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

Continuity of Specifications

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

Continuity of Ordering Part Numbers

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

For More Information

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

S29WS512R, S29WS256R, S29WS128R

512/256/128 Mb (32/16/8M x 16 bit) 1.8 V S29WS-R MirrorBit® Flash

This product family has been retired and is not recommended for designs. For new and current designs, S29WS512P, S29WS256P, and S29WS128P supersedes S29WS512R, S29WS256R, and S29WS128R respectively. This is the factory-recommended migration path. Please refer to the S29WS-P data sheet for specifications and ordering information.

Availability of this document is retained for reference and historical purposes only.

Features

- 65 nm MirrorBit Technology
- Single supply 1.8 V read/program/erase (1.70 V – 1.95 V)
- Wireless Temperature range (-25°C to +85°C)
- 16-bit (Word) data bus width
- Simultaneous Read/Write (SRW) operation
  - Read from one bank while programming or erasing in another bank
  - Memory array is divided into 16 equal size banks
- Programmable linear (8/16) with wrap around and continuos burst read modes
- RDY output for data transfer flow control
- Sector Erase
  - Four 32 Kbyte sectors at top or bottom of memory array
  - All other sectors are 128 Kbytes
- Write Buffer Programming up to 64 -byte groups
- Optional acceleration voltage supply (ACC) to reduce factory programming time
- Suspend and Resume commands for Program and Erase operations
- Write operation status register bits indicate program and erase operation completion
- Program-Erase Endurance
  - 100,000 cycles per sector (typical)
  - 10-year data retention (typical)
- Data Protection
  - Low V_{CC} write inhibit
- Secured Silicon Sector
  - 512 Bytes of Secured Silicon Sector region consisting of One Time Program (OTP) area of 256 bytes each for factory and customer
  - Hardware Sector Protection (via ACC pin)
    - All sectors protected when ACC input is at V_{IL}
  - Boot code controlled sector protection
    - A range of sectors may be protected to prevent program and erase until the next hardware reset or power is removed from the device
  - Dynamic sector protection
    - All sectors are unprotected at power on for simplified system production test & programming
    - A single command is used to protect all sectors from program or erase
    - A single sector at a time may be unprotected by a command to enable programming or erase.
- Common Flash Interface (CFI) data structure
- Offered Packages
  - 512/256/128R: 84-ball FBGA (11.6mm x 8mm) VBH084

Performance Characteristics

Read Access Times (maximum values)	
Speed Option (MHz)	**104**
Synch. Internal Access, ns (t_{IA})	75
Synch. Burst Access, ns (t_{BACC})	7.6
Asynch. Access Time, ns (t_{ACC})	80

Current Consumption (typical values)	
Burst Read @ 104 MHz (I_{CCB})	32 mA
Simultaneous Operation @ 104 MHz (I_{CC5})	52 mA
Program (I_{CC2})	20 mA
Erase (I_{CC2})	20 mA
Standby Mode (I_{CC3})	20 µA

Typical Program & Erase Times (typical values)	
Effective Write Buffer Programming (V_{CC}) Per Word	12.5 µs
Effective Write Buffer Programming (V_{ACC}) Per Word	8 µs
Sector Erase (32 KByte Sector) (V_{CC})	0.35 s
Sector Erase (128 KByte Sector) (V_{CC})	0.8 s

Contents

1. General Description

The Spansion S29WS512/256/128R are Mirrorbit flash products fabricated on 65 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.

2. Ordering Information

This product family has been retired and is not recommended for designs. For new and current designs, S29WS512P, S29WS256P, and S29WS128P supersedes S29WS512R, S29WS256R, and S29WS128R respectively. This is the factory-recommended migration path. Please refer to the S29WS-P data sheet for specifications and ordering information.

Availability of this document is retained for reference and historical purposes only.

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

S29WS 512 R 0P BH W 00 0

Packing Type
0 = Tray (standard; see note 1)
2 = 7-inch Tape and Reel
3 = 13-inch Tape and Reel

Model Number
(Boot Configuration Options)
00 = Uniform, 1 CE
20 = Top Boot, 1 CE

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

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

Speed Option (Burst Frequency)
0P = 66 MHz
0S = 83 MHz
AA = 104 MHz

Process Technology
R = 65 nm MirrorBit Technology

Flash Density
512 =512 Mb
256 =256 Mb
128 =128 Mb

Device Family
S29WS =1.8 Volt-only Simultaneous Read/Write, Burst Mode Flash Memory

2.1 Valid Combinations

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

S29WS-R Valid Combinations (Notes 1, 2)				Model Numbers	Package Type (Note 2)	Product Status
Base Ordering Part Number	Speed Option	Package Type, Material, & Temperature Range	Packing Type			
S29WS512R	0P, 0S, AA	BHW (Low-Halogen, Lead (Pb)-free)	0, 2, 3 (Note 1)	00, 20	11.6 mm x 8 mm 84-ball MCP-Compatible	Advance
S29WS256R						
S29WS128R						

Notes:
1. *Type 0 is standard. Specify other options as required.*
2. *BGA package marking omits leading* S29 *and packing type designator from ordering part number.*

3. Input/Output Descriptions & Logic Symbol

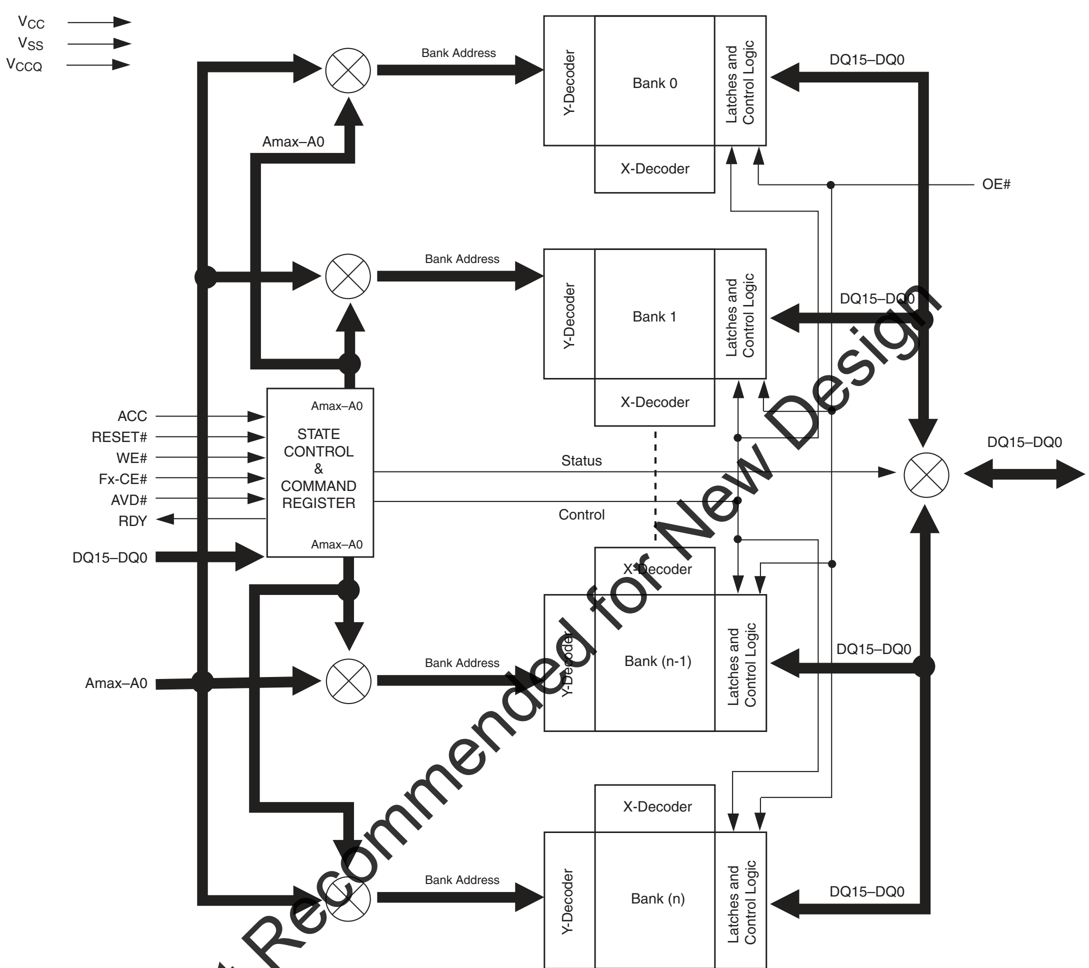
Table identifies the input and output package connections provided on the device.

