126	D127	D0

7.5 Synchronous (Burst) Read Mode & Configuration Register

See *Configuration Registers on page 28*, and Table , *Memory Array Commands on page 81*, for further details.

Figure 7.1 Synchronous/Asynchronous State Diagram

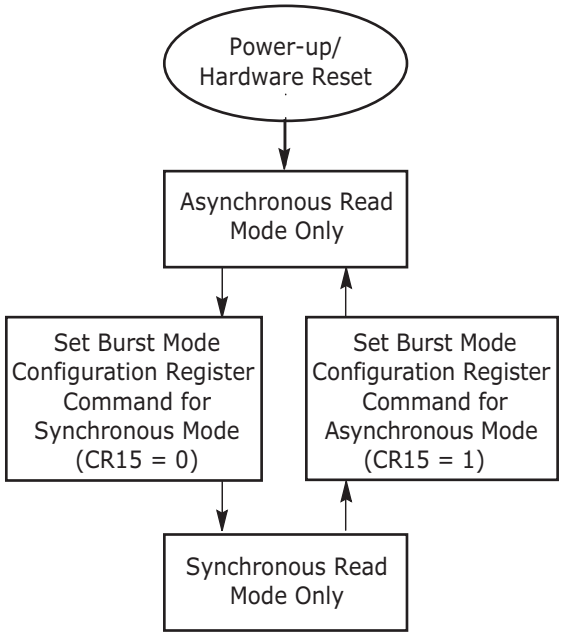
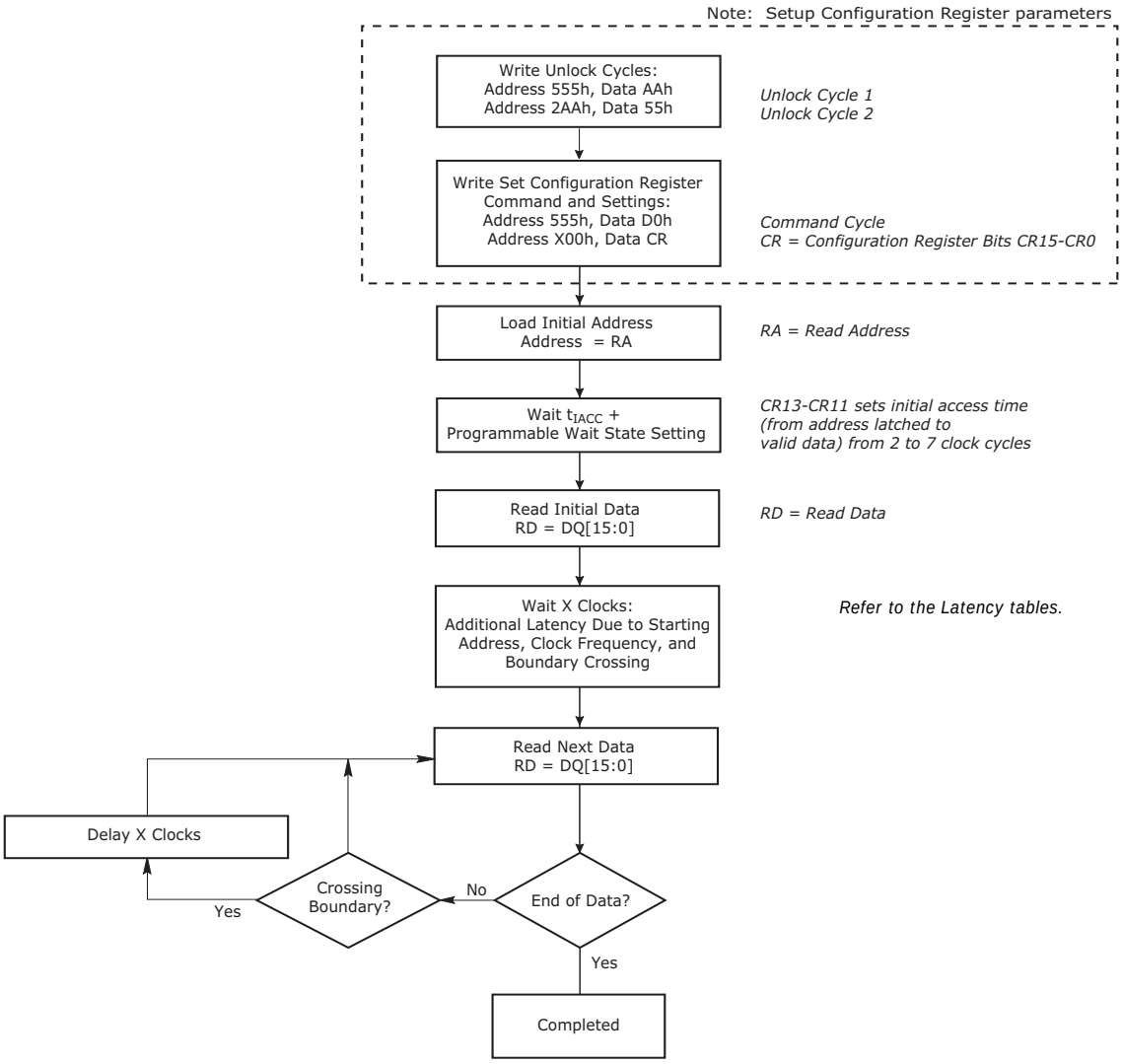


Figure 7.2 Synchronous Read Flow Chart



7.5.1 Continuous Burst Read Mode

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

If the address being read crosses a 128-word line boundary with in the same bank, but not into a program or erase suspended sector (as mentioned above), additional latency cycles are required as reflected by the configuration register table (Table).

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

7.5.2 8-, 16-, 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).

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 provided to the device, then wrap back to the first address in the selected address group.

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

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,...

7.5.3 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 executes up to the maximum memory address of the selected number of words. The burst stops after 8, 16, or 32 addresses and does 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 3Ch-43h, 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 requires 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, which will incur the additional boundary crossing wait state.

7.5.4 Configuration Registers

This device uses two 16-bit configuration registers to set various operational parameters. 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 can also be read using a command sequence (see Table on page 81). The following list describes the register settings.

Configuration Register

CR Bit	Function	Settings (Binary)
CR0.15	Set Device Read Mode	0 = Synchronous Read (Burst Mode) Enabled 1 = Asynchronous Mode (Default)
CR0.14	Reserved (Not used)	0 = Reserved 1 = Reserved (Default)
CR1.0	Programmable Wait State	0000 = initial data is valid on the 2nd rising CLK edge after addresses are latched 0001 = initial data is valid on the 3rd rising CLK edge after addresses are latched 0010 = initial data is valid on the 4th rising CLK edge after addresses are latched 0011 = initial data is valid on the 5th rising CLK edge after addresses are latched 0100 = initial data is valid on the 6th rising CLK edge after addresses are latched 0101 = initial data is valid on the 7th rising CLK edge after addresses are latched 0110 = Reserved 0111 = Reserved 1000 = initial data is valid on the 8th rising CLK edge after addresses are latched 1001 = initial data is valid on the 9th rising CLK edge after addresses are latched 1010 = initial data is valid on the 10th rising CLK edge after addresses are latched 1011 = initial data is valid on the 11th rising CLK edge after addresses are latched 1100 = Reserved 1101 = default 1110 = Reserved 1111 = Reserved
CR0.13		
CR0.12		
CR0.11		
CR0.10	RDY Polarity	0 = RDY signal is active low 1 = RDY signal is active high (Default)
CR0.9	Reserved (Not used)	0 = Reserved 1 = Reserved (Default)
CR0.8	RDY	0 = RDY active one clock cycle before data 1 = RDY active with data (Default)
CR0.7	Reserved (Not used)	0 = Reserved 1 = Reserved (Default)
CR0.6	Reserved	0 = Reserved 1 = Reserved (Default)
CR0.5	Reserved	0 = Reserved (Default) 1 = Reserved
CR0.4	RDY Function	0 = RDY (Default) 1 = Reserved
CR0.3	Burst Wrap Around	0 = No Wrap Around Burst 1 = Wrap Around Burst (Default)
CR0.2	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)
CR0.1		
CR0.0		

Notes:

1. *Device will be in the Asynchronous Mode upon power-up or hardware reset.*
2. *CR1.0 to CR1.3 and CR1.5 to CR1.15 = 1 (Default).*
3. *CR0.3 is ignored if in continuous read mode (no warp around).*

4. *A software reset command is required after reading or writing the configuration registers in order to set the device back to array read mode.*
5. *Refer to Table on page 81 for reading the settings and writing onto configuration registers command sequences.*
6. *Configuration Registers can not be programmed out of order. CR0 must be programmed prior to CR01 otherwise the configuration registers will retain their previous settings.*

7.6 Autoselect

The Autoselect is used for manufacturer ID, Device identification, and sector protection information. This mode is primarily intended for programming equipment to automatically match a device with its corresponding programming algorithm. The Autoselect codes can also be accessed in-system. When verifying sector protection, the sector address must appear on the appropriate highest order address bits (see Table on page 30). The remaining address bits are don't care. The most significant four bits of the address during the third write cycle selects the bank from which the Autoselect codes are read by the host. All other banks can be accessed normally for data read without exiting the Autoselect mode.

- To access the Autoselect codes, the host system must issue the Autoselect command.
- The Autoselect command sequence may be written to an address within a bank that is either in the read or erase-suspend-read mode.
- The Autoselect command may not be written while the device is actively programming or erasing. Autoselect does not support simultaneous operations or burst mode.
- The system must write the reset command to return to the read mode (or erase-suspend-read mode if the bank was previously in Erase Suspend).

See Table on page 81 for command sequence details.

Autoselect Addresses

Description	Address	Read Data
Manufacturer ID Word 00	(BA) + 00h	0001h
Device ID, Word 01	(BA) + 01h	227Eh
Sector Lock/Unlock Word 02	(SA) + 02h	0001h = Locked, 0000h = Unlocked
Indicator Bits Word 03	(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, DQ2 - DQ0 = reserved
Device ID, Word 0E	(BA) + 0Eh	223Dh (WS512P)-1CE# 2242h (WS256P) 2244h (WS128P)
Device ID, Word 0F	(BA) + 0Fh	2200h

Software Functions and Sample Code

Autoselect Entry

(LLD Function = lld_AutoselectEntryCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	BA+AAAh	BA+555h	0x00AAh
Unlock Cycle 2	Write	BA+555h	BA+2AAh	0x0055h
Autoselect Command	Write	BA+AAAh	BA+555h	0x0090h

Autoselect Exit

(LLD Function = lld_AutoselectExitCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	xxxxh	xxxxh	0x00F0h

Notes:

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

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

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

UINT16 manuf_id;

/* Auto Select Entry */

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


/* multiple reads can be performed after entry */

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

/*  Autoselect exit */

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

7.7 Program/Erase Operations

These devices are capable of several modes of programming and or erase operations which are described in detail in the following sections. However, prior to any programming and or erase operation, devices must be setup appropriately as outlined in the configuration register (Table on page 29).

During synchronous write operations, including writing command sequences, the system must drive AVD# and CE# to V_{IL}, and OE# to V_{IH} when providing an address to the device, and drive WE# and CE# to V_{IL}, and OE# to V_{IH} when writing commands or programming data. Addresses are latched on the rising edge of AVD# pulse or rising edge of CLK or falling edge of WE#, whichever occurs first.

During asynchronous write operations, addresses are latched on the rising edge of AVD# or falling edge of WE# while data is latched on the 1st rising edge of WE#, or CE# whichever comes first.

Note the following:

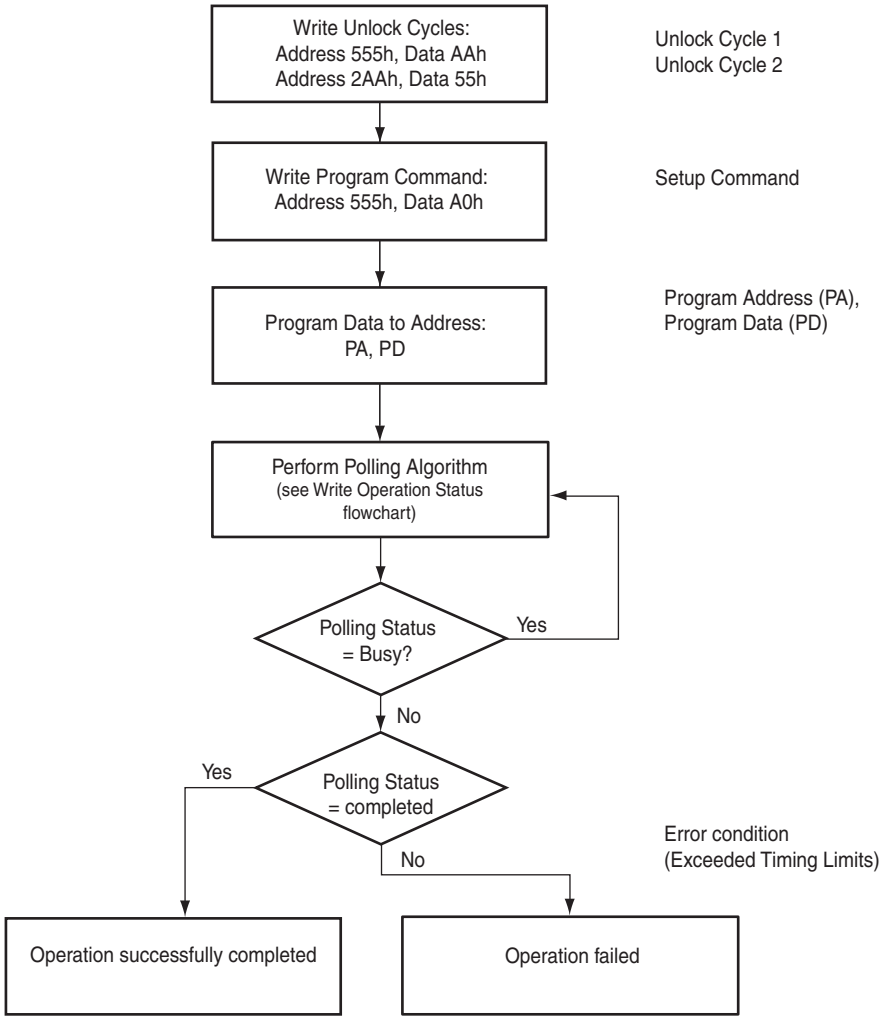
- When the Embedded Program/Erase algorithm is complete, the device returns to the read mode.
- The system can determine the status of the Program/Erase operation. Refer to *Write Operation Status* *on page 44* for further information.
- While *1* can be programmed to *0*, a *0* cannot be programmed to a *1*. Any such attempt will be ignored as only an erase operation can covert a *0* to a *1*. For example:
Old Data = 0011
New Data = 0101
Result = 0001
- Any commands written to the device during the Embedded Program/Erase Algorithm are ignored except the Program/Erase Suspend commands.
- Secured Silicon Sector, Autoselect, and CFI functions are unavailable when a Program/Erase operation is in progress.
- A hardware reset and/or power removal immediately terminates the Program/Erase operation and the Program/Erase 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 only for single word programming operation. See *Write Buffer Programming* *on page 34* when using the write buffer.

7.7.1 Single Word Programming

Single word programming mode is the simplest method of programming. In this mode, four Flash command write cycles are used to program an individual Flash address. While the single word programming method is supported by all Cypress devices, in general it is not recommended for devices that support Write Buffer Programming. See Table on page 81 for the required bus cycles and Figure 7.3 for the flowchart.

When the Embedded Program algorithm is complete, the device 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.

Figure 7.3 Single Word Program



Software Functions and Sample Code

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

7.7.2 Write Buffer Programming

Write Buffer Programming allows the system to write a maximum of 32 words in one programming operation. This results in a faster effective word programming time than the standard *word* programming algorithms. The Write Buffer Programming command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the Write Buffer Load command written at the Sector Address in which programming will occur. At this point, the system writes the number of *word locations minus 1* that will be loaded into the page buffer at the Sector Address in which programming will occur. This tells the device how many write buffer addresses will be loaded with data and therefore when to expect the *Program Buffer to Flash* confirm command. The number of locations to program cannot exceed the size of the write buffer or the operation will abort. (NOTE: the size of the write buffer is dependent upon which data are being loaded. Also note that the number loaded = the number of locations to program minus 1. For example, if the system will program 6 address locations, then 05h should be written to the device.)

The *write-buffer* addresses must be in the same sector for all address/data pairs loaded into the write buffer. It is to be noted that Write Buffer Programming cannot be performed across multiple sectors. If the system attempts to load programming data outside of the selected *write-buffer* addresses, the operation aborts after the Write to Buffer command is executed. Also, the starting address must be the least significant address. All subsequent addresses and write buffer data must be in sequential order.

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 be in sequential order.

After writing the Starting Address/Data pair, the system then writes the remaining address/data pairs into the write buffer. Write buffer locations must be loaded in sequential order starting with the lowest address in the page. Note that if the number of address/data pairs do no match the word count, the program buffer to flash command is ignored.

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 in the following ways:

- Load a value that is greater than the buffer size during the *Number of Locations to Program* step (DQ7 is not valid in this condition).
- 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.

Software Functions and Sample Code

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

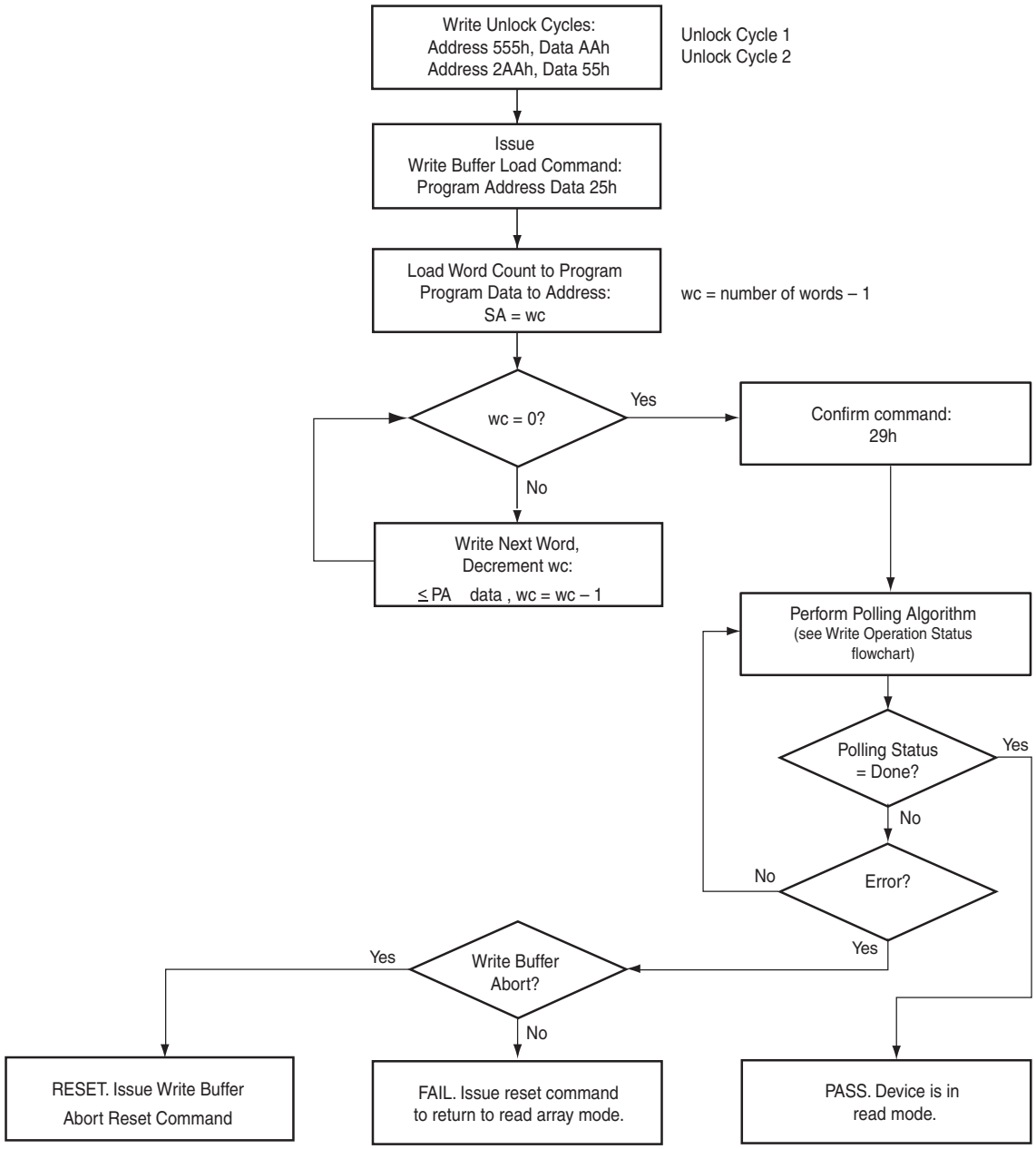
Notes:

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress 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 write buffer. */
/* A write buffer begins at addresses evenly divisible */
/* by 0x20.
UINT16 i; */
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 *)dst ) = 0x0025; /* write write buffer load command */
*( (UINT16 *)dst ) = wc; /* write word count (minus 1) */
for (i=0;i<=wc;i++)
{
*dst++ = *src++; /* ALL dst MUST BE in same Write Buffer */
}
*( (UINT16 *)sector_address ) = 0x0029; /* write confirm command */
/* poll for completion */
/* Example: Write Buffer Abort Reset */
*( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (UINT16 *)base_addr + 0x555 ) = 0x00F0; /* write buffer abort reset */
```

Figure 7.4 Write Buffer Programming Operation



7.7.3 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). Bank address needs to be provided when writing the Program Suspend Command. The status bits are undefined during the t_{PSL} period. To verify that the device is in the suspended state, either:

- wait until after t_{PSL} to check the status bits
- perform a read and check that the status bits return array data
- check whether any Autoselect commands are accepted.

After the programming operation has been suspended, the system can read array data from any non-suspended sector. The Program Suspend command may also be issued during a programming operation while an erase is suspended. In this case, data may be read from any addresses not in Erase Suspend or Program Suspend. If a read is needed from the Secured Silicon Sector area, then user must use the proper command sequences to enter and exit this region.

The system may also write the Autoselect command sequence when the device is in Program Suspend mode. The device allows reading Autoselect codes in the suspended sectors, since the codes are not stored in the memory array. When the device exits the Autoselect mode, the device reverts to Program Suspend mode, and is ready for another valid operation. See *Autoselect* *on page 30* for more information.

After the Program Resume command is written, the device reverts to programming. The system can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard program operation. See *Write Operation Status* *on page 44* for more information.

Note: While a program operation can be suspended and resumed multiple times, a minimum delay of t_{PRS} (Program Resume to Suspend) is required from resume to the next suspend.

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

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

7.7.4 Sector Erase

The sector erase function erases one or more sectors in the memory array (see Table on page 81 and Figure 7.5 on page 39). The device does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically programs and verifies the entire memory for an all zero data pattern prior to electrical erase. After a successful sector erase, all locations within the erased sector contain FFFFh. The system is not required to provide any controls or timings during these operations.

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 *DQ3: Sector Erase Timeout State Indicator on page 47*). 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 on page 44* for information on these status bits.

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

Figure 7.5 on page 39 illustrates the algorithm for the erase operation. Refer to *Program/Erase Operations on page 32* for parameters and timing diagrams.

Software Functions and Sample Code

Sector Erase

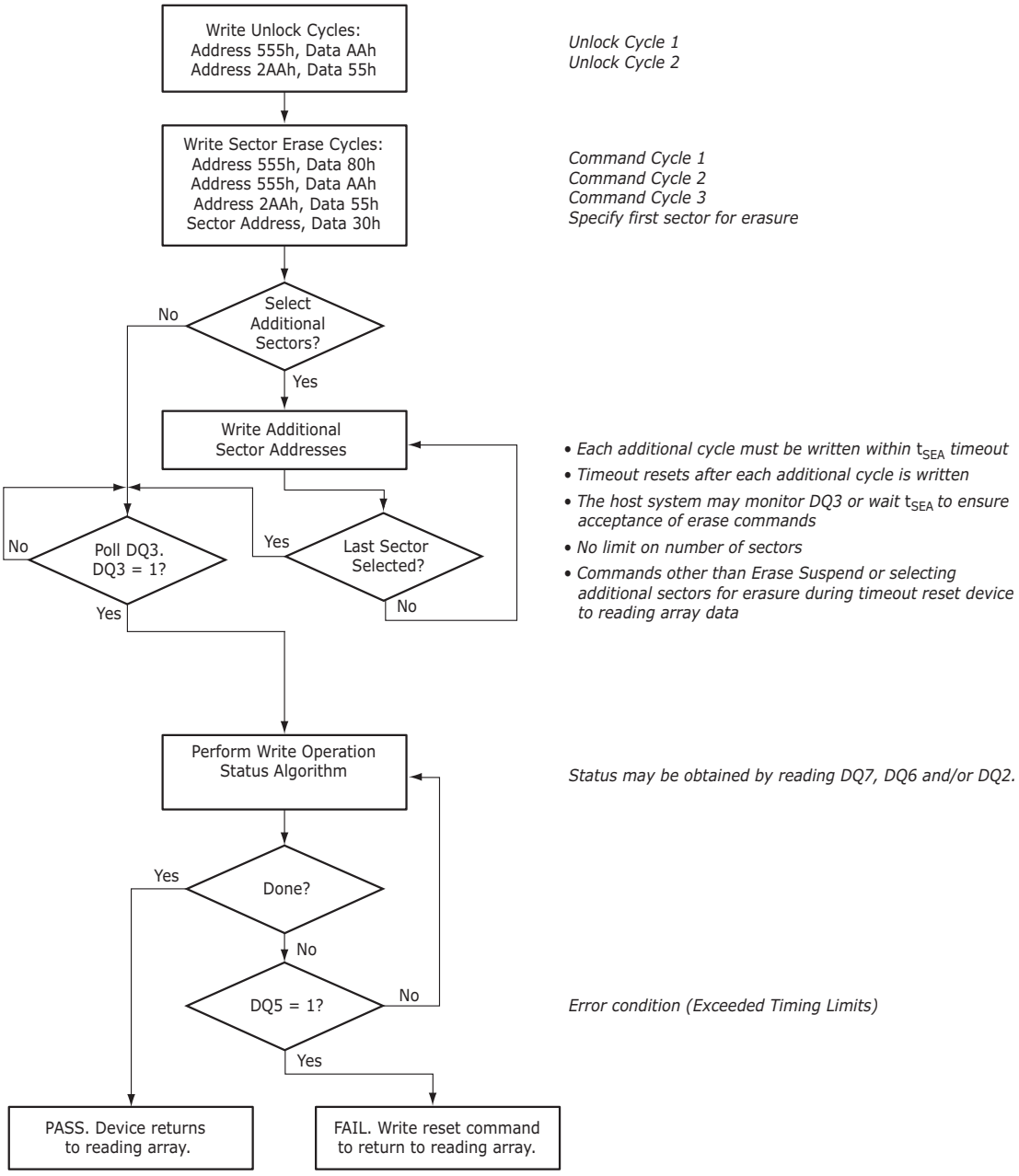
(LLD Function = lld_SectorEraseCmd)

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

Figure 7.5 Sector Erase Operation



Notes:

1. *See Table on page 81 for erase command sequence.*
2. *See the section on DQ3 for information on the sector erase timeout.*

7.7.5 Chip Erase Command Sequence

Chip erase is a six-bus cycle operation as indicated by Table on page 81. These commands invoke the Embedded Erase algorithm, which does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically preprograms and verifies the entire memory for an all zero data pattern prior to electrical erase. After a successful chip erase, all locations of the chip contain FFFFh. The system is not required to provide any controls or timings during these operations. Table 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 on page 44* 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

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

7.7.6 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, after 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 after the t_{SEA} time-out period has expired and during the sector erase operation, the device requires a minimum of t_{ESL} (erase suspend latency) to suspend the erase operation. The status bits are undefined during the t_{ESL} period. To verify that the device is in the suspended state, either:

- wait until after t_{ESL} to check the status bits
- perform a read and check that the status bits return array data
- check whether any Autoselect commands are accepted

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 on page 48 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.

Note: While an erase operation can be suspended and resumed multiple times, a minimum delay of t_{ERS} (Erase Resume to Suspend) is required from resume to the next suspend.

In the erase-suspend-read mode, the system can also issue the Autoselect command sequence. Refer to *Write Buffer Programming on page 34* and *Autoselect on page 30* 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

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

7.7.7 Accelerated Program/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 program and erase functions must not be used more than 10 times per sector. In addition, accelerated 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 accelerated 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 accelerated erase, or device damage may result.
- The ACC pin must not be left floating or unconnected; inconsistent behavior of the device may result.
- ACC locks all sector if set to V_{IL}; ACC should be set to V_{IH} for all other conditions.

7.7.8 Unlock Bypass

The unlock bypass feature allows the system to primarily program faster than using the standard program command sequence, and it is not intended for use during erase. The unlock bypass command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the unlock bypass command, 20h. The device then enters the unlock bypass mode. A two-cycle unlock bypass program command sequence is all that is required to program in this mode. The first cycle in this sequence contains the unlock bypass program command, A0h; the second cycle contains the program address and data. Additional data is programmed in the same manner. This mode dispenses with the initial two unlock cycles required in the standard program command sequence, resulting in faster total programming time. The erase command sequences are four cycles in length instead of six cycles. Table on page 81 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.

The device offers accelerated program operations through the ACC input. When the system asserts V_{HH} on this input, the device automatically enters the Unlock Bypass mode. The system may then write the two-cycle Unlock Bypass program command sequence. The device uses the higher voltage on the ACC input to accelerate the operation.

Refer to *Erase/Program Timing on page 72* for parameters, and Figure 11.15 on page 73 and Figure 11.16 on page 74 for timing diagrams.

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 *Cypress Low Level Driver User's Guide* (available soon on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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.                                               */
```

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.   */
```

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;
```

7.7.9 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 when write buffer programming is used. Reading Data# Polling status on any word other than the last word to be programmed in the write-buffer-page returns false status information. Similarly, attempting to program *1* over a *0* does not return valid Date# 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. 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.

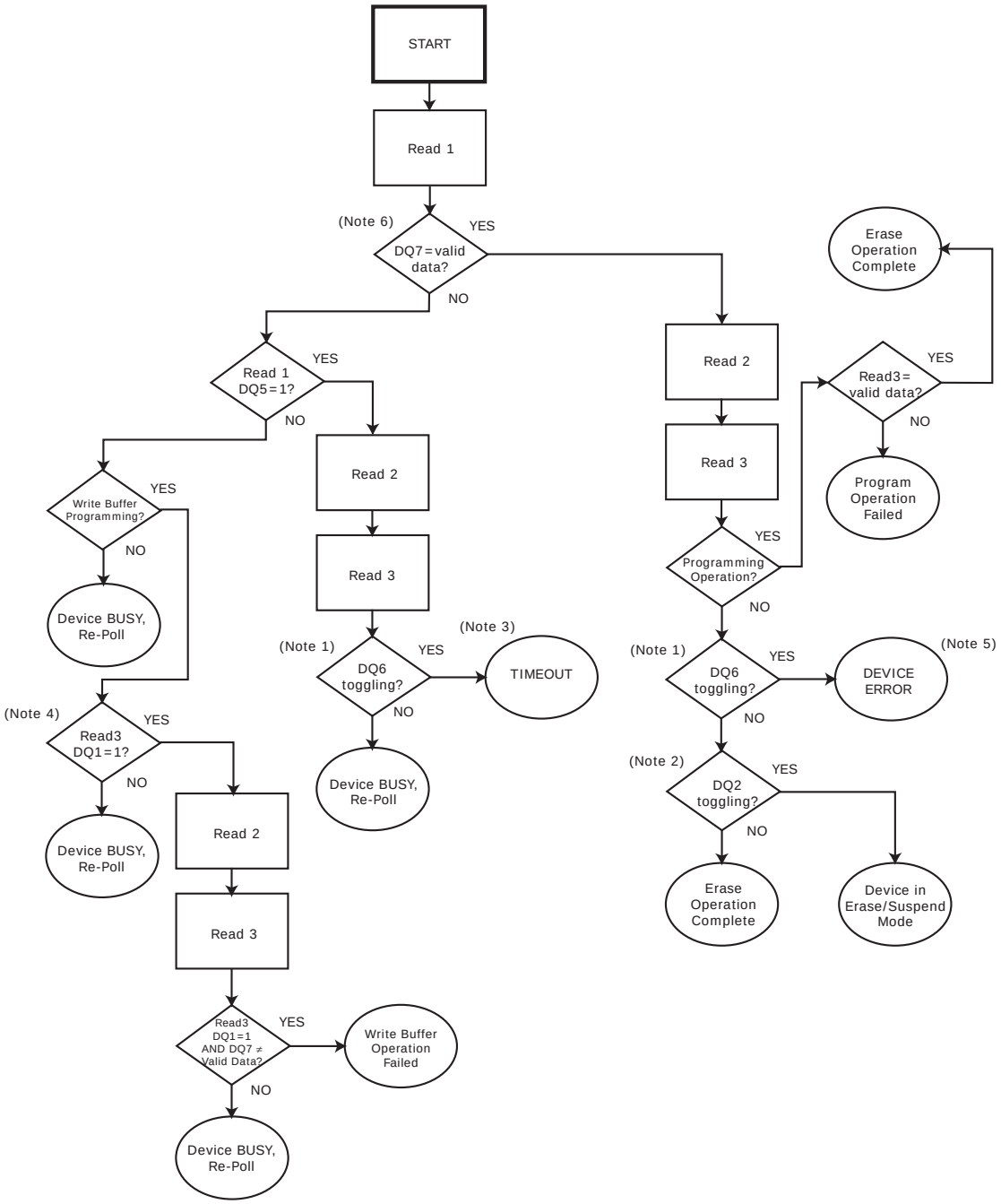
During the Embedded Erase Algorithm, Data# polling produces a *0* on DQ7. When the Embedded Erase algorithm is complete, or if the bank enters the Erase Suspend mode, Data# Polling produces a *1* on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status 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-DQ1 while Output Enable (OE#) is asserted low. That is, the device may change from providing status information to valid data on DQ7. Even if the device has completed the program or erase operation and DQ7 has valid data, the data outputs on DQ6-DQ1 may be still invalid. Valid data on DQ7-DQ1 appears on successive read cycles.