Table 3.1 Input/Output Descriptions

Symbol	Type	Description
Amax – A0	Input	Higher order address lines. Amax = A24 for WS512R, A23 for WS256R, A22 for WS128R
DQ15 – DQ0	I/O	Data input/output
F1-CE#	Input	Flash-1 Chip Enable. Asynchronous relative to CLK. Used to select the first portion of the flash device address space that can be directly selected by one host chip enable signal.
F2-CE#	Input	Flash-2 Chip Enable. Asynchronous relative to CLK. Used to select the first portion of the flash device address space that can be directly selected by one host chip enable signal.
OE#	Input	Output Enable. Asynchronous relative to CLK for the Burst mode
WE#	Input	Write Enable
V_{CC}	Supply	Device Power Supply
V_{CCQ}	Supply	Input/Output Power Supply (must be ramped simultaneously with V_{CC})
V_{SS}	Supply	Ground
NC	No Connect	No Connected internally
RDY	Output	Ready. Indicates when valid burst data is ready to be read
CLK	Input	The first rising edge of CLK in conjunction with AVD# low latches address input and activates burst mode operation. After the initial word is output, subsequent rising edges of CLK increment the internal address counter. CLK should remain low during asynchronous access
AVD#	Input	Address Valid input. Indicates to device that the valid address is present on the address inputs. V_{IL} = for asynchronous mode, indicates valid address; for burst mode, cause staring address to be latched on rising edge of CLK. V_{IH} = device ignores address inputs
RESET#	Input	Hardware Reset. Low = device resets and returns to reading array data.
ACC	Input	Accelerated input. At V_{IL},disables all program and erase functions. Should be at V_{IH} for all other conditions.
RFU	Reserved	Reserved for future use

4. Block Diagrams

Figure 4.1 Simultaneous Operation Circuit



Notes:
1. *Amax = A24 for WS512R, A23 for WS256R, A22 for WS128R.*
2. *n = 15.*

5. Physical Dimensions/Connection Diagrams

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

5.1 Related Documents

The following documents contain information related to this family of flash devices. Click on the title or go to www.spansion.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

5.2 Special Handling Instructions for FBGA Package

Special handling is required for flash memory products in FBGA packages.

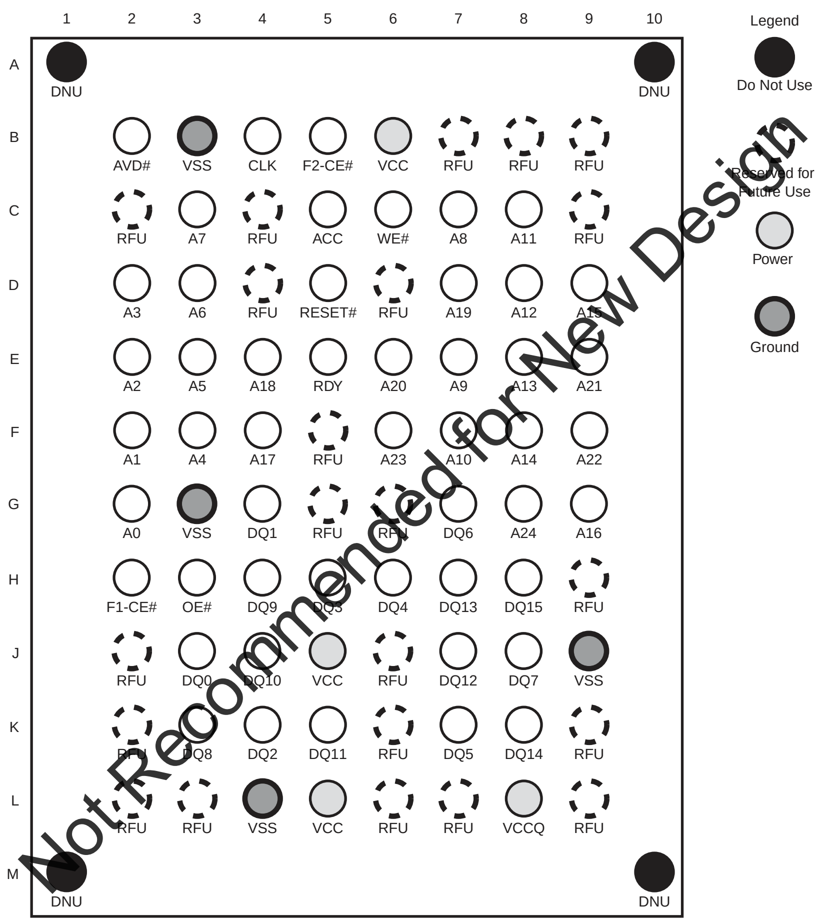
Flash memory devices in FBGA packages may be damaged if exposed to ultrasonic cleaning methods. The package and/or data integrity may be compromised if the package body is exposed to temperatures above 150°C for prolonged periods of time.

5.3 Connection Diagrams and Physical Dimensions

Figure 5.1 84-ball Fine-Pitch Ball Grid Array

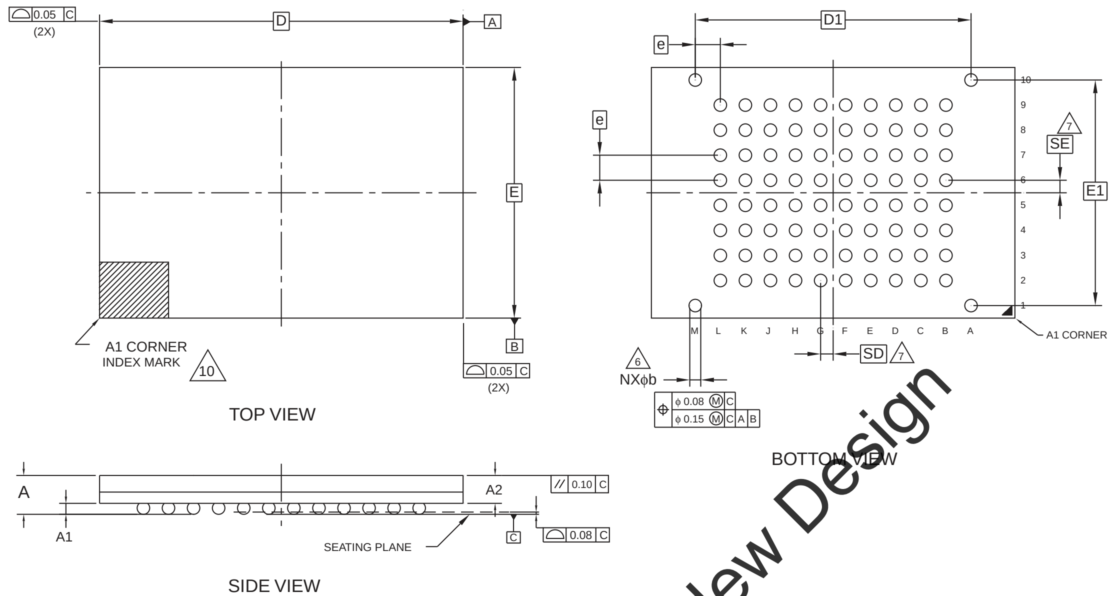
WS512R/WS256R/WS128R

(Top View, Balls Facing Down, MCP Compatible)



Notes
1. *Ball G8 is RFU on the WS256R.*
2. *Ball G8 is A24 on densities ≥ 256 Mbit.*
3. *Address signals numbered greater than Amax of a particular device are reserved for future use and only indicate where the higher order address will be in higher density members of related or future family devices.*

VBH084—84-ball Fine-Pitch Ball Grid Array (FBGA) 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:
1. *BSC is an ANSI standard for Basic Space Centering.*

6. Product Overview

The S29WS-R family consists of 128 Mbit to 1 Gbit, 1.8-V only, simultaneous read/write, burst-mode, flash devices. These devices have a 16 bit (word) wide data bus. All read accesses provide 16 bits of data on each bus transfer cycle. All writes take 16 bits of data from each bus transfer cycle.

Device	Mbits	Mbytes	Mwords	Banks	Mbytes / Bank
S29WS128R	128	16	8	16	1
S29WS256R	256	32	16	16	2
S29WS512R	512	64	32	16	4

The flash memory array is divided into banks as shown in the above table. A bank is the address range within which one program, or erase operation may be in progress at the same time as one read operation is in progress in any other bank of the memory. This multiple bank structure enables Simultaneous Read and Write (SRW) so that code may be executed or data read from one bank while a group of data is programmed, or erased as a background task in one other bank.

Each bank is divided into sectors. A sector is the minimum address range of data which can be erased to an all Ones state. There are four 32 Kbyte sectors which are located either at the bottom or top of the memory array depending on the device model purchased. These are called boot sectors because they are often used for holding boot code or parameters that need to be protected or erased separately from other data in the flash array. All other sectors are a uniform size of 128 Kbytes.

Programming is done via a 64 Byte write buffer. It is possible to program from one to 32 words (64 bytes) in each programming operation.

The S29WS family is capable of continuous, synchronous burst read or linear read (8- or 16-word aligned group) with wrap around. A wrapped burst begins at the initial location and continues to the end of an 8- or 16-word aligned group then “wraps-around” to continue at the beginning of the 8, or 16-word aligned group. The burst completes with the last word before the initial location. Word wrap around burst is generally used for processor cache line fill.

7. Address Space Maps

There are five address spaces within each device:

- A Non-Volatile Flash Memory Array used for storage of data that may be randomly read and reprogrammed
- A Read Only Memory Array used for factory programmed permanent device characteristics information. This area contains the Device Identification (ID) and Common Flash Interface (CFI) information.
- A One Time Programmable (OTP) Non-volatile flash array used for factory programmed permanent data, and customer programmable permanent data. This is called the Secure Silicon Region (SSR).
- An OTP location used to permanently protect the SSR. This is call the SSR Lock.
- A volatile register used to configure device behavior options. This is called the Configuration Register.

The main Flash Memory Array is the primary and default address space but, it may be partially overlaid by the other four address spaces with one alternate address space available at any one time. The location where the alternate address space is overlaid is defined by the address provided in the command that enables each overlay. The portion of the command address that is sufficient to select a sector is used to select the sector that is overlaid by an alternate Address Space Overlay (ASO).