See the following for more information: Table on page 48, shows the outputs for Data# Polling on DQ7. Figure 7.6 on page 45, shows the Data# Polling algorithm; and Figure 11.19 on page 76, shows the Data# Polling timing diagram.

Figure 7.6 Write Operation Status Flowchart



Notes:

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

DQ6: Toggle Bit I

Toggle Bit I on DQ6 indicates whether an Embedded Program or Erase algorithm is in progress or complete, or whether the device has entered the Erase Suspend mode. Toggle Bit I may be read at any address in the same bank, and is valid after the rising edge of the final WE# pulse in the command sequence (prior to the program or erase operation), 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 7.6 on page 45, Figure 11.20 on page 76, and Table on page 46 and Table on page 48.

Toggle Bit I on DQ6 requires Read address to be relatched by toggling AVD# for each reading cycle.

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 to compare outputs for DQ2 and DQ6. See the following for additional information: Figure 7.6 on page 45, *DQ6: Toggle Bit I on page 46*, and Figures 11.19–11.22.

Read address has to be relatched by toggling AVD# for each reading cycle.

DQ6 and DQ2 Indications

If device is	and the system reads	then DQ6	and DQ2
programming,	at any address at the bank being programmed	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 at the bank being programmed	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 7.6 on page 45 for more details.

Note: When verifying the status of a write operation (embedded program/erase) of a memory bank, DQ6 and DQ2 toggle between high and low states in a series of consecutive and contiguous status read cycles. In order for this toggling behavior to be properly observed, the consecutive status bit reads must not be interleaved with read accesses to other memory banks. If it is not possible to temporarily prevent reads to other memory banks, then it is recommended to use the DQ7 status bit as the alternative method of determining the active or inactive status of the write operation.

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 does not 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 ignores the bit that was incorrectly instructed to be programmed from a *0* to a *1*, while any other bits that were correctly requested to be changed from *1* to *0* are programmed. Attempting to program a *0* to a *1* is masked during the programming operation. Under valid DQ5 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 accepts 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 on page 48 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.

Write Operation Status

Status			DQ7 (Note 2)	DQ6	DQ5 (Note 1)	DQ3	DQ2 (Note 2)	DQ1 (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****.*

7.8 Simultaneous Read/Program or Erase

The simultaneous read/program or erase 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 11.25 on page 79 shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to *DC Characteristics on page 63* for read-while-program and read-while-erase current specification.

7.9 Writing Commands/Command Sequences

When the device is configured for Asynchronous read, only Asynchronous write operations are allowed, and CLK is ignored. When in the Synchronous read mode configuration, the device is able to perform both Asynchronous and Synchronous write operations. CLK and AVD# induced address latches are supported in the Synchronous programming mode. During a synchronous write operation, to write a command or command sequence (which includes programming data to the device and erasing sectors of memory), the system must drive AVD# and CE# to V_{IL}, and OE# to V_{IH} when providing an address to the device, and drive WE# and CE# to V_{IL}, and OE# to V_{IH} when writing commands or data. During an asynchronous write operation, the system must drive CE# and WE# to V_{IL} and OE# to V_{IH} when providing an address, command, and data. Addresses are latched on the last falling edge of WE# or CE#, while data is latched on the 1st rising edge of WE# or CE#. An erase operation can erase one sector, multiple sectors, or the entire device. Table on page 11 and Table on page 12 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* *on page 63* 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.

7.10 Handshaking

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

7.11 Hardware Reset

The RESET# input provides a hardware method of resetting the device to reading array data. When RESET# is driven low for at least a period of t_{RP}, the device immediately terminates any operation in progress, tristates all outputs, resets the configuration register, and ignores all read/write commands for the duration of the RESET# pulse. The device also resets the internal state machine to reading array data.

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

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

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

See Figure 11.5 on page 65 and Figure 11.14 on page 72 for timing diagrams.

7.12 Software Reset

Software reset is part of the command set (see Table on page 81) that also returns the device to array read mode and must be used for the following conditions:

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

Software Functions and Sample Code

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 *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress Flash memory software development guidelines.

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

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

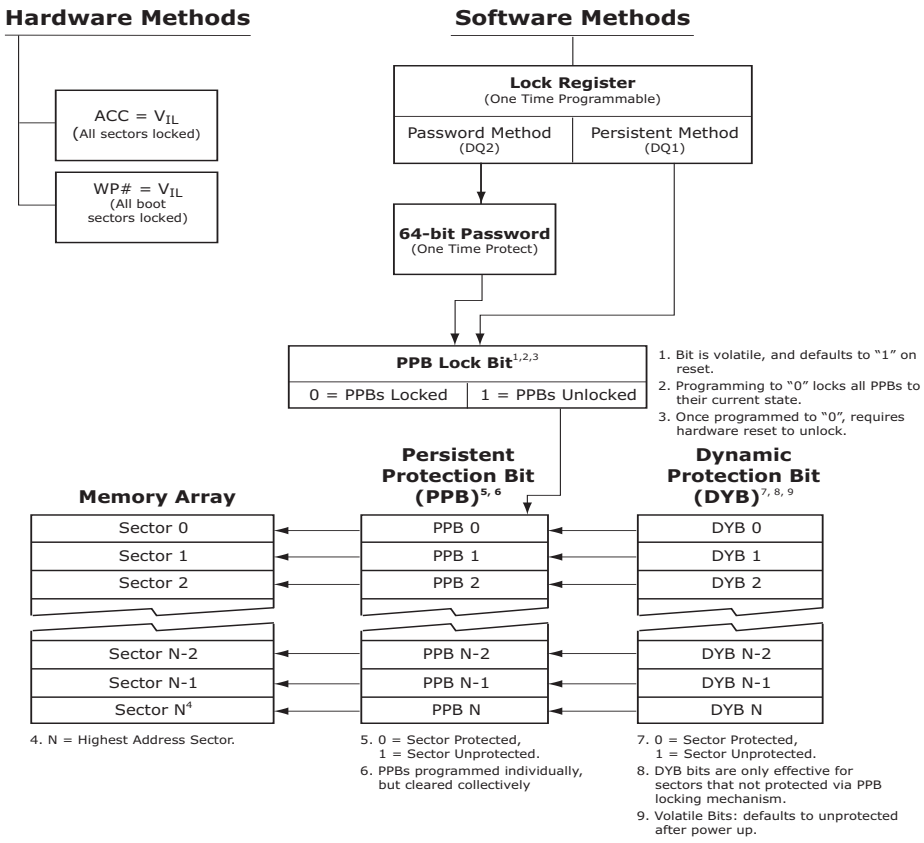
- This command resets the banks to the read and address bits are ignored.
- Reset commands are ignored once erasure has begun until the operation is complete.
- Once programming begins, the device ignores reset commands until the operation is complete

- The reset command may be written between the cycles in a program command sequence before programming begins (prior to the third cycle). This resets the bank to which the system was writing to the read mode.
- If the program command sequence is written to a bank that is in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.
- The reset command may be also written during an Autoselect command sequence.
- If a bank has entered the Autoselect mode while in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.
- If DQ1 goes high during a Write Buffer Programming operation, the system must write the “Write to Buffer Abort Reset” command sequence to RESET the device to reading array data. The standard RESET command does not work during this condition.
- 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 on page 51.

Figure 8.1 Advanced Sector Protection/Unprotection



8.1 Advanced Sector Protection Software Examples

Sector Protection Schemes

Unique Device PPB Lock Bit 0 = locked 1 = unlocked		Sector PPB 0 = protected 1 = unprotected	Sector DYB 0 = protected 1 = 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 contains all possible combinations of the DYB, PPB, and PPB Lock Bit relating to the status of the sector.

8.2 Lock Register

The Lock Register consists of 4 bits. The Secured Silicon Sector Protection Bit is DQ0, Persistent Protection Mode Lock Bit is DQ1, Password Protection Mode Lock Bit is DQ2, Persistent Sector Protection OTP bit is DQ3. If DQ0 is ‘0’, it means that the Customer Secured Silicon area is locked and if DQ0 is ‘1’, it means that it is unlocked. When DQ2 is set to ‘1’ and DQ1 is set to ‘0’, the device can only be used in the Persistent Protection Mode. When the device is set to Password Protection Mode, DQ1 is required to be set to ‘1’ and DQ2 is required to be set to ‘0’. DQ3 is programmed in the Cypress factory. When the device is programmed to disable all PPB erase command, DQ3 outputs a ‘0’, when the lock register bits are read. Similarly, if the device is programmed to enable all PPB erase command, DQ3 outputs a ‘1’ when the lock register bits are read. Likewise the DQ4 bit is also programmed in the Cypress Factory. DQ4 is the bit which indicates whether Volatile Sector Protection Bit (DYB) is protected or not after boot-up. When the device is programmed to set all Volatile Sector Protection Bit protected after power-up, DQ4 outputs a ‘0’ when the lock register bits are read. Similarly, when the device is programmed to set all Volatile Sector Protection Bit un-protected after power-up, DQ4 outputs a ‘1’. Each of these bits in the lock register are non-volatile. DQ15-DQ5 are reserved and will be 1’s.

For programming lock register bits refer to Table on page 83.

Lock Register

DQ15-5	DQ4	DQ3	DQ2	DQ1	DQ0
1’s	Reserved (default = 1)	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	Secured Silicon Sector Protection Bit

Notes:

1. *If the password mode is chosen, the password must be programmed and verified before setting the corresponding lock register bit (DQ2). Failing to program and verifying the password prior to setting lock register (DQ2), causes all sectors to lock out.*
2. *It is recommended a sector protection method to be chosen by programming DQ1 or DQ2 prior to shipment.*
3. *After the Lock Register Bits Command Set Entry sequence is written, reads and writes for Bank 0 are disabled, while reads from other banks are allowed until exiting this mode. Simultaneous operation is only valid as long as lock register program command is not executed.*
4. *If both lock bits are selected to be programmed (to zeros) at the same time, the operation aborts.*
5. *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.*
6. *During erase/program suspend, ASP entry commands are not allowed.*
7. *Data Polling can be done immediately after the lock register programming command sequence (no delay required). Note that status polling can be done only in bank 0*
8. *Reads from other banks (simultaneous operation) are not allowed during lock register programming. This restriction applies to both synchronous and asynchronous read operations.*

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 8.3–8.6.

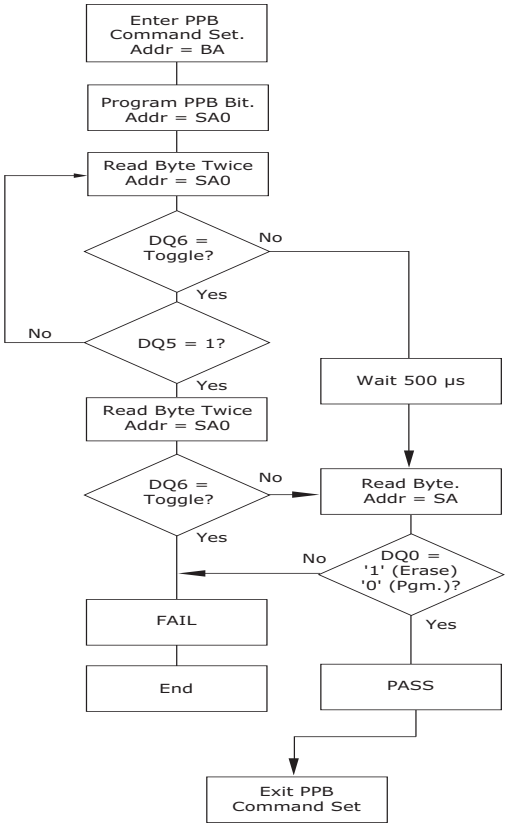
8.3 Persistent Protection Bits

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. While programming PPB for a sector, array data can *not* be read from any other banks.
3. Entry command disables reads and writes for the bank selected.
4. Reads within that bank return the PPB status for that sector.
5. Reads from other banks are allowed while program/erase is not allowed.
6. All Reads must be performed using the Asynchronous mode.
7. The specific sector address (Amax-A14) are written at the same time as the program command.
8. If the PPB Lock Bit is set, the PPB Program or erase command does not execute and times-out without programming or erasing the PPB.
9. There are no means for individually erasing a specific PPB and no specific sector address is required for this operation.
10. The PPB Exit command must be issued after the execution which resets the device to read mode and re-enables reads and writes for Bank 0
11. 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 shown in Figure 8.2 on page 53.
12. During PPB program / erase data polling can be done synchronously.
13. If the user attempts to program or erase a protected sector, the device ignores the command and returns to read mode.

Figure 8.2 PPB Program/Erase Algorithm



8.4 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 are 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*).
2. The default state of DYB is unprotected after power up and all sectors can be modified depending on the status of PPB bit for that sector, (erased to *1*). Then the sectors can be modified depending upon the PPB state of that sector (see Table on page 51).
3. It is possible to have sectors that are persistently locked with sectors that are left in the dynamic state.
4. The DYB Set or Clear commands for the dynamic sectors signify protected or unprotected state 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 locks the PPBs, and the device operates normally again.
5. Data polling is not available for DYB program / erase.
6. DYB read data can be done synchronously.
7. If the user attempts to program or erase a protected sector, the device ignores the command and returns to read mode.

8.5 Persistent Protection Bit Lock Bit

The Persistent Protection Bit Lock Bit is a global volatile bit for all sectors. When set (programmed to *0*), it locks all PPBs and when cleared (programmed to *1*), allows the PPBs to be changed. 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.

8.6 Password Protection Method

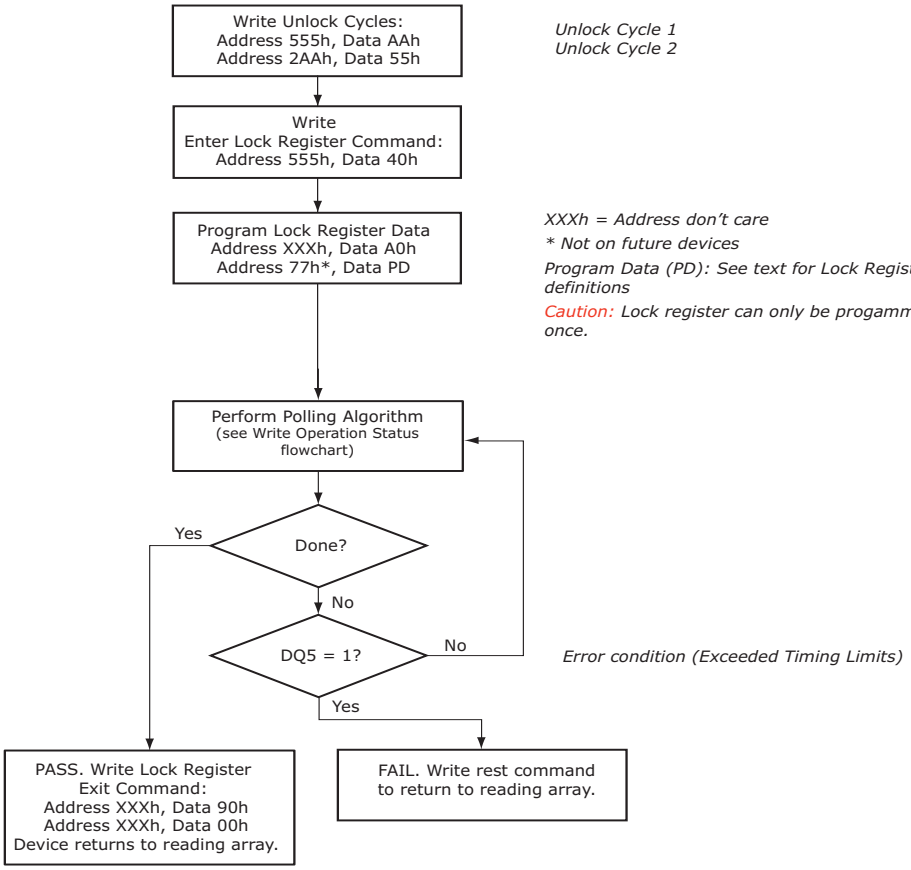
The Password Protection Method allows an even higher level of security than the Persistent Sector Protection Mode by requiring a 64 bit password for unlocking the device PPB Lock Bit. In addition to this password requirement, after power up and reset, the PPB Lock Bit is set *0* to maintain the password 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. If the password mode is chosen, the password must be programmed and verified before setting the corresponding lock register bit (DQ2). Failing to program and verifying the password prior to setting lock register (DQ2), causes all sectors to lock out.
2. 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.
3. 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*.
4. The password is all *1*s when shipped from the factory.
5. All 64-bit password combinations are valid as a password.

6. There is no means to verify what the password is after it is set.
7. The Password Mode Lock Bit, once set, prevents reading the 64-bit password on the data bus and further password programming.
8. The Password Mode Lock Bit is not erasable.
9. The lower two address bits (A1–A0) are valid during the Password Read, Password Program, and Password Unlock.
10. The exact password must be entered in order for the unlocking function to occur.
11. 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.
12. Approximately 1 µs is required for unlocking the device after the valid 64-bit password is given to the device.
13. Password verification is only allowed during the password programming operation.
14. All further commands to the password region are disabled and all operations are ignored.
15. If the password is lost after setting the Password Mode Lock Bit, there is no way to clear the PPB Lock Bit.
16. 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.
17. If the user attempts to program or erase a protected sector, the device ignores the command and returns to read mode.
18. 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.
19. 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 8.3 Lock Register Program Algorithm



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 (including Secured Silicon Area) are locked.
- When ACC is at V_{IL}, all sectors (including Secured Silicon Area) 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.

S29WS512P Sector Protection

	Dual Boot Configuration
Bank 0	SA000-SA003 WP# Protected
Bank 1-7	No Sector WP# Protection
Bank 8-14	No Sector WP# Protection
Bank 15	SA514-SA517 WP# Protected

S29WS256P Sector Protection

	Dual Boot Configuration
Bank 0	SA000-SA003 WP# Protected
Bank 1-7	No Sector WP# Protection
Bank 8-14	No Sector WP# Protection
Bank 15	SA258-SA261 WP# Protected

S29WS128P Sector Protection

	Dual Boot Configuration
Bank 0	SA000-SA003 WP# Protected
Bank 1-7	No Sector WP# Protection
Bank 8-14	No Sector WP# Protection
Bank 15	SA130-SA133 WP# Protected

If the system asserts V_{IL} on the WP# pin, the device disables program and erase functions in the *outermost* boot sectors, as well as Secured Silicon Area. 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 (including the Secured Silicon Area) 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* *on page 63* represents the standby current specification

9.2 Automatic Sleep Mode

The automatic sleep mode minimizes Flash device energy consumption only while in asynchronous main array read 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* *on page 63* 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 is greater.

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

9.4 Output Disable (OE#)

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

10. Secured Silicon Sector Flash Memory Region

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

Please note the following general conditions:

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

Secured Silicon Sector Addresses

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

10.1 Factory Secured Silicon Sector

The Factory Secured Silicon Sector is always protected when shipped from the factory and has the Factory Indicator Bit (DQ7) permanently set to a *1*. This prevents cloning of a factory locked part and ensures the security of the ESN and customer code once the product is shipped to the field.

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

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

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

10.2 Customer Secured Silicon Sector

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

Please note the following:

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

10.3 Secured Silicon Sector Entry/Exit Command Sequences

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

See Command Definition Table [Secured Silicon Sector Command Table, Appendix
Table on page 81 for address and data requirements for both command sequences.

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

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

After the system has written the Enter Secured Silicon Sector command sequence, it may read the Secured Silicon Sector by using the addresses normally occupied by sector SA0 within the memory array. This mode of operation continues until the system issues the Exit Secured Silicon Sector command sequence, or until power is removed from the device.

Software Functions and Sample Code

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

Secured Silicon Sector Entry

(LLD Function = lld_SecSiSectorEntryCmd)

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

Note:
Base = Base Address.

```
/* Example: Secured Silicon 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 Secured Silicon Sector Entry Cmd    */
```

Secured Silicon 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 Secured Silicon Sector mode, you program */
/* words using the programming algorithm.     */
```

Secured Silicon Sector Exit

(LLD Function = lld_SecSiSectorExitCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	Base + AAAh	Base + 555h	00AAh
Unlock Cycle 2	Write	Base + 554h	Base + 2AAh	0055h
Exit Cycle 3	Write	Base + AAAh	Base + 555h	0090h
Exit Cycle 4	Write	Any address	Any address	0000h

Note:
Base = Base Address.