Any address range, within the overlaid sector, not defined by an overlay address map, is reserved for future use. All read accesses outside of an address map within the selected sector, return non-valid data. The locations will display actively driven data but the meaning of whatever ones or zeros appear are not defined.

There are three operation modes for each bank that determine what portions of the address space are readable at any given time:

- Read Mode
- Embedded Algorithm (EA) Mode
- Address Space Overlay (ASO) Mode

Each bank of the device can be in any operation mode but, only one bank can be in EA or ASO mode at any one time.

In Read Mode a Flash Memory Array bank may read directly by asynchronous or synchronous accesses from the host system bus. The Flash Control Unit (CU) puts all banks in Read mode during Power-on, a Hardware Reset, after a Command Reset, or after a bank is returned to Read mode from EA mode. A bank with a suspended EA is considered to be returned to Read mode even though some or all of the data in the bank may be in an invalid state and thus not useful if read.

In EA mode the flash memory array data in a bank is stable but undefined, and effectively unavailable for read access from the host system. While in EA mode the bank is used by the CU in the execution of commands. Typical EA mode operations are programming or erasing of data in the flash array. All other banks are available for read access while the one bank is in EA mode. This ability to read from one bank while another bank is used in the execution of a command is called Simultaneous Read and Write (SRW) and allows for continued operation of the system via the reading of data or execution of code from other banks while one bank is programming or erasing data as a relatively long time frame background task.

In ASO mode, one of the overlay address spaces are overlaid in a bank (entered). That bank is in ASO mode and no other bank may be in EA or ASO mode. All EA activity must be completed before entering any ASO mode. A command for entering an EA or ASO mode while another bank is in EA or ASO mode will be ignored.

While an ASO mode is active (entered) in a bank, a read for flash array data to any other bank is allowed. ASO mode selects a specific sector for the overlaid address space. Other sectors in the ASO bank still provide flash array data and may be read during ASO mode.

The ASOs are functionally tied to the lowest address bank. The commands used to overlay (enter) these areas must select a sector address within the lowest address bank.

While SSR Lock, SSR, or Configuration Register is overlaid only the SSR Lock, SSR, or Configuration Register respectively may be programmed in the overlaid sector. While any of these ASO areas are being programmed the ASO bank switches to EA mode. The ID/CFI and factory portion of the SSR ASO is not customer programmable.

The address nomenclature used in this document is a shorthand form that shows addresses are formed from a concatenation of high order bits, sufficient to select a Sector Address (SA), with low order bits to select a location within the sector. When in Read mode and reading from the flash array the entire address is used to select a specific word for asynchronous read or the starting word address of a burst read. When writing a command, the address bits between SA and the command specified least significant bits must be Zero to allow for future extension of an overlay address map.

7.1 Data Address & Quantity Nomenclature

A **Bit** is a single One or Zero data value. A **Byte** is a group of 8 bits aligned on an 8 bit address boundary. A **Word** is a group of 16 bits aligned on a 16 bit address boundary. A Kbyte (KB) is 1024 Bytes (not 1000 Bytes).

Throughout this document **quantities of data are generally expressed in terms of bytes**. Example: most sectors have 128 Kbytes of data and is written as 128 Kbytes or 128 KB. **Addresses are also expressed in byte units.** A 128 Kbyte sector has an address range from 00000h to 1FFFFh Byte locations. Byte units are used because most host systems and software for these systems use byte resolution addresses. Software & hardware developers most often calculate code and data sizes in terms of bytes, so this is more familiar terminology than describing data sizes in bits or words. In general, data units will not be abbreviated if possible so that full unit names of Byte, Word, or bit are used. However, there may be cases where capital B is used for byte units and lower case b is used for bit units, in situations where space is limited such as in table column headers.

In some cases data quantities will also be expressed in word or bit units in addition to the quantity shown in bytes. This may be done as an aid to readers familiar with prior device generation documentation which often provided only word or bit unit values. Word units may also be used to emphasize that, in the memory devices described in this documentation, data is always exchanged with the host system in word units. Each bus cycle transfer of read or write data on the host system bus is a transfer 16 bits of data. A read bus cycle is always a 16 bit wide transfer of data to the host system whether the host system chooses to look at all the bits or not. A write bus cycle is always a transfer of 16 bits to the memory device and the device will store all 16 bits to a register. In the case of a program operation all 16 bits of each word to be programmed will be stored in the flash array.

Because data is always transferred in word units, the memory devices being discussed use only the address signals from the system necessary to select words. Most host systems use address line A0 to select bytes and a1 to select words. Flash memories with word wide data paths have traditionally started their address signal numbering with A0 being the selector for words because a byte select input is not needed. So, system address a[max] to a1 are connected to flash A[max] to A0.

In prior generation flash documentation, address values used in commands to the flash were documented from the viewpoint of the flash device - the bit pattern appearing on flash address inputs A10 to A0. However, most software is written with addresses expressed in bytes. This means the address patterns shown in flash command tables have traditionally been shifted by one bit to express them as byte address values in flash control programs. Example: a prior generation flash data sheet would show a command write of data value xxA0h to address 555h; this is an address pattern of 10101010101b on flash address inputs A10 to A0; but software would define this as a byte address value of AAAh since the least significant address bit is not used by the flash); which is 101010101010b on system address bus a11 to a0. Because system a11 to a1 is connected to flash A10 to A0 the flash word address of 555h and the system byte address of AAAh provides the same bit pattern on the same address inputs. Because all address values are being documented as system byte addresses, that are more familiar to software writers, the command tables have addresses that are shifted from those shown in prior generation devices.

7.2 Flash Memory Array

The Non-Volatile Flash Memory Array is organized as shown in the following tables. Devices have either all uniform size sectors or four smaller sectors at either the top or bottom of the device.

Table 7.1 System Versus Flash View of Address

System Address Signals	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
System Byte Address Hex	A				A				A			
Binary Pattern	1	0	1	0	1	0	1	0	1	0	1	0
Flash Word Address Hex	5			5				5				
Flash Address Signals	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	

Table 7.2 S29WS512R Sector and Memory Address Map (Bottom Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
32	4	32	0	SA000	000000h–003FFFh	000000h–007FFFh	Sector Starting Address to Sector Ending Address
				SA001	004000h–007FFFh	008000h–00FFFFh	
				SA002	008000h–00BFFFh	010000h–017FFFh	
				SA003	00C000h–00FFFFh	018000h–01FFFFh	
	31	128		SA004	010000h–01FFFFh	020000h–03FFFFh	
				⋮	⋮	⋮	
				SA034	1F0000h–1FFFFFh	3E0000h–3FFFFFh	
32	480	128	1	SA035–SA066	200000h–3FFFFFh	400000h–7FFFFFh	
			2	SA067–SA098	⋮	⋮	
			3	SA097–SA130	⋮	⋮	
			4	SA131–SA162	⋮	⋮	
			5	SA163–SA194	⋮	⋮	
			6	SA195–SA226	⋮	⋮	
			7	SA227–SA258	⋮	⋮	
			8	SA259–SA290	1000000h–11FFFFFh	2000000h–23FFFFFh	
			9	SA291–SA322	⋮	⋮	
			10	SA323–SA354	⋮	⋮	
			11	SA355–SA386	⋮	⋮	
			12	SA387–SA418	⋮	⋮	
			13	SA419–SA450	⋮	⋮	
			14	SA451–SA482	1C00000h–1DFFFFFh	3800000h–3BFFFFFh	
			15	SA483–SA514	1E00000F–1FFFFFFh	3C00000F–3FFFFFFh	

Note

All tables have been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the byte address pattern x000000h–x1FFFFh.

Table 7.3 S29WS512R Sector and Memory Address Map (Top Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
32	480	128	0	SA000–SA031	000000h–1FFFFFh	000000h–3FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA032–SA063	200000h–3FFFFFh	400000h–7FFFFFh	
			2	SA064–SA095	⋮	⋮	
			3	SA096–SA127	⋮	⋮	
			4	SA128–SA159	⋮	⋮	
			5	SA160–SA191	⋮	⋮	
			6	SA192–SA223	⋮	⋮	
			7	SA224–SA255	⋮	⋮	
			8	SA256–SA287	1000000h–11FFFFFh	2000000h–23FFFFFh	
			9	SA288–SA319	⋮	⋮	
			10	SA320–SA351	⋮	⋮	
			11	SA352–SA383	⋮	⋮	
			12	SA384–SA415	⋮	⋮	
			13	SA416–SA447	⋮	⋮	
			14	SA448–SA479	1C00000h–1DFFFFFh	3800000h–3BFFFFFh	
32	31	128	15	SA480	1E00000h–1E0FFFF	3C00000h–3C1FFFF	Sector Starting Address – Sector Ending Address
				⋮	⋮	⋮	
				SA510	1FE0000h–1FEFFFFh	3FC0000h–3FDFFFFh	
	4	32		SA511	1FF0000h–1FF3FFFh	3FE0000h–3FE7FFFh	
				SA512	1FF4000h–1FF7FFFh	3FE8000h–3FEFFFFh	
				SA513	1FF8000h–1FFBFFFh	3FF0000h–3FF7FFFh	
				SA514	1FFC000h–1FFFFFFh	3FF8000h–3FFFFFFh	

Table 7.4 S29WS512R Sector and Memory Address Map (Uniform Sectors)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
32	512	128	0	SA000–SA031	000000h–1FFFFFh	000000h–3FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA032–SA063	200000h–3FFFFFh	400000h–7FFFFFh	
			2	SA064–SA095	⋮	⋮	
			3	SA096–SA127	⋮	⋮	
			4	SA128–SA159	⋮	⋮	
			5	SA160–SA191	⋮	⋮	
			6	SA192–SA223	⋮	⋮	
			7	SA224–SA255	⋮	⋮	
			8	SA256–SA287	1000000h–11FFFFFh	2000000h–23FFFFFh	
			9	SA288–SA319	⋮	⋮	
			10	SA320–SA351	⋮	⋮	
			11	SA352–SA383	⋮	⋮	
			12	SA384–SA415	⋮	⋮	
			13	SA416–SA447	⋮	⋮	
			14	SA448–SA479	1C00000h–1DFFFFFh	3800000h–3BFFFFFh	
			15	SA480–SA511	1E00000h–1FFFFFFh	3C00000h–3FFFFFFh	

Note

All tables have been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the byte address pattern x000000h–x1FFFFh.