```
/* Example: Secured Silicon 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 Secured Silicon Sector Exit cycle 3 */
  *( (UINT16 *)base_addr + 0x000 ) = 0x0000;   /* write Secured Silicon Sector Exit cycle 4 */
```

11. 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 + 2.5 V
V_{CC} (Note 1)	–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 11.1 Maximum Negative Overshoot Waveform

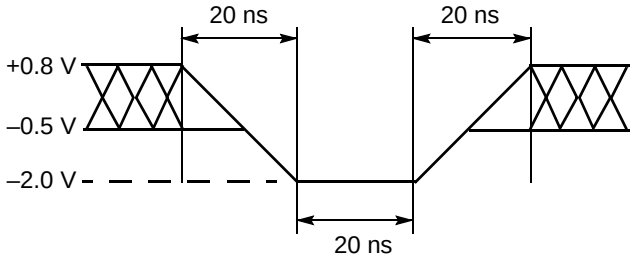
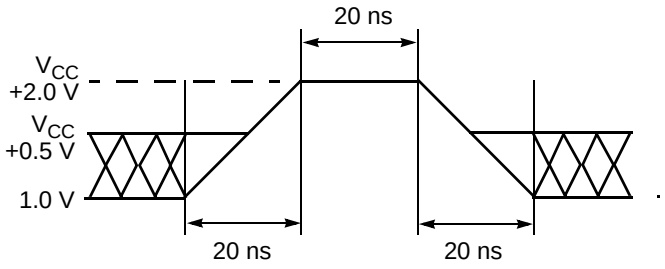


Figure 11.2 Maximum Positive Overshoot Waveform



11.2 Operating Ranges

Wireless (I) Devices	Ambient Temperature (T_A)	–25°C to +85°C
Supply Voltages	V_{CC} Supply Voltages	+1.70 V to +1.95 V

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

11.3 DC Characteristics

11.3.1 CMOS Compatible

CMOS Compatible (Sheet 1 of 2)

Parameter	Description	Test Conditions (Note 1)		Min	Typ	Max	Unit
I_{LI}	Input Load Current	$V_{IN} = V_{SS}$ to V_{CC}, $V_{CC} = V_{CC}$max				±1	µA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{SS}$ 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		32	37	mA
			66 Mhz		35	41	mA
			80 Mhz		39	46	mA
			104 Mhz		44	51	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 16	54 Mhz		32	37	mA
			66 Mhz		35	41	mA
			80 Mhz		39	46	mA
			104 Mhz		44	51	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 32	54 Mhz		33	38	mA
			66 Mhz		36	42	mA
			80 Mhz		40	47	mA
			104 Mhz		45	52	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = Continuous	54 Mhz		34	39	mA
			66 Mhz		37	43	mA
			80 Mhz		41	48	mA
			104 Mhz		50	57	mA
I_{CC1}	V_{CC} Active Asynchronous Read Current (Note 2)	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}	10 MHz		40	80	mA
			5 MHz		20	40	mA
			1 MHz		10	20	mA
I_{CC2}	V_{CC} Active Program/Erase Current (Note 2)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}	V_{ACC}		1	5	µA
			V_{CC}		20	60	mA
I_{CC3}	V_{CC} Standby Current (Note 3)	CE# = RESET# = V_{CC} ± 0.2 V	V_{ACC}		1	5	µA
			V_{CC}		20	70	µA
I_{CC4}	V_{CC} Reset Current	RESET# = V_{IL}, CLK = V_{IL}			30	60	µA
I_{CC5}	V_{CC} Active Current (Read While Program/Erase)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}, 5 MHz			40	60	mA
I_{CC6}	V_{CC} Sleep Current	CE# = V_{IL}, OE# = V_{IH}, (V_{CCQ} or V_{SSQ} biased at Rail to Rail for all inputs)			5	40	µA
I_{CC7}	V_{CC} Active Page Read Current	OE# = V_{IH}, 8 word Page Read			10	15	mA
I_{ACC}	Accelerated Program Current (Note 4)	CE# = V_{IL}, OE# = V_{IH}, V_{ACC} = 9.5 V	V_{ACC}		7	10	mA
			V_{CC}		15	20	mA
V_{IL}	Input Low Voltage			–0.2		0.4	V
V_{IH}	Input High Voltage			V_{CC} – 0.4		V_{CC} + 0.4	
V_{OL}	Output Low Voltage	I_{OL} = 100 µA, $V_{CC} = V_{CC\ min}$				0.1	V

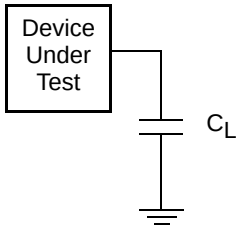
CMOS Compatible (Sheet 2 of 2)

Parameter	Description	Test Conditions (Note 1)	Min	Typ	Max	Unit
V_{OH}	Output High Voltage	I_{OH} = –100 µA, V_{CC} = $V_{CC\ min}$	V_{CC} – 0.1			V
V_{HH}	Voltage for Accelerated Program		8.5		9.5	V
V_{LKO}	Low V_{CC} Lock-out Voltage				1.4	V

Notes:
1. *Maximum I_{CC} specifications are tested with V_{CC} = V_{CC}max.*
2. *I_{CC} active while Embedded Erase or Embedded Program is in progress.*
3. *Device enters automatic sleep mode when addresses are stable for t_{ACC} + 20 ns. Typical sleep mode current is equal to I_{CC3}.*
4. *Total current during accelerated programming is the sum of V_{ACC} and V_{CC} currents.*
5. *V_{CCQ} = V_{CC} during all I_{CC} measurements.*
6. *V_{IH} = V_{CC} <Helv>± 0.2V and V_{IL} □0.1V*

11.4 Test Conditions

Figure 11.3 Test Setup



Test Specifications

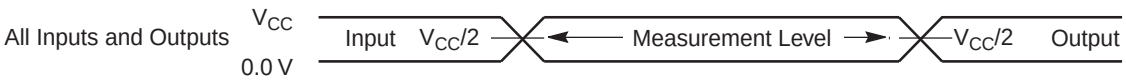
Test Condition	All Speed Options	Unit
Output Load Capacitance, C_L (including jig capacitance)	30	pF
Input Rise and Fall Times	1.0 - 1.50	ns
Input Pulse Levels	0.0–V_{CC}	V
Input timing measurement reference levels	V_{CC}/2	V
Output timing measurement reference levels	V_{CC}/2	V

11.5 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)

11.6 Switching Waveforms

Figure 11.4 Input Waveforms and Measurement Levels



V_{CC} Power-up

Parameter	Description	Test Setup	Time	Unit
t_{VCS}	V_{CC} Setup Time	Min	30	µs
t_{RH}	Time between RESET# (high) and CE# (low)	Min	200	ns

11.7 Power-up/Initialization

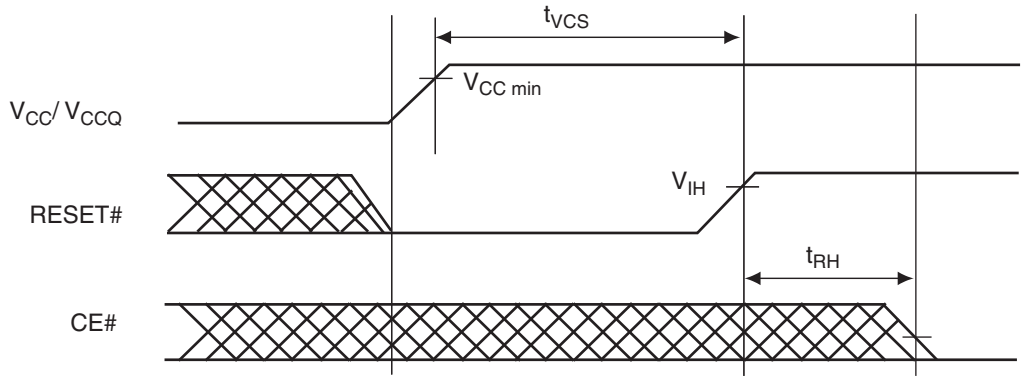
Power supply must reach its minimum voltage range before applying/removing the next supply voltage.

RESET# must ramp down to V_{IL} level before V_{CC}/V_{CCQ} can start ramp up.

V_{CC} and V_{CCQ} must be ramped simultaneously for proper power-up.

The S29WS-P device ramp rate is > 1V/400 µs. For V_{CC} ramp rate <1V/400 µs, a hardware reset is required.

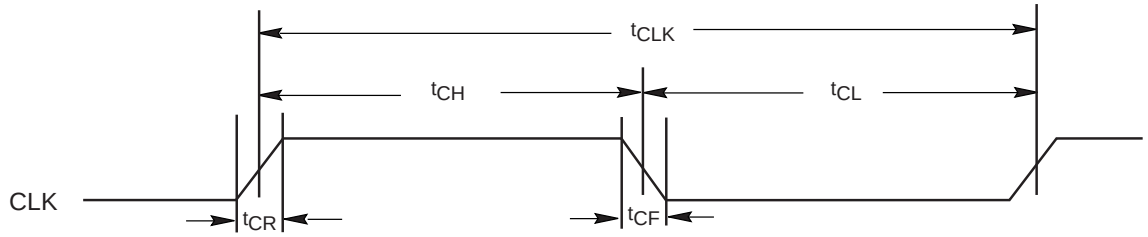
Figure 11.5 V_{CC} Power-up Diagram



11.8 CLK Characterization

Parameter	Description		54 MHz	66 MHz	80 MHz	104 MHz	Unit
f_{CLK}	CLK Frequency	Max	54	66	80	104	MHz
		Min	60 KHz in 8 word Burst, 120 KHz in 16 word Burst, 250 KHz in 32 word Burst, 1 MHz in Continuous Mode				
t_{CLK}	CLK Period	Min	18.5	15.1	12.5	9.62	ns
t_{CL}/t_{CH}	CLK Low/High Time	Min	0.45 t_{CLK}				ns
		Max	0.55 t_{CLK}				
t_{CR}	CLK Rise Time	Max	3.0	3.0	2.5	1.5	ns
t_{CF}	CLK Fall Time						

Figure 11.6 CLK Characterization



11.9 AC Characteristics

11.9.1 Synchronous/Burst Read

Parameter		Description		54 MHz	66 MHz	80 MHz	104 MHz	Unit
JEDEC	Standard							
	t_{IACC}	Synchronous Access Time	Max	(WS-1) * t_{CK} + t_{BACC}				ns
	t_{BACC}	Burst Access Time Valid Clock to Output Delay	Max	13.5	11.2	9	7.6	ns
	t_{ACS}	Address Setup Time to CLK (Note 1)	Min	5	4	4	3.5	ns
	t_{ACH}	Address Hold Time from CLK (Note 1)	Min	6	6	5	5	ns
	t_{BDH}	Data Hold Time	Min	4	3	3	2	ns
	t_{RDY}	Chip Enable to RDY Active	Max	10				ns
	t_{OE}	Output Enable to RDY Low	Max	13.5	11.2	9	7.6	ns
	t_{CEZ}	Chip Enable to High Z	Max	10	10	10	7	ns
	t_{OEZ}	Output Enable to High Z	Max	10	10	10	7	ns
	t_{CES}	CE# Setup Time to CLK	Min	6				ns
	t_{RACC}	Ready Access Time from CLK	Max	13.5	11.2	9	7.6	ns
	t_{CAS}	CE# Setup Time to AVD#	Min	0				ns
	t_{AVC}	AVD# Low to CLK Setup Time	Min	6				ns
	t_{AVD}	AVD# Pulse	Min	t_{CLK}				ns

Notes:

1. *Addresses are latched on the rising edge of CLK*
2. *Synchronous Access Time is calculated using the formula (#of WS - 1)*(clock period) + (t_{BACC} or Clock to Out)*

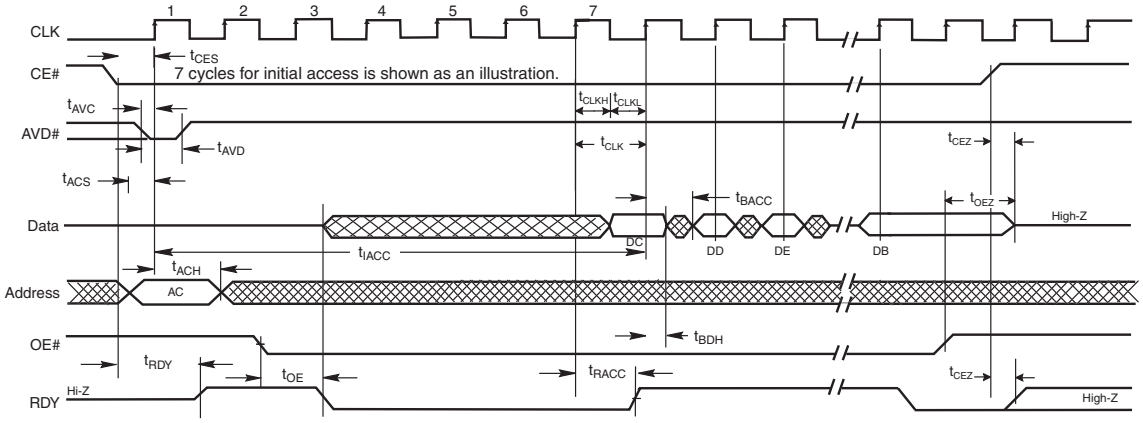
Non-Continuous Burst Mode with Wrap Around Burst Mode.

Max Frequency	Wait State Requirement
Frequency ≤ 27 MHz	3
27 MHz < Frequency ≤ 40 MHz	4
40 MHz < Frequency ≤ 54 MHz	5
54 MHz < Frequency ≤ 66 MHz	6
66 MHz < Frequency ≤ 80 MHz	7
80 MHz < Frequency ≤ 95 MHz	8
95 MHz < Frequency ≤ 104 MHz	11

Continuous Burst Mode with No Wrap Around Burst Mode.

Max Frequency	Wait State Requirement
Frequency ≤ 27 MHz	3
27 MHz < Frequency ≤ 40 MHz	4
40 MHz < Frequency ≤ 54 MHz	6
54 MHz < Frequency ≤ 66 MHz	7
67 MHz < Frequency ≤ 80 MHz	8
80 MHz < Frequency ≤ 95 MHz	9
95 MHz < Frequency ≤ 104 MHz	11

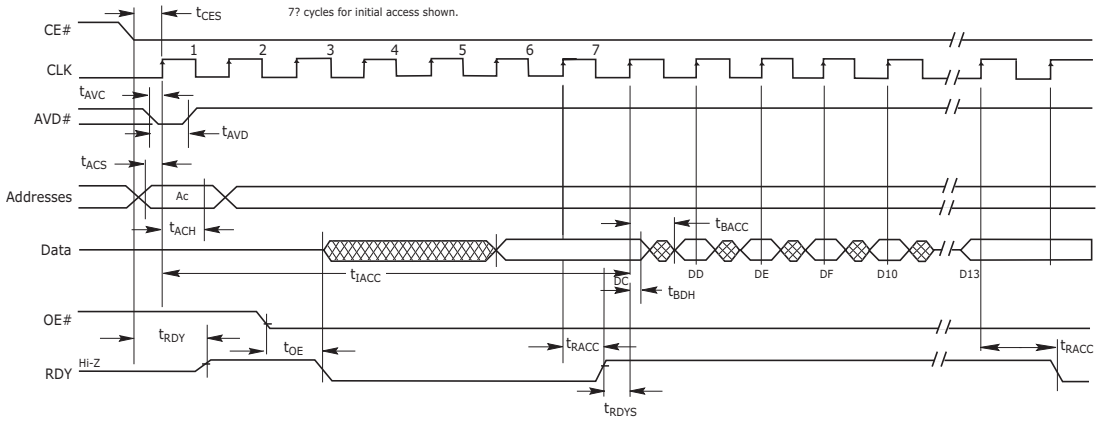
Figure 11.7 8-Word Linear Synchronous Single Data Rate Burst with Wrap Around



Notes:

1. *Figure shows for illustration the total number of wait states set to seven cycles.*
2. *The device is configured synchronous single data rate mode and RDY active with data.*
3. *CE# (High) drives the RDY to Hi-Z while OE# (High) drives the DQ(15:0) pins to Hi-Z.*

Figure 11.8 8-word Linear Single Data Read Synchronous Burst without Wrap Around



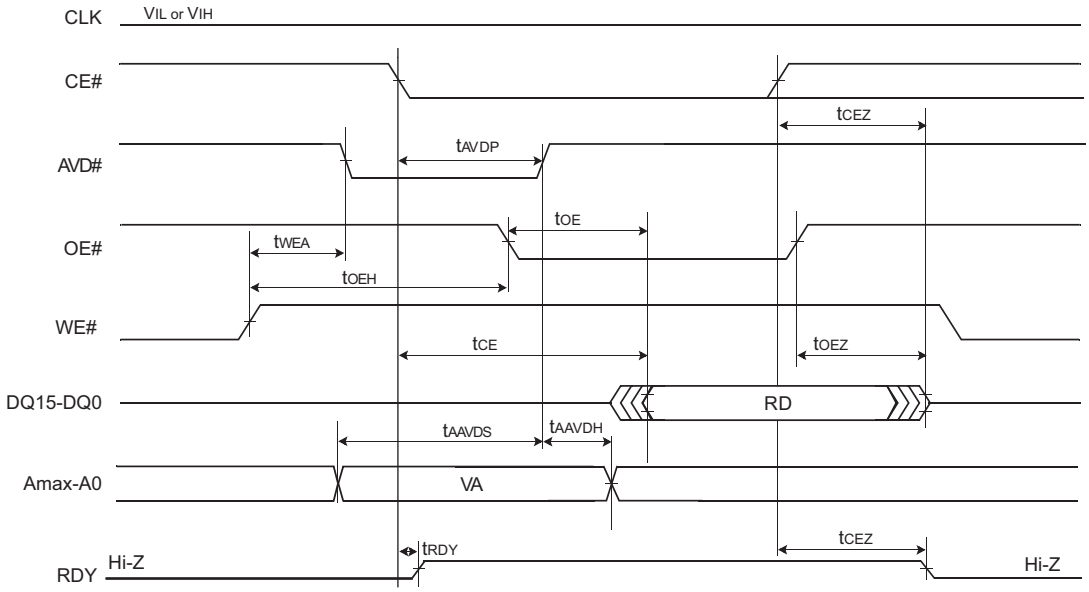
Notes:

1. *Figure shows for illustration the total number of wait states set to seven cycles.*
2. *The device is configured synchronous single data rate mode and RDY active with data.*
3. *CE# (High) drives the RDY to Hi-Z while OE# (High) drives the DQ(15:0) pins to HI-Z.*

11.9.2 Asynchronous Mode Read

Parameter		Description			Asynchronous	Unit
JEDEC	Standard					
	t_{CE}	Access Time from CE# Low		Max	83	ns
	t_{ACC}	Asynchronous Access Time		Max	80	ns
	t_{AVDP}	AVD# Low Time		Min	7.5	ns
	t_{AAVDS}	Address Setup Time to Rising Edge of AVD#		Min	6	ns
	t_{AAVDH}	Address Hold Time from Rising Edge of AVD#		Min	4	ns
	t_{OE}	Output Enable to Output Valid		Max	13.5	ns
	t_{OEH}	Output Enable Hold Time	Read	Min	0	ns
			Toggle and Data# Polling	Min	4	ns
	t_{OEZ}	Output Enable to High Z		Max	7.6	ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0	ns
	t_{PACC}	Intra Page Access Time		Max	20	ns
	t_{CEZ}	Chip Enable to High Z		Max	7.6	ns

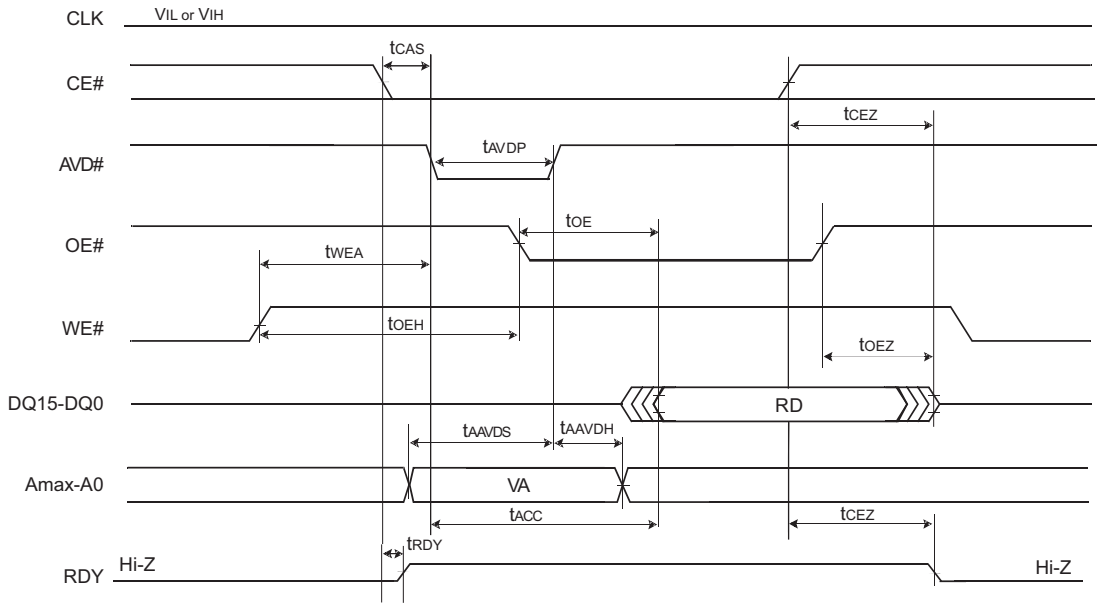
Figure 11.9 Asynchronous Read Mode (AVD# Toggling - Case 1)



Notes:

1. *Valid Address and AVD# Transition occur before CE# is driven Low.*
2. *VA = Valid Read Address, RD = Read Data.*

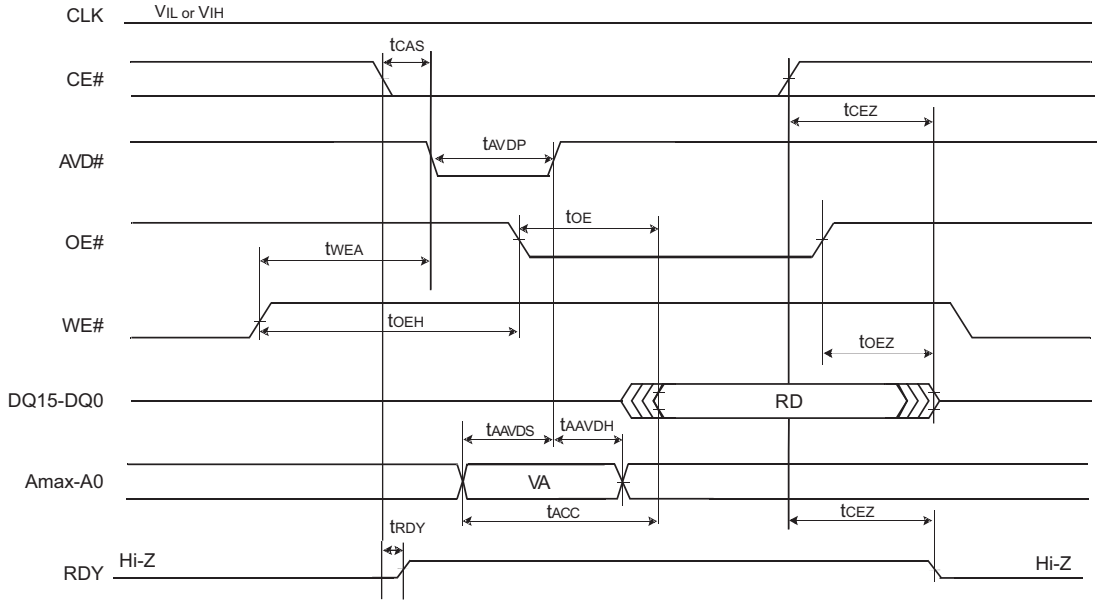
Figure 11.10 Asynchronous Read Mode (AVD# Toggling - Case 2)



Notes:

1. *AVD# Transition occurs after CE# is driven to Low and Valid Address Transition occurs before AVD# is driven to Low.*
2. *VA = Valid Read Address, RD = Read Data.*

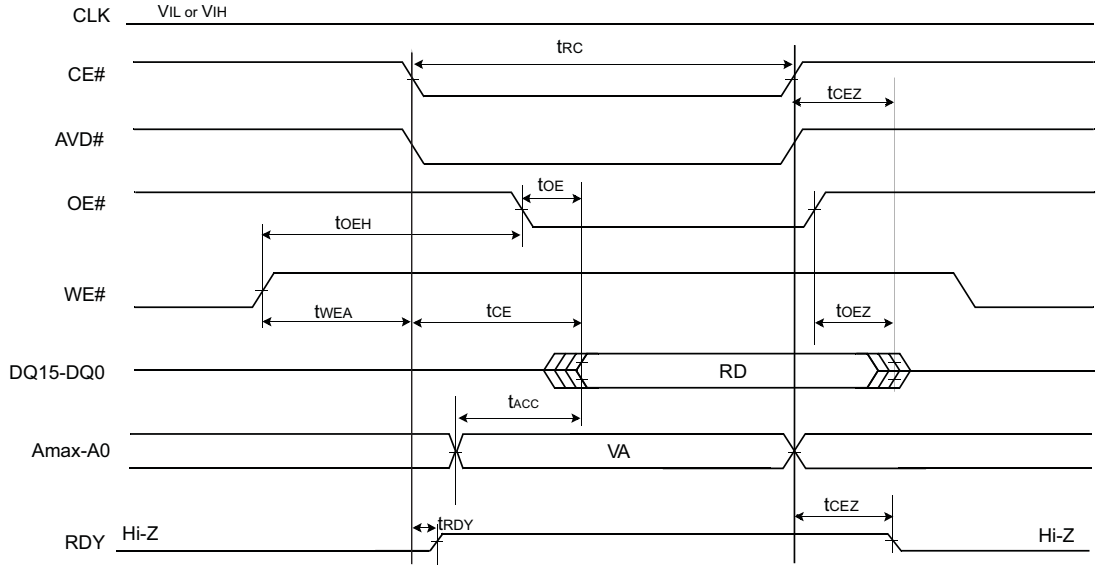
Figure 11.11 Asynchronous Read Mode (AVD# Toggling - Case 3)



Notes:

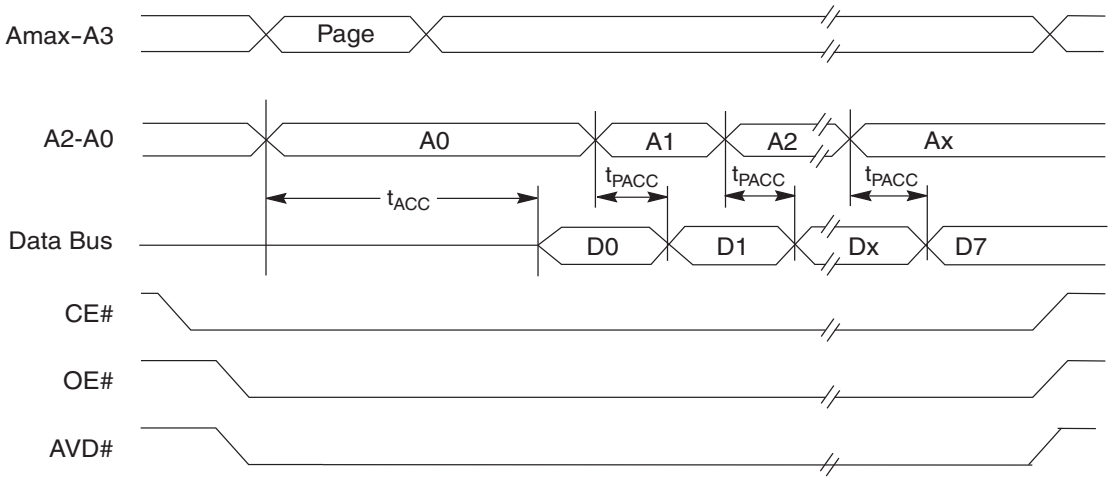
1. *AVD# Transition occurs after CE# is driven to Low and AVD# is driven low before Valid Address Transition.*
2. *VA = Valid Read Address, RD = Read Data.*

Figure 11.12 Asynchronous Read Mode (AVD# tied to CE#)



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

Figure 11.13 Asynchronous Page Mode Read



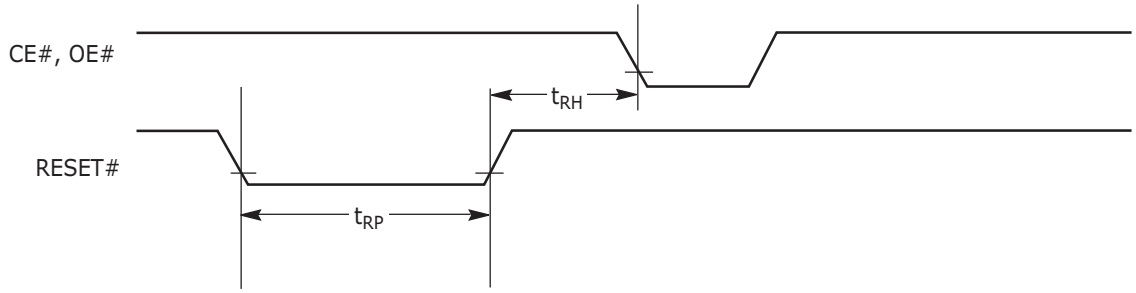
Note
RA = Read Address, RD = Read Data.

11.9.3 Hardware Reset (RESET#)

Hardware Reset

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

Figure 11.14 Reset Timings



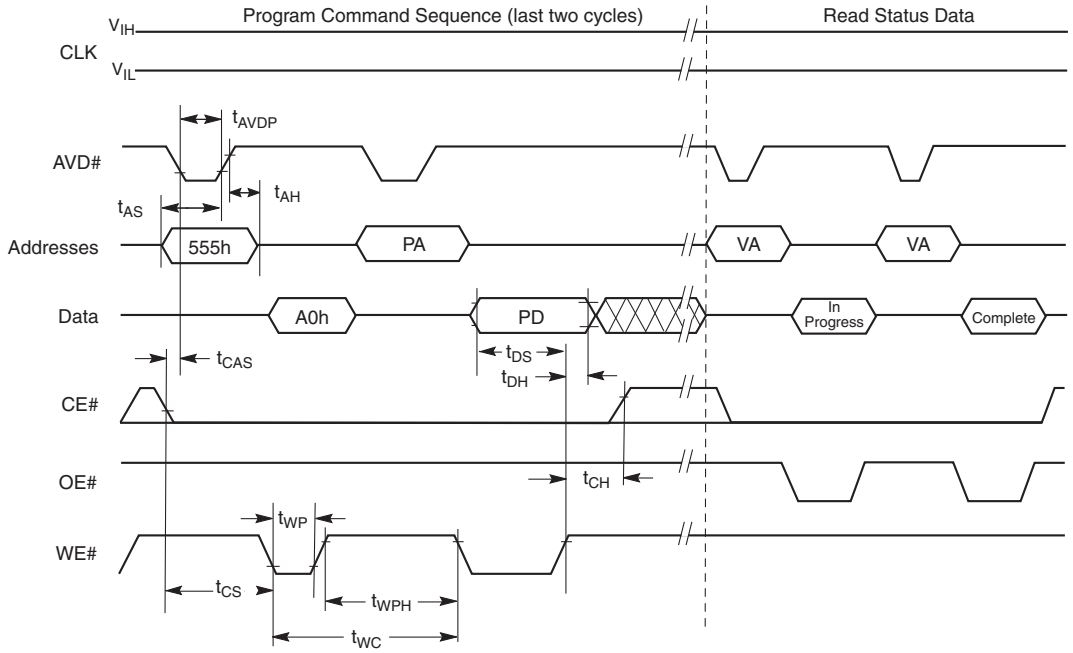
11.9.4 Erase/Program Timing

Parameter		Description			54 MHz	66 MHz	80 MHz	104 MHz	Unit
JEDEC	Standard								
t_{AVAV}	t_{WC}	Write Cycle Time (Note 1)		Min	60				ns
t_{AVWL}	t_{AS}	Address Setup Time (Note 2)	Synchronous	Min	5	5	5	3.5	ns
			Asynchronous		6	6	6	6	
t_{WLAX}	t_{AH}	Address Hold Time (Note 2)	Synchronous	Min	7	7	6	5	ns
			Asynchronous		7	7	6	5	
	t_{AVDP}	AVD# Low Time		Min	6				ns
t_{DVWH}	t_{DS}	Data Setup Time		Min	20				ns
t_{WHDX}	t_{DH}	Data Hold Time		Min	0				ns
t_{GHWL}	t_{GHWL}	Read Recovery Time Before Write		Min	0				ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0				ns
t_{WHEH}	t_{CH}	CE# Hold Time		Min	0				ns
t_{WLWH}	t_{WP}	Write Pulse Width		Min	25				ns
t_{WHWL}	t_{WPH}	Write Pulse Width High		Min	20				ns
	$t_{SR/W}$	Latency Between Read and Write Operations		Min	0				ns
	t_{VID}	V_{ACC} Rise and Fall Time		Min	500				ns
	t_{VIDS}	V_{ACC} Setup Time (During Accelerated Programming)		Min	1				µs
t_{ELWL}	t_{CS}	CE# Setup Time to WE#		Min	4				ns
	t_{AVSW}	AVD# Setup Time to WE#		Min	4				ns
	t_{AVHW}	AVD# Hold Time to WE#		Min	4				ns
	t_{AVSC}	AVD# Setup Time to CLK		Min	5	5	5	3	ns
	t_{AVHC}	AVD# Hold Time to CLK		Min	5	5	5	3	ns
	t_{SEA}	Sector Erase Accept Time-out		Min	50				µs

Parameter		Description		54 MHz	66 MHz	80 MHz	104 MHz	Unit
JEDEC	Standard							
	t_{ESL}	Erase Suspend Latency	Max	40				µs
	t_{PSL}	Program Suspend Latency	Max	40				µs
	t_{ASP}	Toggle Time During Erase within a Protected Sector	Typ	0				µs
	t_{PSP}	Toggle Time During Programming Within a Protected Sector	Typ	0				µs

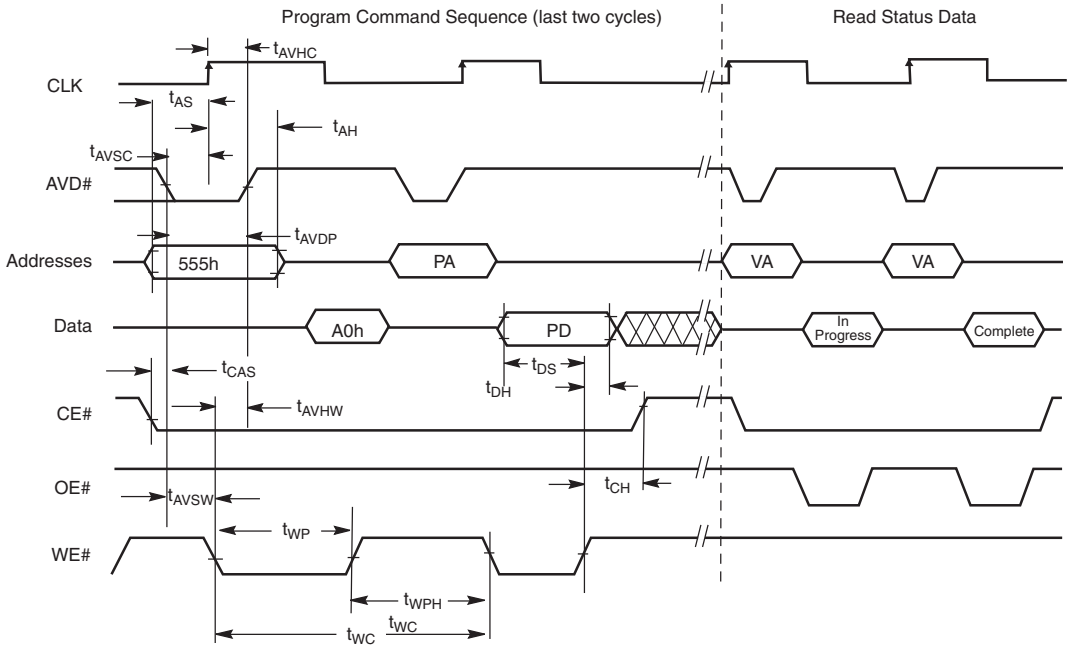
Notes
1. *Sampled, not 100% tested.*
2. *In programming operations, addresses are latched on the active edge of CLK for programming synchronously or rising edge of AVD# for programming asynchronously.*
3. *See the Erase and Programming Performance on page 80 section for more information. Does not include the preprogramming time.*

Figure 11.15 Asynchronous Program Operation Timings



Notes:
1. *PA = Program Address, PD = Program Data, VA = Valid Address for reading status bits.*
2. In progress *and* complete *refer to status of program operation.*
3. *CLK can be either V_{IL} or V_{IH}.*

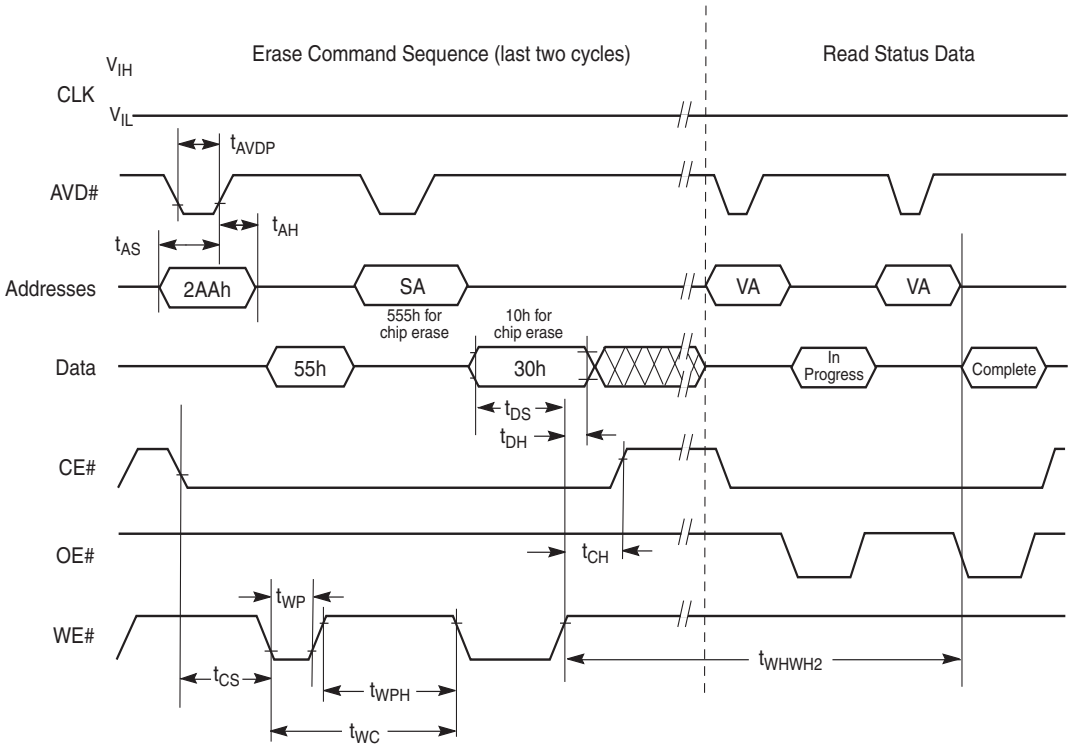
Figure 11.16 Synchronous Program Operation Timings



Notes:

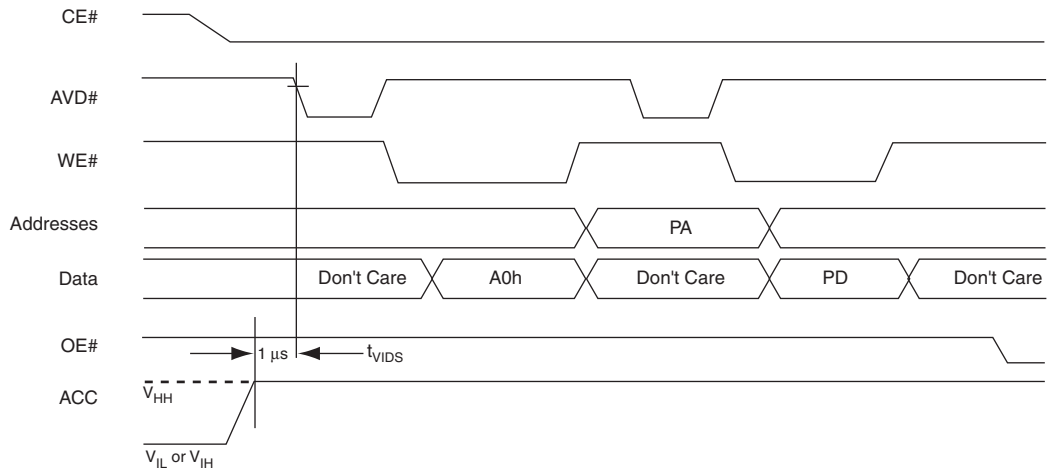
1. *PA = Program Address, PD = Program Data, VA = Valid Address for reading status bits.*
2. In progress *and* complete *refer to status of program operation.*
3. *Addresses are latched on the first rising edge of CLK.*

Figure 11.17 Chip/Sector Erase Command Sequence



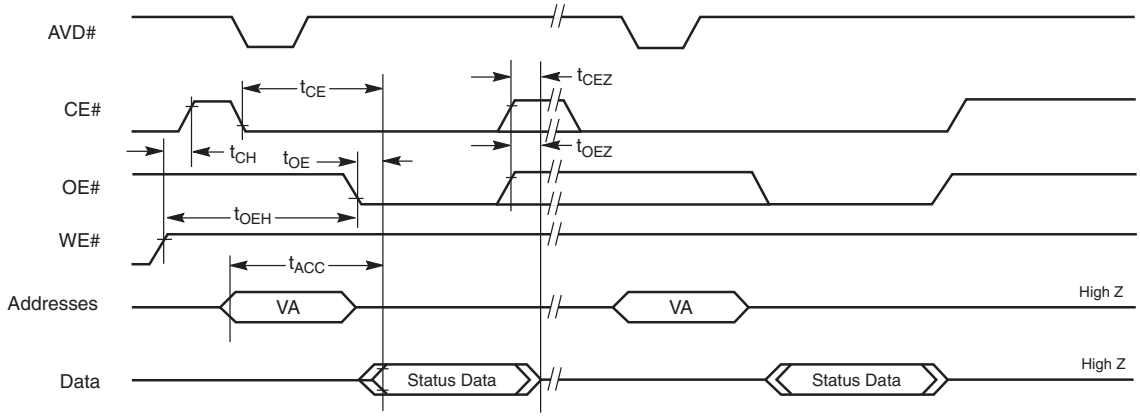
Note:
SA is the sector address for Sector Erase.

Figure 11.18 Accelerated Unlock Bypass Programming Timing



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

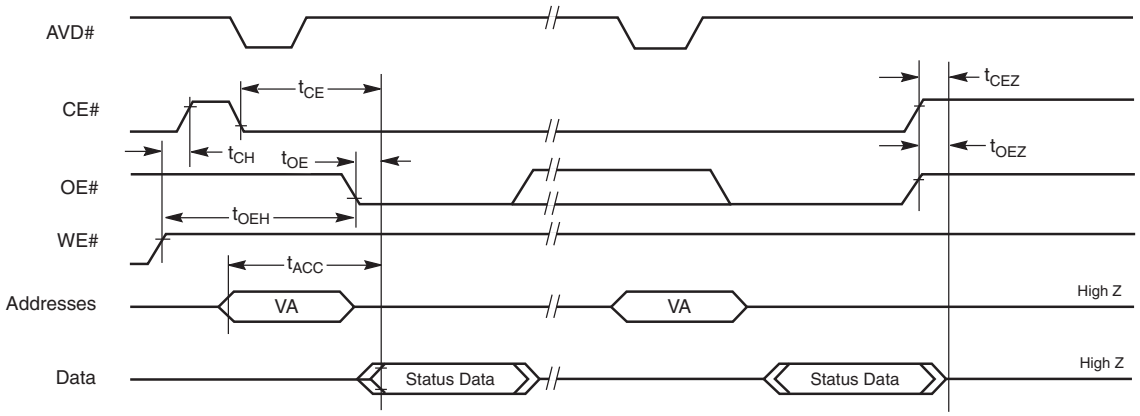
Figure 11.19 Data# Polling Timings (During Embedded Algorithm)



Notes:

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

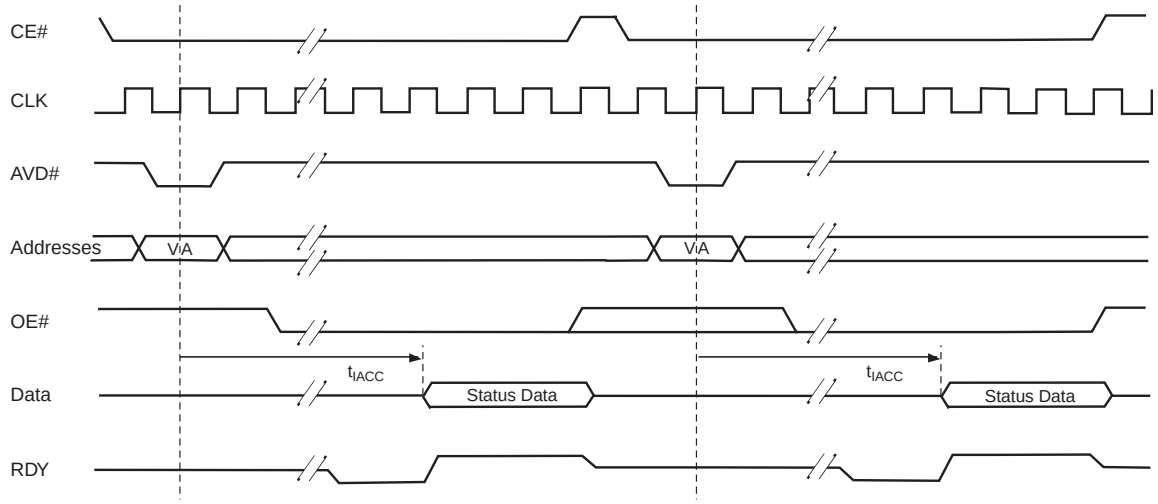
Figure 11.20 Toggle Bit Timings (During Embedded Algorithm)



Notes:

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

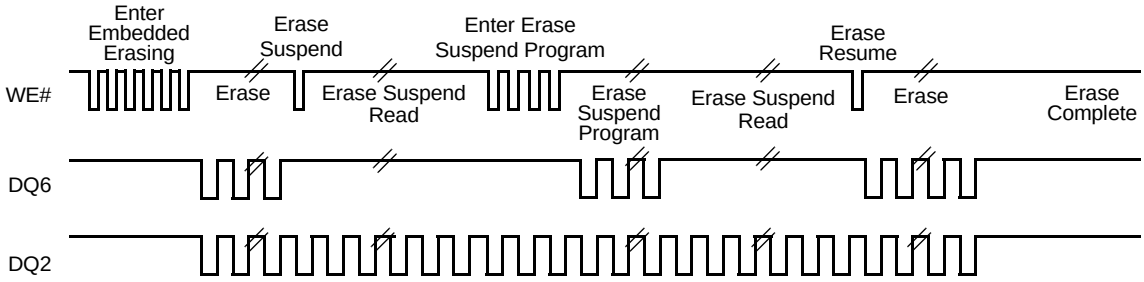
Figure 11.21 Synchronous Data Polling Timings/Toggle Bit Timings



Notes:

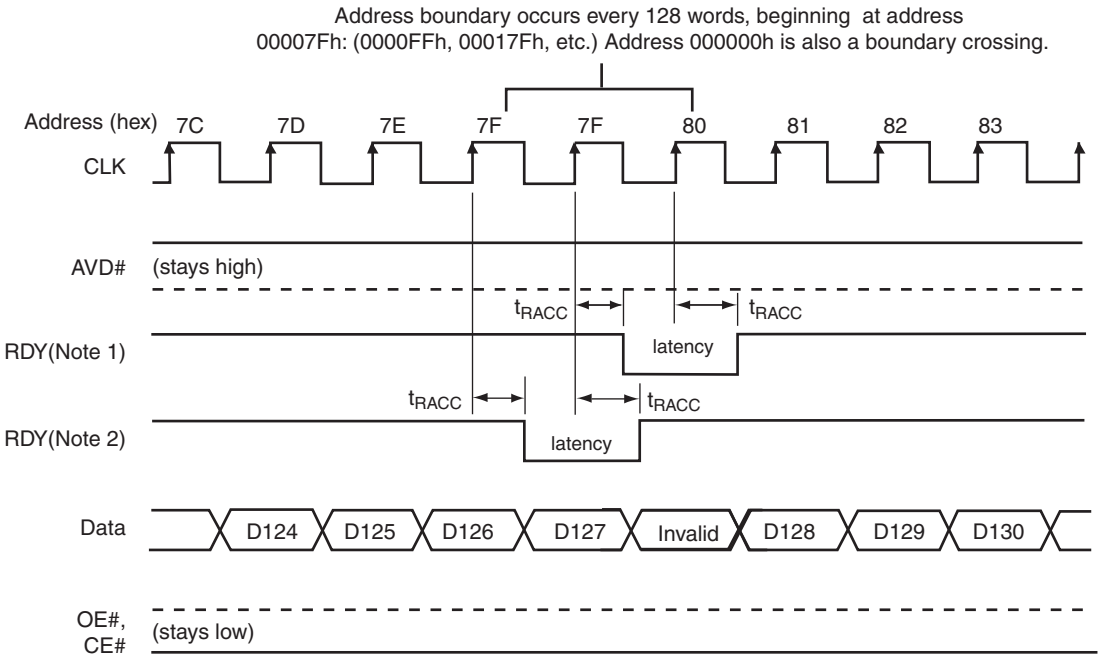
1. *The timings are similar to synchronous read timings.*
2. *VA = Valid Address. Two read cycles are required to determine status. When the Embedded Algorithm operation is complete, the toggle bits will stop toggling.*
3. *RDY is active with data (D8 = 0 in the Configuration Register). When D8 = 1 in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

Figure 11.22 DQ2 vs. DQ6



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

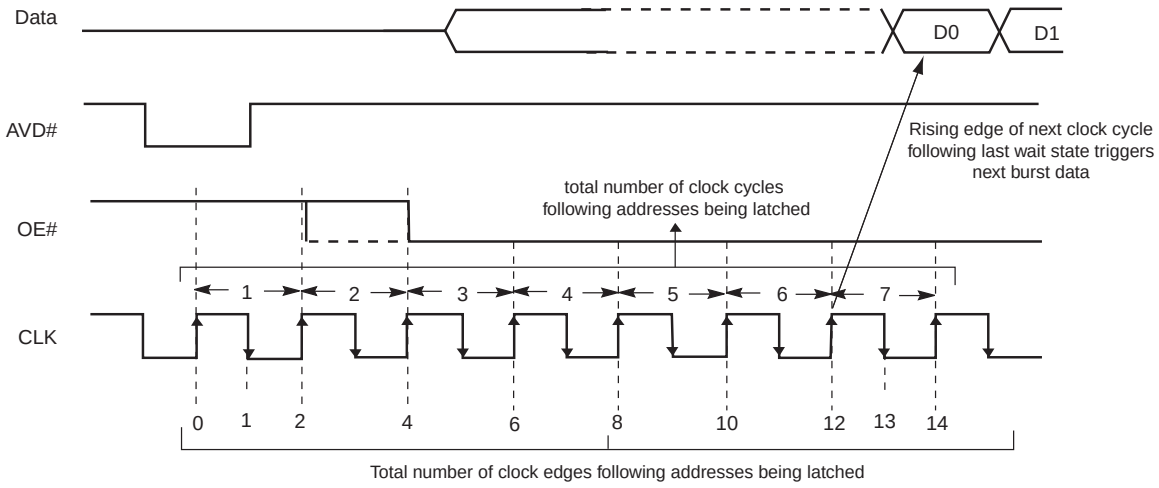
Figure 11.23 Latency with Boundary Crossing



Notes:

1. *RDY active with data (CR0.8 = 0 in the Configuration Register).*
2. *RDY active one clock cycle before data (CR0.8 = 1 in the Configuration Register).*
3. *Figure shows the device not crossing a bank in the process of performing an erase or program.*

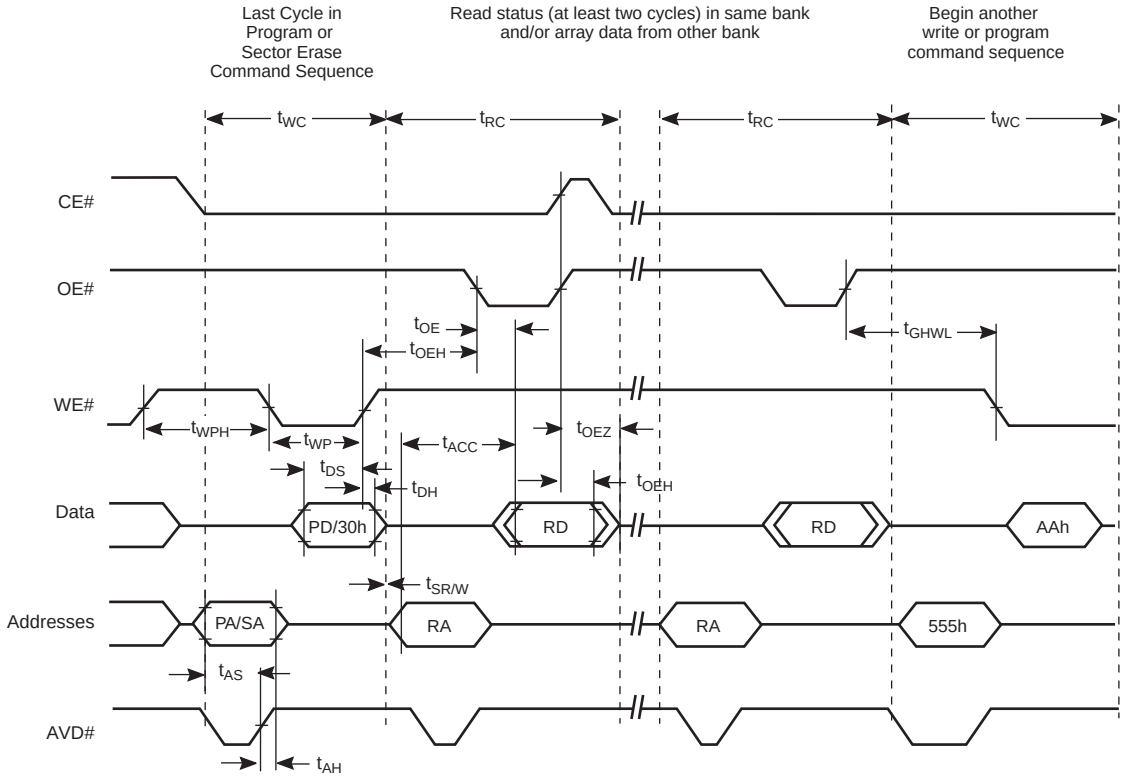
Figure 11.24 Wait State Configuration Register Setup



Example of Programmable Wait States

CR1.0	Programmable Wait State	0000 = initial data is valid on the 2rd rising CLK edge after addresses are latched
CR0.13		0001 =initial data is valid on the 3rd rising CLK edge after addresses are latched
CR0.12		0010 = initial data is valid on the 4th rising CLK edge after addresses are latched
CR0.11		0011 = initial data is valid on the 5th rising CLK edge after addresses are latched 0100 = initial data is valid on the 6th rising CLK edge after addresses are latched 0101 = initial data is valid on the 7th rising CLK edge after addresses are latched 0110 = Reserved 0111 = Reserved 1000 = initial data is valid on the 8th rising CLK edge after addresses are latched 1001 = initial data is valid on the 9th rising CLK edge after addresses are latched 101 1= initial data is valid on the 10th rising CLK edge after addresses are latched . . 1101 = Reserved 1110 = Reserved 1111 = Reserved

Figure 11.25 Back-to-Back Read/Write Cycle Timings



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

11.10 Erase and Programming Performance

Parameter			Typ (Note 1)	Max (Note 2)	Unit	Comments
Sector Erase Time	64 Kword	V_{CC}	0.6	3.0	s	Excludes 00h programming prior to erasure (Note 3)
	16 Kword	V_{CC}	0.35	1.75		
Chip Erase Time		V_{CC}	78.4 (WS128P) 155.2 (WS256P) 308.8 (WS512P)	154 (WS128P) 308 (WS256P) 616 (WS512P)	s	
Single Word Programming Time		V_{CC}	40	400	µs	Excludes system level overhead (Note 4)
		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		
		ACC	192	1920		
Chip Programming Time (using 32 word buffer)		V_{CC}	50.4 (WS128P) 100.8 (WS256P) 201.6 (WS512P)	157.3 (WS128P) 314.6 (WS256P) 1008 (WS512P)	s	Excludes system level overhead (Note 4)
		ACC	33.6 (WS128P) 67.2 (WS256P) 134.4 (WS512P)	100.7 (WS128P) 201.3 (WS256P) 402.6 (WS512P)		
Erase Suspend/Erase Resume (t_{ERS})			40		µs	
Program Suspend/Program Resume (t_{PRS})			40		µs	

Notes:

1. *Typical program and erase values are measured at T_C = 25°C, 1.8 V V_{CC}, 10,000 cycles using checkerboard patterns. Sampled, but not 100% tested.*
2. *Under worst case conditions of 90°C, V_{CC} = 1.70 V, 100,000 cycles.*
3. *In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 00h before erasure.*
4. *System-level overhead is the time required to execute the two- or four-bus-cycle sequence for the program command. See Table on page 81 and Table on page 83 for further information on command definitions.*

11.10.1 BGA Ball Capacitance

Parameter Symbol	Parameter Description	Test Setup	Typ	Max	Unit
C_{IN}	Input Capacitance	V_{IN} = 0	2	10	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0	2	10	pF

Notes

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

12. Appendix

This section contains information relating to software control or interfacing with the Flash device. For additional information and assistance regarding software, see *Additional Resources* *on page 8*, or explore the Web at www.Cypress.com.

Memory Array Commands

Command Sequence (Notes)		Cycles	Bus Cycles (Note 1 - 6)											
			First		Second		Third		Fourth		Fifth		Sixth	
			Addr	Data (19)	Addr	Data (19)	Addr	Data (19)	Addr	Data (19)	Addr	Data (19)	Addr	Data (19)