Table 7.5 S29WS256R Sector and Memory Address Map (Bottom Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
16	4	32	0	SA000	000000h–003FFFh	000000h–007FFFh	Sector Starting Address – Sector Ending Address
				SA001	004000h–007FFFh	008000h–00FFFFh	
				SA002	008000h–00BFFFh	010000h–017FFFh	
				SA003	00C000h–00FFFFh	018000h–01FFFFh	
	15	128		SA004	010000h–01FFFFh	020000h–03FFFFh	
				⋮	⋮	⋮	
				SA018	0F0000h–0FFFFFh	170000h–1FFFFFh	
16	240	128	1	SA019–SA034	100000h–1FFFFFh	200000h–3FFFFFh	
			2	SA035–SA050	⋮	⋮	
			3	SA051–SA066	⋮	⋮	
			4	SA067–SA082	⋮	⋮	
			5	SA083–SA098	⋮	⋮	
			6	SA099–SA114	⋮	⋮	
			7	SA115–SA130	⋮	⋮	
			8	SA131–SA146	800000h–8FFFFFh	1000000h–11FFFFFh	
			9	SA147–SA162	⋮	⋮	
			10	SA163–SA178	⋮	⋮	
			11	SA179–SA194	⋮	⋮	
			12	SA195–SA210	⋮	⋮	
			13	SA211–SA226	⋮	⋮	
			14	SA227–SA242	E00000h–EFFFFFh	1C00000h–1DFFFFFh	
			15	SA243–SA258	F00000h–FFFFFFh	1E00000h–1FFFFFFh	

Note

All tables have been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the byte address pattern x000000h–x1FFFFh.

Table 7.6 S29WS256R Sector and Memory Address Map (Top Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
16	240	128	0	SA000–SA015	000000h–0FFFFFh	000000h–1FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA016–SA031	100000h–1FFFFFh	200000h–3FFFFFh	
			2	SA032–SA047	⋮	⋮	
			3	SA048–SA063	⋮	⋮	
			4	SA064–SA079	⋮	⋮	
			5	SA080–SA095	⋮	⋮	
			6	SA096–SA111	⋮	⋮	
			7	SA112–SA127	⋮	⋮	
			8	SA128–SA143	800000h–8FFFFFh	1000000h–11FFFFFh	
			9	SA144–SA159	⋮	⋮	
			10	SA160–SA175	⋮	⋮	
			11	SA176–SA191	⋮	⋮	
			12	SA192–SA207	⋮	⋮	
			13	SA208–SA223	⋮	⋮	
			14	SA224–SA239	E00000h–EFFFFFh	1C00000h–1DFFFFFh	
	15	128	15	SA240	F00000h–F0FFFFh	1E00000h–1E1FFFFh	
				⋮	⋮	⋮	
				SA254	FE0000h–FEFFFFh	1FC0000h–1FDFFFFh	
	4	32		SA255	FF0000h–FF3FFFh	1FE0000h–1FE7FFFh	
				SA256	FF4000h–FF7FFFh	1FE8000h–1FEFFFFh	
				SA257	FF8000h–FFBFFFh	1FF0000h–1FF7FFFh	
				SA258	FFC000h–FFFFFFh	1FF8000h–1FFFFFFh	

Note

All tables have been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the byte address pattern x000000h–x1FFFFh.

Table 7.7 S29WS256R Sector and Memory Address Map (Uniform Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
16	256	128	0	SA000–SA015	000000h–0FFFFFh	000000h–1FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA016–SA031	100000h–1FFFFFh	200000h–3FFFFFh	
			2	SA032–SA047	⋮	⋮	
			3	SA048–SA063	⋮	⋮	
			4	SA064–SA079	⋮	⋮	
			5	SA080–SA095	⋮	⋮	
			6	SA096–SA111	⋮	⋮	
			7	SA112–SA127	⋮	⋮	
			8	SA128–SA143	800000h–8FFFFFh	1000000h–11FFFFFh	
			9	SA144–SA159	⋮	⋮	
			10	SA160–SA175	⋮	⋮	
			11	SA176–SA191	⋮	⋮	
			12	SA192–SA207	⋮	⋮	
			13	SA208–SA223	⋮	⋮	
			14	SA224–SA239	E00000h–EFFFFFh	1C00000h–1DFFFFFh	
			15	SA240–SA255	F00000h–FFFFFFh	1E00000h–1FFFFFFh	

Table 7.8 S29WS128R Sector and Memory Address Map (Bottom Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
8	4	32	0	SA000	000000h–003FFFh	000000h–007FFFh	Sector Starting Address – Sector Ending Address
				SA001	004000h–007FFFh	008000h–00FFFFh	
				SA002	008000h–00BFFFh	010000h–017FFFh	
				SA003	00C000h–00FFFFh	018000h–01FFFFh	
	7	128		SA004	010000h–01FFFFh	020000h–03FFFFh	
				⋮	⋮	⋮	
				SA010	070000h–07FFFFh	0E0000h–FFFFFh	
8	120	128	1	SA011–SA018	080000h–0FFFFFh	100000h–1FFFFFh	
			2	SA019–SA026	⋮	⋮	
			3	SA027–SA034	⋮	⋮	
			4	SA035–SA042	⋮	⋮	
			5	SA043–SA050	⋮	⋮	
			6	SA051–SA058	⋮	⋮	
			7	SA059–SA066	⋮	⋮	
			8	SA067–SA074	400000h–47FFFFh	800000h–8FFFFFh	
			9	SA075–SA082	⋮	⋮	
			10	SA083–SA090	⋮	⋮	
			11	SA091–SA098	⋮	⋮	
			12	SA099–SA106	⋮	⋮	
			13	SA107–SA114	⋮	⋮	
			14	SA115–SA122	700000h–77FFFFh	E00000h–EFFFFFh	
			15	SA123–SA130	780000h–7FFFFFh	F00000h–FFFFFFh	

Note

All tables have been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the byte address pattern x000000h–x1FFFFh.

Table 7.9 S29WS128R Sector and Memory Address Map (Top Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
8	120	128	0	SA000–SA007	000000h–07FFFFh	000000h–0FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA008–SA015	080000h–0FFFFFh	100000h–1FFFFFh	
			2	SA016–SA023	⋮	⋮	
			3	SA024–SA031	⋮	⋮	
			4	SA032–SA039	⋮	⋮	
			5	SA040–SA047	⋮	⋮	
			6	SA048–SA055	⋮	⋮	
			7	SA056–SA063	⋮	⋮	
			8	SA064–SA071	400000h–47FFFFh	800000h–8FFFFFh	
			9	SA072–SA079	⋮	⋮	
			10	SA080–SA087	⋮	⋮	
			11	SA088–SA095	⋮	⋮	
			12	SA096–SA103	⋮	⋮	
			13	SA104–SA111	⋮	⋮	
			14	SA112–SA119	700000h–77FFFFh	E00000h–EFFFFFh	
	7	128	15	SA120	780000h–78FFFFh	F00000h–F1FFFFh	
				⋮	⋮	⋮	
				SA126	7E0000h–7EFFFFh	FC0000h–FDFFFFh	
	4	32		SA127	7F0000h–7F3FFFh	FE0000h–FE7FFFh	
				SA128	7F4000h–7F7FFFh	FE8000h–FEFFFFh	
				SA129	7F8000h–7FBFFFh	FF0000h–FF7FFFh	
				SA130	7FC000h–7FFFFFh	FF8000h–FFFFFFh	

Note

All tables have been condensed to show sector-related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA008–SA009) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 KB sectors have the byte address pattern x000000h–x1FFFFh.

Table 7.10 S29WS128R Sector and Memory Address Map (Uniform Boot)

Bank Size (Mbit)	Sector Count	Sector Size (Kbyte)	Bank	Sector Range	Address Range (word)	Address Range (byte)	Notes
16	128	128	0	SA000–SA007	000000h–07FFFFh	000000h–0FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA008–SA015	080000h–0FFFFFh	100000h–1FFFFFh	
			2	SA016–SA023	⋮	⋮	
			3	SA024–SA031	⋮	⋮	
			4	SA032–SA039	⋮	⋮	
			5	SA040–SA047	⋮	⋮	
			6	SA048–SA055	⋮	⋮	
			7	SA056–SA063	⋮	⋮	
			8	SA064–SA071	400000h–47FFFFh	800000h–8FFFFFh	
			9	SA072–SA079	⋮	⋮	
			10	SA080–SA087	⋮	⋮	
			11	SA088–SA095	⋮	⋮	
			12	SA096–SA103	⋮	⋮	
			13	SA104–SA111	⋮	⋮	
			14	SA112–SA119	700000h–77FFFFh	E00000h–EFFFFFh	
			15	SA120–SA127	780000h–7FFFFFh	F00000h–FFFFFFh	

7.3 Device ID and CFI (ID-CFI)

There are two traditional methods for systems to identify the type of flash memory installed in the system. One has been traditionally been called Autoselect and is now referred to as Device Identification (ID). A command is used to enable an address space overlay where up to 16 word locations can be read to get JEDEC manufacturer identification (ID), device ID, and some configuration and protection status information from the flash memory. The system can use the manufacturer and device IDs to select the appropriate driver software to use with the flash device. The other method is called Common Flash Interface (CFI). It also uses a command to enable an address space overlay where an extendable table of standard information about how the flash memory is organized and behaves can be read. With this method the driver software does not have to be written with the specifics of each possible memory device in mind. Instead the driver software is written in a more general way to handle many different devices but adjusts the driver behavior based on the information in the CFI table stored in the flash memory. Traditionally these two address spaces have used separate commands and were separate overlays. However, the mapping of these two address spaces are now non-overlapping and so can be combined in to a single address space and appear together in a single overlay. Either of the traditional commands used to access enter the Autoselect (ID) or CFI overlay will cause the now combined ID-CFI address map to appear.

A write at any sector address, in bank zero, having the least significant byte address value of AAh, with xx98h or xx90h data, switches the addressed sector to an overlay of the ID-CFI address map. These are called ID-CFI Enter commands and are only valid when written to the specified bank when it is in read mode. The ID-CFI address map appears within, and replaces flash array data of, the selected sector address range. The ID-CFI enter commands use the same address and data values used on previous generation memories to access the JEDEC Manufacturer ID (Autoselect) and Common Flash Interface (CFI) information, respectively. While the ID-CFI address space is overlaid, any write with xxF0h data to the device will exit the overlay and return the selected sector to showing flash memory array data. Thus, the ID-CFI address space and commands are backward compatible with standard memory discovery algorithms.

Within the ID-CFI address map there are two subsections:

Table 7.11 ID-CFI Address Map Overview

Byte Address	Description	Size Allocated (Bytes)	Read/Write
(SA) + 00000h to 0001Fh	JEDEC ID (traditional Autoselect values)	32	Read Only
(SA) + 00020h to CEh	CFI data structure	174	Read Only

For the complete address map in *Device ID and Common Flash Memory Interface Address Map* on page 65.

7.3.1 JEDEC Device ID

The Joint Electron Device Engineering Council (JEDEC) standard JEP106x defines a method for reading the manufacturer ID and device ID of a compliant memory. This information is primarily intended for programming equipment to automatically match a device with the corresponding programming algorithm.

The JEDEC ID information is structured to work with any memory data bus width (e.g. x8, x16, x32). The Query addressing is always relative to the device word (largest supported) with data always presented on the lowest order byte (D7 - D0 outputs). Because the data bus is word wide each code byte is located in the lower half of each word location and the high order byte is always zero.

7.3.2 Common Flash Memory Interface

The Common Flash Interface (CFI) specification defines a standardized data structure that may be read from a flash memory device, which allows vendor-specified software algorithms to be used for entire families of devices. The data structure contains information for system configuration such as various electrical and timing parameters, and special functions supported by the device. Software support can then be device-independent, JEDEC ID-independent, and forward-and-backward-compatible for the specified flash device families.

The system can read CFI information at the addresses within the selected sector as shown in Section 12.2, *Device ID and Common Flash Memory Interface Address Map* on page 65.

Like the JEDEC Device ID information, CFI information is structured to work with any memory data bus width (e.g. x8, x16, x32). The Query addressing is always relative to the device word (largest supported) with data always presented on the lowest order byte (D7 - D0 outputs). Because the data bus is word wide each code byte is located in the lower half of each word location and the high order byte is always zero.

For further information, please refer to the Spansion CFI Version 1.4 (or later) Specification and the Spansion CFI Publication 100 (see also JEDEC publications JEP137-A and JESD68.01). Please contact JEDEC (http://www.jedec.org) for their standards.

7.3.3 Secure Silicon Region

The Secure Silicon Region (SSR) provides an extra flash memory area that can be programmed once and permanently protected from further changes. The SSR is 512 bytes in length. It consists of 256 bytes for factory data and 256 bytes for customer-secured data.

The SSR is overlaid in the sector address specified by the SSR enter command.

Table 7.12 Secure Silicon Region

Byte Address Range	Secure Silicon Region	Size
(SA) + 0000h to 00FFh	Factory	256 Bytes
(SA) + 0100h to 01FFh	Customer	256 Bytes

7.3.4 Configuration Register

The Configuration Register Enter command is only valid when written to a bank that is in Read mode. The configuration register mode address map appears within, and replaces flash array data of, the selected sector address range. The meaning of the configuration register bits is defined in *Configuration Register* on page 29. In configuration register mode a write of 00F0h to any address will return the sector to Standard Read mode.

8. Device Operations

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

The address space of the Flash Memory Array is divided into banks. There are three operation modes for each bank:

- Read Mode
- Embedded Algorithm (EA) Mode
- Address Space Overlay (ASO) Mode

Each bank of the device can be in any operation mode but, only one bank can be in EA or ASO mode at any one time.

In Read Mode a Flash Memory Array bank may be read by simply selecting the memory, supplying the address, and taking read data when it is ready. This is done by asynchronous or burst accesses from the host system bus. The CU puts all banks in Read mode during Power-on, a Hardware Reset, after a Command Reset, or after a bank is returned to Read mode from EA mode.

During a burst read access valid read data is indicated by the RDY signal being High. When RDY is Low burst read data is not valid and wait states must be added. The use of the RDY signal to indicate when valid data is transferred on the system data bus is called handshaking or flow control.

EA and ASO modes are initiated by writing specific address and data patterns into command registers (see Table 12.1 on page 63). The command registers do not occupy any memory locations; they are loaded by write bus cycles with the address and data information needed to execute a command. The contents of the registers serve as input to the Control Unit (CU) and the CU dictates 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 all banks to Read mode.

The flash memory array data in a bank that is in EA mode, is stable but undefined, and effectively unavailable for read access from the host system. While in EA mode the bank is used by the CU in the execution of commands. Typical command operations are programming or erasing of data in the flash array. All other banks are available for read access while the one bank is in EA mode. This ability to read from one bank while another bank is used in the execution of a command is called Simultaneous Read and Write (SRW) and allows for continued operation of the system via the reading of data or code from other banks while one bank is programming or erasing data as a relatively long time frame background task. Only a status register read command can be used in a bank in EA mode to retrieve the EA status.

While any one of the overlay address spaces are overlaid in a bank (entered) that bank is in ASO mode and no other bank may be in EA or ASO mode. All EA activity must be completed or suspended before entering any ASO mode. A command for entering an EA or ASO mode while another bank is in EA or ASO mode will be ignored.

While an ASO mode is active (entered) in a bank, a read for flash array data to any other bank is allowed. ASO mode selects a specific sector for the overlaid address space. Other sectors in the ASO bank still provide flash array data and may be read during ASO mode.

While SSR Lock, SSR, or Configuration Register is overlaid only the SSR Lock, SSR, or Configuration Register respectively may be programmed in the overlaid sector. While any of these ASO areas are being programmed the ASO bank switches to EA mode. The ID/CFI and factory portion of the SSR ASO is not customer programmable. An attempt to program in these areas will fail.

8.1 Device Bus Operations

The Device Bus Operations table describes the required state of each control pin for any particular bus operation. The Control Unit (CU) is set to the idle state for reading array data upon device power-up, or after a hardware reset. This ensures that no spurious alteration of the memory content occurs during the power transition.

Table 8.1 Device Bus Operations

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

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

8.2 Asynchronous Read

To read data from the memory array, the system must first assert a valid address while driving AVD# and CE# to V_{IL}. WE# must remain at V_{IH}. CLK may toggle or remain at V_{IL} or V_{IH}. The rising edge of AVD# will latch the address. The data appears on DQ15–DQ0 when CE# is Low, OE# is Low, AVD# is High, and the asynchronous access times are satisfied.

In order to use Asynchronous Read Mode the configuration register bit 15 must be set to 1.

Address access time (t_{ACC}) is equal to the delay from stable addresses to valid output data. The chip enable access time (t_{CE}) is the delay from stable CE# to valid data at the outputs. See 11.9.2, *AC Characteristics–Asynchronous Read* on page 53.

8.3 Page Mode Read

The device is capable of fast page mode read. This mode provides random read access speed for locations within a page. Address bits Amax–A3 select a 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.

The random or initial page access is t_{ACC} or t_{CE} and subsequent page read accesses (as long as the locations specified falls within that page) is equivalent to t_{PACC}. When CE# is deasserted (=V_{IH}), the re-assertion of CE# for subsequent access has access time of t_{ACC} or t_{CE}. See Figure 11.13 on page 55.