Asynchronous Read (7)		1	RA	RD										
Reset (8)		1	XXX	F0										
Autoselect (9)	Manufacturer ID	4	555	AA	2AA	55	(BA) 555	90	(BA) X00	0001				
	Device ID (10)	6	555	AA	2AA	55	(BA) 555	90	(BA) X01	227E	(BA) X0E	(10)	(BA) X0F	(10)
	Indicator Bits	4	555	AA	2AA	55	(BA) 555	90	(BA) X03	(12)				
	Sector Unlock/Lock Verify (11)	4	555	AA	2AA	55	(SA) 555	90	(SA) X02	0000/ 0001				
Single word		4	555	AA	2AA	55	555	A0	PA	Data				
Write Buffer to Flash Program (17)		6	555	AA	2AA	55	SA	25	SA	WC	PA (20)	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
Program/Erase Suspend (15)		1	BA	B0										
Program/Erase Resume (16)		1	BA	30										
Set Configuration Register (21)		5	555	AA	2AA	55	555	D0	X00	CR0	X01	CR1		
Read Configuration Register		4	555	AA	2AA	55	555	C6	X0 (0 or 1)	CR (0 or 1)				
CFI Query (17)		1	(BA) 55	98										
Unlock Bypass Mode	Unlock Bypass Entry (18)	3	555	AA	2AA	55	555	20						
	Unlock Bypass Program (13, 14)	2	XX	A0	PA	PD								
	Unlock Bypass Sector Erase (13, 14)	2	XX	80	SA	30								
	Unlock Bypass Erase (13, 14)	2	XX	80	XXX	10								
	Unlock Bypass CFI (13, 14)	1	XX	98										
	Unlock Bypass Reset	2	XX	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.

Notes

1. *See Table on page 13 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Except for the following, all bus cycles are write cycle: read cycle, fourth through sixth cycles of the Autoselect commands, fourth cycle of the configuration register verify and password verify commands, and any cycle reading at RD(0) and RD(1).*
4. *Data bits DQ15–DQ8 are don't care in command sequences, except for RD, PD, WD, PWD, and PWD3-PWD0.*
5. *Unless otherwise noted, address bits Amax–A14 are don't cares.*
6. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*
7. *No unlock or command cycles required when bank is reading array data.*
8. *The Reset command is required to return to reading array data (or to the erase-suspend-read mode if previously in Erase Suspend) when a bank is in the autoselect mode, or if DQ5 goes high (while the bank is providing status information) or performing sector lock/unlock.*
9. *The fourth cycle of the autoselect address is a read cycle. The system must provide the bank address.*
10. *(BA) + 0Eh ----> For WS128 = 2244h, WS256 = 2242h, WS512 = 223Dh. (BA) + 0Fh ----> For WS064/128/256/512 = 2200h*
11. *The data is 0000h for an unlocked sector and 0001h for a locked sector*
12. *See* Table , *Autoselect Addresses on page 30*.
13. *The Unlock Bypass command sequence is required prior to this command sequence.*
14. *The Unlock Bypass Reset command is required to return to reading array data when the bank is in the unlock bypass mode.*
15. *The system may read and program in non-erasing sectors, or enter the autoselect mode, when in the Erase Suspend mode. The Program/Erase Suspend command is valid only during a program/ erase operation, and requires the bank address.*
16. *The Program/Erase Resume command is valid only during the Program/Erase Suspend mode, and requires the bank address.*
17. *The total number of cycles in the command sequence is determined by the number of words written to the write buffer. The maximum number of cycles in the command sequence is 37.*
18. *Write Buffer Programming can be initiated after Unlock Bypass Entry.*
19. *Data is always output at the rising edge of clock.*
20. *Must be the lowest address.*
21. *Configuration Registers can not be programmed out of order. CR0 must be programmed prior to CR01 otherwise the configuration registers will retain their previous settings*

Sector Protection Commands (Sheet 1 of 2)

Command Sequence (Notes)		Cycles	Bus Cycles (Note 1 - 6)													
			First		Second		Third		Fourth		Fifth		Sixth		Seventh	
			Addr	Data (10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)
Secured Silicon Sector	Entry (5)	3	555	AA	2AA	55	555	88								
	Program	4	555	AA	2AA	55	555	A0	PA	PD						
	Read	1	SA	data												
	Exit (7)	4	555	AA	2AA	55	555	90	XX	00						
Lock Register	Register Command Set Entry (5)	3	555	AA	2AA	55	555	40								
	Register Bits Program (6)	2	XX	A0	00	data										
	Register Bits Read	1	00	data												
	Register Command Set Exit (7)	2	XX	90	XX	00										
Password	Protection Command Set Entry	3	555	AA	2AA	55	555	60								
	Program (9)	2	XX	A0	00/ 01/ 02/ 03	PW D0/ 1/ 2/ 3/										
	Read Password (10)	4	00	PW D0	01	PW D1	02	PW D2	03	PW D3						
	Unlock (9)	7	00	25	00	03	00	PW D0	01	PW D1	02	PW D2	03	PW D3	00	29
	Protection Command Set Exit	2	XX	90	XX	00										
PPB	Non-Volatile Sector Protection Command Set Entry (5)	3	555	AA	2AA	55	(BA) 555	C0								
	Program	2	XX	A0	(BA) SA	00										
	All Erase (8)	2	XX	80	XX	30										
	Status Read	1	(BA) SA	RD(0)												
	Non-Volatile Sector Protection Command Set Exit (7)	2	XX	90	XX	00										

Sector Protection Commands (Sheet 2 of 2)

Command Sequence (Notes)		Cycles	Bus Cycles (Note 1 - 6)													
			First		Second		Third		Fourth		Fifth		Sixth		Seventh	
			Addr	Data (10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)	Addr	Data ((10)
PPB Lock Bit	Global Volatile Sector Protection Freeze Command Set Entry (5)	3	555	AA	2AA	55	555	50								
	Set	2	XX	A0	XX	00										
	Status Read	1	XX	RD(0)												
	Global Volatile Sector Protection Freeze Command Set Exit (7)	2	XX	90	XX	00										
DYB	Volatile Sector Protection Command Set Entry (5)	3	555	AA	2AA	55	(BA) 555	E0								
	Set	2	XX	A0	(BA) SA	00										
	Clear	2	XX	A0	(BA) SA	01										
	Status Read	1	(BA) SA	RD(0)												
	Volatile Sector Protection Command Set Exit (7)	2	XX	90	XX	00										
Accelerated	Program	2	555	A0	PA	Data										
	Sector Erase	2	555	80	SA	30										
	Chip Erase	2	555	80	555	10										
	Asynchronous Read	1	RA	RD												
	Write to Buffer	4	SA	25	SA	WC	PA	PD	WBL	PD						
	Program Buffer to Flash	1	SA	29												

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: WS128P = A22–A14, WS256P = 23–A14
BA = Bank Address: WS128P = A22-A20, and A19; WS256P = A23-A20
CR = Configuration Register data bits D15–D0
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) = DQ0 protection indicator bit. If protected, DQ0 = 0, if unprotected, DQ0 = 1.
WBL = Write Buffer Location. Address must be within the same write buffer page as PA.
WC = Word Count. Number of write buffer locations to load minus 1.

Notes

1. *See Table for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Except for the following, all bus cycles are write cycle: read cycle, fourth through sixth cycles of the Autoselect commands, fourth cycle of the configuration register verify and password verify commands, and any cycle reading at RD(0) and RD(1).*
4. *Data bits DQ15–DQ8 are don't care in command sequences, except for RD, PD, WD, PWD, and PWD3-PWD0.*
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 aborts and returns the device to the default Persistent Sector Protection Mode during 2nd bus cycle. Note that on all future devices, addresses equal 00h, but is currently 77h for the WS512P only.*
7. *Exit command must be issued to reset the device into read mode; device may otherwise be placed in an unknown state.*
8. *"All PPB Erase" command pre-programs all PPBs before erasure to prevent over-erasure.*
9. *Entire two bus-cycle sequence must be entered for each portion of the password.*
10. *Full address range is required for reading password.*

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 software 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 – within that bank. All reads outside of the CFI address range, within the bank, returns non-valid data. Reads from other banks are allowed, writes are not. To terminate reading CFI data, the system must write the reset command.

The following is a C source code example of using the CFI Entry and Exit functions. Refer to the *Cypress Low Level Driver User's Guide* (available on www.Cypress.com) for general information on Cypress 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 JESD68.01). Please contact your sales office for copies of these documents.

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)

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	0005h	Typical Program Time per single word write 2^N µs (e.g. 30us)
20h	0009h	Typical Program Time using buffer 2^N µs (e.g. 300us) (00h = not supported)
21h	000Ah	Typical time for sector erase 2^N ms
22h	0000h	Typical time for full chip erase 2^N ms (00h = not supported)
23h	0003h	Max. Program Time per single word [2^N times typical value]
24h	0003h	Max. Program Time using buffer [2^N times typical value]
25h	0003h	Max. time for sector erase [2^N times typical value]
26h	0000h	Max. time for full chip erase [2^N times typical value] (00h = not supported)

Device Geometry Definition

Addresses	Data	Description
27h	0018h (WS128P) 0019h (WS256P) 001Ah (WS512P)	Device Size = 2^N byte
28h 29h	0001h 0000h	Flash Device Interface 0h=x8; 1h=x16; 2h=x8/x16; 3h=x32 [lower byte] [upper byte] (00h = not supported)
2Ah 2Bh	0006h 0000h	Max. number of bytes in multi-byte buffer write = 2^N [lower byte] [upper byte] (00h = not supported)
2Ch	0003h	Number of Erase Block Regions within device 01h = Uniform Sector; 02h = Boot + Uniform; 03h = Boot + Uniform + Boot
2Dh 2Eh 2Fh 30h	0003h 0000h 0080h 0000h	Erase Block Region 1 Information (Small Sector Section) [lower byte] - Number of sectors. 00h=1 sector; 01h=2 sectors ... 03h=4 sectors [upper byte] [lower byte] - Equation =>(n = Density in Bytes of any 1 sector/256)h [upper byte]
31h 32h 33h 34h	007Dh (WS128P) 00FDh (WS256P) 00FDh (WS512P) 0001h 0000h 0002h	Erase Block Region 2 Information (Large Sector Section) [lower byte] - Number of sectors. [upper byte] [lower byte] - Equation =>(n = Density in Bytes of any 1 sector/256)h [upper byte]
35h 36h 37h 38h	0003h 0000h 0080h 0000h	Erase Block Region 3 Information (Small Sector Section) [lower byte] - Number of sectors. 00h=1 sector; 01h=2 sectors ... 03h=4 sectors [upper byte] [lower byte] - Equation =>(n = Density in Bytes of any 1 sector/256)h [upper byte]
39h 3Ah 3Bh 3Ch	0000h 0000h 0000h 0000h	Erase Block Region 4 Information

Primary Vendor-Specific Extended Query (Sheet 1 of 2)

Addresses	Data	Description
40h 41h 42h	0050h 0052h 0049h	Query-unique ASCII string *PRI*
43h	0031h	Major CFI version number, ASCII
44h	0034h	Minor CFI version number, ASCII
45h	0101b	Address Sensitive Unlock (Bits 1-0) 00b = Required, 01b = Not Required Silicon Technology (Bits 5-2) 0011b = 130nm; 0100b = 110nm; 0101b = 90nm 001010b = 000Ah
46h	0002h	Erase Suspend 0 = Not Supported, 1 = To Read Only, 2 = To Read & Write
47h	0001h	Sector Protection per Group 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 08h = Advanced Sector Protection; 07h = New Sector Protection Scheme
4Ah	07Bh (WS128P) 0F3h (WS256P) 1E3h (WS512P)	Simultaneous Operation Number of Sectors in all banks except boot bank
4Bh	0001h	Burst Mode Type 00 = Not Supported, 01 = Supported
4Ch	0002h	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	Write Protect Function 00h = No Boot, 01h = Dual Boot, 02h = Bottom Boot, 03h = Top Boot
50h	0001h	Program Suspend. 00h = not supported
51h	0001h	Unlock Bypass 00 = Not Supported, 01=Supported
52h	0008h	Secured Silicon 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 (e.g. 10us => n=14)
54h	0014h	Hardware Reset Low Time-out not during an embedded algorithm to read mode Maximum 2^N ns (e.g. 10us => n=14)
55h	0005h	Erase Suspend Time-out Maximum 2^N µs
56h	0005h	Program Suspend Time-out Maximum 2^N µs
57h	0010h	Bank Organization: X = Number of banks
58h	0007h (WS064P) 000Bh (WS128P) 0013h (WS256P) 0023h (WS512P)	Bank 0 Region Information. X = Number of sectors in bank

Primary Vendor-Specific Extended Query (Sheet 2 of 2)

Addresses	Data	Description
59h	0004h (WS064P) 0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 1 Region Information. X = Number of sectors in bank
5Ah	0004h (WS064P) 0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 2 Region Information. X = Number of sectors in bank
5Bh	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 3 Region Information. X = Number of sectors in bank
5Ch	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 4 Region Information. X = Number of sectors in bank
5Dh	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 5 Region Information. X = Number of sectors in bank
5Eh	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 6 Region Information. X = Number of sectors in bank
5Fh	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 7 Region Information. X = Number of sectors in bank
60h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 8 Region Information. X = Number of sectors in bank
61h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 9 Region Information. X = Number of sectors in bank
62h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 10 Region Information. X = Number of sectors in bank
63h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 11 Region Information. X = Number of sectors in bank
64h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 12 Region Information. X = Number of sectors in bank
65h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 13 Region Information. X = Number of sectors in bank
66h	0008h (WS128P) 0010h (WS256P) 0020h (WS512P)	Bank 14 Region Information. X = Number of sectors in bank
67h	000Bh (WS128P) 0013h (WS256P) 0023h (WS512P)	Bank 15 Region Information. X = Number of sectors in bank

13. Revision History

Document Title:S29WS512P, S29WS256P, S29WS128P 512/256/128 Mb (32/16/8 M x 16 bit), 1.8 V, Simultaneous Read/Write Flash Document Number: 002-01747				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	-	WIOB	11/03/2006	**Spansion Publication Number:** S30MS02GR_SP1_QDB **A6:Features**: Removed Zero Hold mode **Switching Waveforms**: Revised VCC Power-up diagram **Timing Diagrams**: Changed tCR to tRDY in figure 11.7 and figure 11.8
			11/08/2006	**A7:Features:** Updated Effective Write Buffer Programming Per Word **Erase/Program Timing:** tESL changed to Max tPSL changed to Max **CMOS Compatible:** Removed Note 2 from table.
			03/09/2007	**A8:Asynchronous Mode Read**: Changed tCR to tRDY in figures 11.9 through 11.12 **Common Flash Memory Interface**: Revised Device Geometry table: Changed WS512P data to 00FDh Address 32h - Data changed to 001h Address 33h - Data changed to 000h Address 34h - Data changed to 002h Revised CFI table: removed Uniform Bottom, Uniform Top, and All sectors for Address 4Fh **DC Characteristics**: Revised ICCB Burst table
			03/27/2007	**A9:DC Characteristics**: Revised ICCB for 108 MHz frequencies to TBA **Synchronous/Burst Read**: Revised tRACC to 7.6 ns **Asynchronous Mode Read**: Revised tAAVDH to 4 ns
			04/20/2007	**A10:AC Characteristics:** Removed wait state below 14 MHz, wait state 2 Added additional wait state to all wait state frequency in table 11.4 Added Continuous Burst Mode Synchronous Wait State Requirement table Revised Burst Access Time to (WS-1) * tCK + (tBACC)
			09/28/2007	**A11:Data Sheet Status:** Changed to Production **Global:** Changed all 108 MHz to 104 MHz **Latency:** Added 10 wait state and 11 wait state latency tables **Configuration Registers:** Added two more configurations to CR0.11 for 10th and 11th rising CLK edge **AC Characteristics**: Revised tCES to 6 ns Revised tAVD to tCLK **DC Characteristics:** Changed description of ICC2 to VCC Active Program/Erase Current Change descritpion of ICC5 to VCC Active Current (Read while Program/Erase) **Erase/Program Timing and Performance:** Revised: tERS to 40 µs tESL to 40 µs tPSL to 40 µs tPRS to 40 µs **Output Slew Rate:** Deleted Programmable Outuput Slew Rate Control section
			01/28/2008	**A12:Configuration Registers:** Changed CR0.14 default setting to 1 **AC Characteristics:** Added device Vcc ramp rate limit. Updated timing diagrams for Synchronous/Burst Read, Asynchronous Program Operation, Synchronous Program Operation, and Chip Sector Erase Command Sequence. **Program/Erase Operations:** Added details to Program and Erase Suspend/Resume operations
*A	5046533	WIOB	12/14/2015	Updated to Cypress Template
*B	5790689	NIBK	06/29/2017	Updated to Cypress Logo and Copyright.

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

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

Products

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

PSoC®Solutions

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

Cypress Developer Community

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

Technical Support

cypress.com/support

© Cypress Semiconductor Corporation, 2006-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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

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

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View S29WS256P0PBFW000 on WIN SOURCE
- Cypress Semiconductor Corp Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management