Table 8.2 Word Select

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

8.4 Synchronous (Burst) Read Mode and Configuration Register

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

In order to use Synchronous (Burst) Read Mode the configuration register bit 15 must be set to 0.

Prior to entering burst mode, the system should determine how many wait states are needed for the initial word of each burst access (see table below), what mode of burst operation is desired, how the RDY signal transitions with valid data, and output drive strength. The system would then write the configuration register command sequence. See *Configuration Register* on page 29 for further details.

When the appropriate number of initial Wait States have occurred, data is output after the **rising edge** of the CLK. Subsequent words are output t_{BACC} after the rising edge of each successive clock cycle, which automatically increments the internal address counter. RDY indicates the initial latency and any subsequent wait states. The device has 3 burst options: Continuous Burst, 16-Word Burst with Wrap Around, and 8-Word Burst with Wrap Around. If the device is operated in Continuous Burst mode or 16-word Burst mode,

The device has 3 burst options: Continuous Burst, 16-Word Burst with Wrap Around, and 8-Word Burst with Wrap Around. If the device is operated in Continuous Burst mode or 16-word Burst mode, 0 to 7 wait states may be inserted at 8 word boundary crossings. Wait states are inserted between the last word below the boundary and the first word above the boundary. The number of wait states inserted are based on the initial address and the initial number of wait states. See the Address Latency tables for the number of wait states inserted.

The device also has a fixed internal address boundary that occurs every 128 words (256 Bytes). When a 128 word boundary is crossed, 0 to 2 additional wait states are inserted. The 128-word boundary can only be crossed when the device is operated in continuous burst mode. See Table 8.1 on page 24 for the number of wait states inserted at the 128-word boundary.

The following table shows the number of initial wait states needed at different burst frequencies.

Table 8.3 Initial Wait States vs. Frequency

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 ≤ 79 MHz	7
79 MHz < Frequency ≤ 95 MHz	8
95 MHz < Frequency ≤ 104 MHz	9
104 MHz < Frequency ≤ 120 MHz	10

The following tables show the address related latency (note that ws = wait state).

Table 8.4 Address Latency for 8 to13 Wait States

Word	Initial Wait	Wait States Inserted at 8 Word (16 byte) Boundaries After Initial Wait States								
0	8 to 13 wait states	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	1 ws	D8
2		D2	D3	D4	D5	D6	D7	2 ws		D8
3		D3	D4	D5	D6	D7	3 ws			D8
4		D4	D5	D6	D7	4 ws				D8
5		D5	D6	D7	5 ws					D8
6		D6	D7	6 ws						D8
7		D7	7 ws							D8

Table 8.5 Address Latency for 7 Wait States

Word	Initial Wait	Wait States Inserted at 8 Word (16 byte) Boundaries After Initial Wait States								
0	7 wait states	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	0 ws	D8
2		D2	D3	D4	D5	D6	D7	1 ws		D8
3		D3	D4	D5	D6	D7	2 ws			D8
4		D4	D5	D6	D7	3 ws				D8
5		D5	D6	D7	4 ws					D8
6		D6	D7	5 ws						D8
7		D7	6 ws							D8

Table 8.6 Address Latency for 6 Wait States

Word	Initial Wait	Wait States Inserted at 8 Word (16 byte) Boundaries After Initial Wait States								
0	6 wait states	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	0 ws	D8
2		D2	D3	D4	D5	D6	D7	0 ws		D8
3		D3	D4	D5	D6	D7	1 ws			D8
4		D4	D5	D6	D7	2 ws				D8
5		D5	D6	D7	3 ws					D8
6		D6	D7	4 ws						D8
7		D7	5 ws							D8

Table 8.7 Address Latency for 5 Wait States

Word	Initial Wait	Wait States Inserted at 8 Word (16 byte) Boundaries After Initial Wait States								
0	5 wait states	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	0 ws	D8
2		D2	D3	D4	D5	D6	D7	0 ws		D8
3		D3	D4	D5	D6	D7	0 ws			D8
4		D4	D5	D6	D7	1 ws				D8
5		D5	D6	D7	2 ws					D8
6		D6	D7	3 ws						D8
7		D7	4 ws							D8

Table 8.8 Address Latency for 4 Wait States

Word	Initial Wait	Wait States Inserted at 8 Word (16 byte) Boundaries After Initial Wait States								
0	4 wait states	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	0 ws	D8
2		D2	D3	D4	D5	D6	D7	0 ws		D8
3		D3	D4	D5	D6	D7	0 ws			D8
4		D4	D5	D6	D7	0 ws				D8
5		D5	D6	D7	1 ws					D8
6		D6	D7	2 ws						D8
7		D7	3 ws							D8

Table 8.9 Address Latency for 3 Wait States

Word	Initial Wait	Wait States Inserted at 8 Word (16 byte) Boundaries After Initial Wait States								
0	3 wait states	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	0 ws	D8
2		D2	D3	D4	D5	D6	D7	0 ws		D8
3		D3	D4	D5	D6	D7	0 ws			D8
4		D4	D5	D6	D7	0 ws				D8
5		D5	D6	D7	0 ws					D8
6		D6	D7	1 ws						D8
7		D7	2 ws							D8

Table 8.10 256 Byte Boundary Crossing Latency - Additional Wait States

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

8.4.1 Continuous Burst

The device continues to output sequential burst data from the memory array, wrapping around to address 0000000h after it reaches the highest addressable memory location, until the system drives CE# high, RESET# low, or AVD# low in conjunction with a new address. See Table 8.1, *Device Bus Operations* on page 24.

If the host system crosses a bank boundary while reading in burst mode, and the subsequent bank is not programming or erasing, an address boundary crossing latency is required. If the host system crosses the bank boundary while the subsequent bank is programming or erasing, continuous burst halts (RDY will be disabled and data will continue to be driven).

8.4.2 8-, 16-Word Linear Burst with Wrap Around

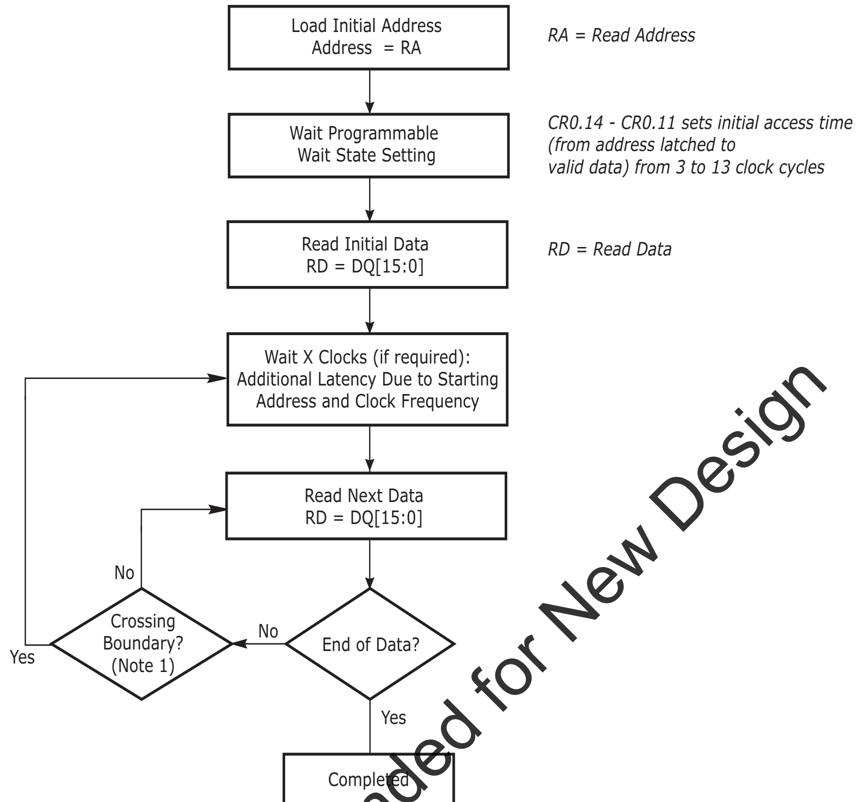
Table 8.11 Burst Address Groups

Mode	Group Size	Group Byte Address Ranges
8-word	16 bytes	0-Fh, 10-1Fh, 20-2Fh,...
16-word	32 bytes	0-1Fh, 20-3Fh, 30-4Fh,...

The remaining two modes are fixed length linear burst with wrap around, in which a fixed number of words are read from consecutive addresses. In each of these modes, the burst addresses read are determined by the group within which the starting address falls. The groups are sized according to the number of words read in a single burst sequence for a given mode (see Table 8.11).

As an example: if the starting address in the 8-word mode is system byte address 3Ch, the address range to be read would be byte address 30-3Fh, and the burst sequence would be 3C-3E-30-32-34-36-38-3Ah. The burst sequence begins with the starting address written to the device, wraps back to the first address in the selected group, and outputs a maximum of 8 words. No additional wait states will be required within the 8-word burst. The 8th word will continue to be driven until the burst operation is aborted (CE# goes to V_{IH}, a new address is latched in for a new burst operation, or a hardware reset). In a similar fashion, the 16-word Linear Wrap modes begin their burst sequence on the starting address written to the device, and then wrap back to the first address in the selected address group. Additional wait states could be added the first time the device crosses from one to the other group of 8 words in a 16-word burst. The number will depend on the starting address and the wait state set within the configuration register. See Table 8.4 on page 26 to Table 8.9 on page 27. **Note that in these two burst read modes the address pointer does not cross the boundary that occurs every 128 words; thus, no address boundary crossing wait states are inserted for linear burst with wrap.**

Figure 8.1 Synchronous Read



Note

1. Required only if device is performing a Continuous Burst operation.

8.4.3 Configuration Register

Configuration register (CR) sets various operational parameters associated with burst mode. Upon power-up or hardware reset, the device defaults to the idle state, and the configuration register settings are in their default state. The host system should determine the proper settings for the 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 12.1 on page 63). The table below describes the register settings and indicates the default state of each bit after power-on or a hardware reset. The configuration register bits are not affected by a command reset.

Table 8.12 Configuration Register

CR Bit	Function	Settings (Binary)
CR.15	Device Read Mode	0 = Synchronous Read Mode 1 = Asynchronous Read Mode (Default)
CR.14 CR.13 CR.12 CR.11	Programmable Read Wait States	0000 = Reserved 0001 = Initial data is valid on the 3rd rising CLK edge after addresses are latched 0010 = 4th 0011 = 5th ⋮ 1011 = 13th (Default) 1100 = Reserved 1101 = Reserved 1110 = Reserved 1111 = Reserved
CR.10	RDY Polarity	0 = RDY signal is active low 1 = RDY signal is active high (Default)
CR.9	Reserved	0 = Reserved 1 = Reserved (Default)
CR.8	RDY Timing	0 = RDY active one clock cycle before data 1 = RDY active with data (Default)
CR.7	Output Drive Strength	0 = Full Drive= Current Driver Strength (Default) 1 = Half Drive
CR.6	Reserved	0 = Reserved 1 = Reserved (Default)
CR.5	Reserved	0 = Reserved (Default) 1 = Reserved
CR.4	Reserved	0 = Reserved (Default) 1 = Reserved
CR.3	Reserved	0 = Reserved 1 = Reserved (Default)
CR.2 CR.1 CR.0	Burst Length	000 = Continuous (Default) 010 = 8-Word (16-Byte) Linear Burst with wrap around 011 = 16-Word (32-Byte) Linear Burst with wrap around (All other bit settings are reserved)

8.4.3.1 Device Read Mode

Configuration Register bit 15 (CR.15) controls whether read accesses via the bus interface are in asynchronous or burst mode. Asynchronous mode is the default after power-on or hardware reset. Write accesses are always conducted with asynchronous mode timing, independent of the read mode.

8.4.3.2 Wait States

Configuration Register bits 14 to 11 (CR.[14..11]) define the number of CLK cycles after the AVD# Low cycle that captures the initial address until the device presents valid data on the data bus. The bits from 14 to 11 are in most to least significant order.

When the corresponding number of Wait States have occurred, data is presented on the data bus after the rising edge of the CLK. Subsequent words are output t_{BACC} after the rising edge of each successive clock cycle, which automatically increments the internal address counter. Prior to setting the device into synchronous read mode, the system should set CR[14..11] according to the CLK frequency. Appropriate settings are indicated in Table 8.3 on page 26

8.4.3.3 RDY Polarity

Configuration Register bit 10 (CR.10) controls whether the RDY signal indicates valid data when High or when Low. When this bit is zero the RDY signal indicates data is valid when the signal is Low. When this bit is one the RDY signal indicates data is valid when the signal is High. The default for this bit is set to one after power-on or a hardware reset.

8.4.3.4 RDY Timing

Configuration Register bit 8 (CR.8) controls whether the RDY signal indicates valid data on the same cycle that data is valid or one cycle before data is valid. When this bit is zero, the RDY signal indicates data is valid in the same cycle the data is valid. When this bit is one the RDY signal indicates data is valid one cycle before data is valid. The default for this bit is set to one after power-on or a hardware reset.

8.4.3.5 Output Drive Strength

Configuration Register bit 7 (CR.7) controls whether the data outputs drive with full or half strength. When this bit is zero the data outputs drive with full strength. When this bit is one the data outputs drive with half strength. The default for this bit is cleared to zero after power-on or a hardware reset.

8.4.3.6 Burst Length

Configuration Register bits 2 to 0 (CR.[2..0]) define the length of burst read and write accesses. The bits from 2 to 0 are in most to least significant order. See Table 8.13 for code meaning & default value.

8.5 Status Register

The status of program and erase operations is provided by a single 16 bit status register. The status register read command is written followed by a read of the status register for each access of the status register information. The status register can be read in synchronous or asynchronous bus access mode. If read in synchronous (burst) mode for more than access cycle the same status results will appear in each of the read cycles of the burst access until the burst is terminated.

Table 8.13 Status Register Reset State

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB 1 at Reset	ESSB 0 at Reset	ESB 0 at Reset	PSB 0 at Reset	RFU 0 at Reset	PSSB 0 at Reset	SLSB 0 at Reset	BSB 0 at Reset

Notes
1. *Status bits 15 to 8 always return the current progress code.*
2. *Bit 7 reflects the device status.*
3. *If the device is busy, Bit 0 is used to check whether the addressed bank is busy or some other bank is busy.*
4. *All the other bits reflect the status of the device.*

Table 8.14 Status Register - Bit 7

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
0 Device busy programming or erasing	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	VALID
1 Device ready	VALID	VALID	VALID	VALID	VALID	VALID	VALID

Notes

1. *Bit 7 is set when there is no erase or program operation in progress in the device.*
2. *Bits 1 through 6 are valid if and only if Bit 7 is set.*

Table 8.15 Status Register - Bit 6

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
1 Bits 6:1 only valid when Bit 7 = 1	0 No Erase in Suspension	X	X	X	X	X	X
1 Bit 6:1 only valid when Bit 7 = 1	1 Erase in Suspension	X	X	X	X	X	X

Notes

1. *Upon issuing the "Erase Suspend" Command, the user must continue to read status until DRB becomes 1 before accessing another sector within the same bank.*
2. *Cleared by "Erase Resume" Command.*

Table 8.16 Status Register - Bit 5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
1 Bits 6:1 only valid when Bit 7 = 1	X	0 Erase successful	X	X	X	X	X
1 Bit 6:1 only valid when Bit 7 = 1	X	1 Erase error	X	X	X	X	X

Notes

1. *ESB bit reflects "success" or "failure" of the most recent erase operation.*
2. *Cleared by "Clear Status Register" Command as well as by hardware reset.*

Table 8.17 Status Register - Bit 4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
1 Bits 6:1 only valid when Bit 7 = 1	X	X	0 Program successful	X	X	X	X
1 Bit 6:1 only valid when Bit 7 = 1	X	X	1 Program fail	X	X	X	X

Notes
1. *PSB bit reflects "success" or "failure" of the most recent program operation.*
2. *Cleared by "Clear Status Register" Command as well as by hardware reset.*

Table 8.18 Status Register - Bit 3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
1 Bits 6:1 only valid when Bit 7 = 1	X	X	X	X	X	X	X

Notes
1. *This Register is reserved for future use.*
2. *Cleared by "Clear Status Register" Command as well as by hardware reset.*

Table 8.19 Status Register - Bit 2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
1 Bits 6:1 only valid when Bit 7 = 1	X	X	X	X	0 No Program in suspension	X	X
1 Bit 6:1 only valid when Bit 7 = 1	X	X	X	X	1 Program in suspension	X	X

Notes
1. *Upon issuing the "Program Suspend" Command, the user must continue to read status until DRB becomes 1 before accessing another sector within the same bank.*
2. *Cleared by "Program Resume" Command.*

Table 8.20 Status Register - Bit 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
1 Bits 6:1 only valid when Bit 7 = 1	X	X	X	X	X	0 Sector not locked during operation	X
1 Bit 6:1 only valid when Bit 7 = 1	X	X	X	X	X	1 Sector locked error	X

Notes
1. *SLSB indicates that a program or erase operation failed to program or erase because the sector was locked or the operation was attempted on the protected Secure Silicon Region.*
2. *SLSB reflects the status of the most recent program or erase operation.*
3. *SLSB is cleared by "Clear Status Register" or by hardware reset.*

Table 8.21 Status Register - Bit 0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Ready Bit. Overall status	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	RFU	Program Suspend Status Bit	Sector Lock Status Bit	Bank Status Bit
DRB	ESSB	ESB	PSB	RFU	PSSB	SLSB	BSB
0 Bits 6:1 only valid when Bit 7 = 1	X	X	X	X	X	X	0 Program or Erase op. in addressed Bank
0 Bits 6:1 only valid when Bit 7 = 1	X	X	X	X	X	X	1 Program or Erase op. in other Bank
1 Bit 6:1 only valid when Bit 7 = 1	X	X	X	X	X	X	0 No active Program or Erase op.
1 Bit 6:1 only valid when Bit 7 = 1	X	X	X	X	X	X	1 invalid

Notes
BSB is used to check if a program or erase operation in progress in the current bank.

8.6 Blank Check

The Blank Check command will confirm if the selected sector is erased.

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

- Blank Check is only functional in Asynchronous Read mode (Configuration Register - CR[15] = 1).
- To initiate a Blank Check on Sector X, write 33h to address AAAh in Sector X. while the device is in the Idle state (not during program suspend, not during erase suspend,...).

- The Blank Check command may not be written while the device is actively programming or erasing. Blank Check does not support simultaneous operations.
- Use the Status Register read to confirm if the device is still busy and when compete if the sector is blank or not.
- Bit 5 of the Status Register will be cleared to zero if the sector is erased and set to one if not erased.
- Bit 7 & Bit 0 of the Status Register will show if the device is performing a Blank Check (similar to an erase operation).
- As soon as any bit is found to not be erased, the device will halt the operation and report the results.
- Once the Blank Check is completed, the device will to return to the Idle State.

8.7 Simultaneous Read/Write

The simultaneous read/write feature allows the host system to read data from one bank of memory while programming or erasing another bank of memory. An erase operation may also be suspended to read from or program another location within the same bank (note: programming to the sector being erased is not allowed). Figure 11.19, *Back-to-Back Read/Write Cycle Timings* on page 61 shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to the *DC Characteristics* on page 48 table for read-while-program and read-while-erase current specification.

8.8 Writing Commands/Command Sequences

The device accepts Asynchronous write bus operations. During an asynchronous write bus operation, the system must drive CE# and WE# to V_{IL} and OE# to V_{IH} when providing an address and data. When latching an address, AVD# must be driven to V_{IL}. Addresses are latched on the rising edge of AVD#, while data is latched on the rising edge of WE#. See the Table 8.1, *Device Bus Operations* on page 24 for the signal combinations that define each phase of a write bus operation to the device. Each write is a command or part of a command sequence to the device. The address provided in each write operation may be a bit pattern used to help identify the write as a command to the device. The upper portion of the address may also select the bank or sector the command operation is to be performed. A *Bank Address* (BA) is the set of address bits required to uniquely select a bank. Similarly, a *Sector Address* (SA) is the address bits required to uniquely select a sector. The data in each write identifies the command operation to be performed or supplies information needed to perform the operation. See Table 12.1, *Command Definitions* on page 63 for a listing of the commands accepted by the device. I_{CC2} in *DC Characteristics* on page 48 represents the active current specification for a write (Embedded Algorithm) operation.

8.9 Program/Erase Operations

For all program and/or erase operations, including writing command sequences, the system must drive AVD# and CE# to V_{IL}, and OE# to V_{IH} when providing an address to the device, and drive WE# and CE# to V_{IL}, and OE# to V_{IH} when writing commands or programming data.

Addresses are latched on the rising edge of AVD# during asynchronous writes. Data is latched on the rising edge of WE# during asynchronous writes.

Note the following:

- When the Embedded Program algorithm is complete, the device returns to the calling routing (Erase Suspend, SSR Lock, Secure Silicon Region, or Idle State).
- The system can determine the status of the program operation by reading the Status Register. Refer to *Status Register* on page 31 for information on these status bits.
- A *0* cannot be programmed back to a *1*. A succeeding read shows that the data is still *0*. Only erase operations can convert a *0* to a *1*.

old data	0011
new data	0101
results	0001

- Any commands written to the device during the Embedded Program Algorithm are ignored except the Reads from the non-Programming Bank, Program Suspend, and Status Read command. Any commands written to the device during the Embedded Erase Algorithm are ignored except Reads from the non-Erasing Bank, Erase Suspend and Status Read command.

- A hardware reset immediately terminates the program/erase operation and the program command sequence should be reinitiated once the device has returned to the idle state, to ensure data integrity.

8.9.1 Write Buffer Programming

Write Buffer Programming allows the system to write 1 to 32 words in one programming operation. The Write Buffer Programming command sequence is initiated by first writing the Write Buffer Load command written at the Sector Address + AAAh in which programming occurs. Next, the system writes the number of *word locations minus 1* at the Sector Address + 555h. This tells the device how many write buffer addresses are loaded with data and therefore when to expect the *Program Buffer to Flash* command. The Sector Address must match during the Write Buffer Load command and during the Write Word Count command and the Sector must be unlocked or the operation will abort and return to the initiating state.

The write buffer is used to program data within a 64 byte page aligned on a 64 byte boundary.Thus, a full page Write Buffer programming operation must be aligned on a page boundary. Programming operations of less than a full page may start on any word boundary but may not cross a page boundary.

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. The Sector address must match the Write Buffer Load Sector Address or the operation will abort and return to the initiating state. All subsequent address/data pairs must be in sequential order. All write buffer addresses must be within the same page. If the system attempts to load data outside this range, the operation aborts after the Write to Buffer command is executed and the device will indicate a Program Fail in the Status Register at bit location 4 (PSB). A “Clear Status Register” must be issued to clear the PSB status bit.

The counter decrements for each data load operation.

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 + AAAh. The device then *goes busy*. The Embedded Program algorithm automatically programs and verifies the data for the correct data pattern. The system is not required to provide any controls or timings during these operations. If the incorrect number of write buffer locations have been loaded and the *Program Buffer to Flash* command is issued, the Status Register will indicate a program fail at bit location 4 (PSB). A “Clear Status Register” must be issued to clear the PSB status bit.

The write-buffer *embedded* programming operation can be suspended using the Program Suspend command. When the Embedded Program algorithm is complete, the device then returns to Erase Suspend, SSR Lock, Secure Silicon Region, or Idle state. The system can determine the status of the program operation by reading the Status Register. Refer to *Status Register* on page 31 for information on these status bits.

The Write Buffer Programming Sequence will be Aborted under the following conditions:

- Load a value greater than the buffer size during the Number of Locations step.
- Write an address that is outside the Page of the Starting Address during the write buffer data loading stage of the operation. Or, have a sector address not matching the one used in the initial command cycle of the write buffer operation.
- The sector to be programmed is locked.
- The Program Buffer to flash command is not issued after the Write Word Count number of data words is loaded.

When any of the conditions that cause an abort of write buffer command occur the abort is immediate and will indicate a Program Fail in the Status Register at bit location 4 (PSB). The next successful program operation will clear the failure status or a “Clear Status Register” may be issued to clear the PSB status bit.

The Write Buffer Programming Sequence can be stopped and reset by the following: Hardware Reset or Power cycle.

Software Functions and Sample Code

Table 8.22 Write Buffer Program

(LLD Functions Used = lld_WriteToBufferCmd, lld_ProgramBufferToFlashCmd)

Cycle	Description	Operation	Word Address	Data
1	Write Buffer Load Command	Write	Program Address + 555h	0025h
2	Write Word Count	Write	Program Address + AAAh	Word Count (N–1)h
Number of words (N) loaded into the write buffer can be from 1 to 32 words.				
3 to 34	Load Buffer Word N	Write	Program Address, Word N	Word N
Last	Write Buffer to Flash	Write	Sector Address + 555h	0029h

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

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

```
/* Example: Write Buffer Programming Command           */
/* NOTES: Write buffer programming limited to 16 words. */
/*        All addresses to be written to the flash in   */
/*        one operation must be within the same flash   */
/*        page. A flash page begins at addresses        */
/*        evenly divisible by 0x20.                     */ /*Addresses are given as word addresses.*/
  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 *)sector_address + 0x555 )     = 0x0025;      /* write write buffer load command */
  *( (UINT16 *)sector_address + 0xAAA)     = wc;         /* write word count (minus 1)      */
for(x = wc + 1; x>0; x--)           //  Loop until wc+1 address/data pairs have been entered.
{
  *dst = *src; /* ALL dst MUST BE SAME PAGE */            /* write source data to destination */
  dst++;                                                 /* increment destination pointer    */
  src++;                                                 /* increment source pointer         */
  } ;                                                    /* do it again                      */
*( (UINT16 *)sector_address + 0x555)     = 0x0029;      /* write confirm command          */
  /* poll for completion */
do
{
*((UNIT16 *)sector_address + 0x555)    = 0x0070;      /* enter the Read Status Register command.*/
reg     = *(dst-1);                                           /* Read the Status Register.*/
} while (reg & 0x80) !=0x80);                            /* Loop until bit 7 is 1.*/


/* Example: Write Buffer Abort Reset                     */
*( (UINT16 *)addr + 0x555 ) = 0x00F0;                    /* write buffer abort reset        */
```

8.9.2 Program Suspend/Program Resume Commands

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

After the programming operation has been suspended, the system can read array data from any non-suspended sector and page. 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.

After the Program Resume command is written, the device reverts to programming and the status bits are updated. The system can determine the status of the program operation by reading the Status Register, just as in the standard program operation. See *Status Register* on page 31 for more information.

The system must write the Program Resume command to exit the Program Suspend mode and continue the programming operation. Further writes of the Program Resume command are ignored. Another Program Suspend command can be written after the device has resumed programming.

Software Functions and Sample Code

Table 8.23 Program Suspend

(LLD Function = lld_ProgramSuspendCmd)

Cycle	Operation	Byte Address	Word Address	Data
1	Write	Bank Address	Bank Address	0051h

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

```
/* Example: Program suspend command */
  *( (UINT16 *)base_addr + 0x000 ) = 0x0051;   /* write suspend command           */
```

Table 8.24 Program Resume

(LLD Function = lld_ProgramResumeCmd)

Cycle	Operation	Byte Address	Word Address	Data
1	Write	Bank Address	Bank Address	0050h

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

```
/* Example: Program resume command */
  *( (UINT16 *)base_addr + 0x000 ) = 0x0050;   /* write resume command            */
```

8.9.3 Sector Erase

The sector erase function erases one sector in the memory array. (See Table 12.1 on page 63) 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. Sector Erase requires 2 commands. Each of the Sector Addresses must match, the lower addresses must be correct, and the sector must be unlocked previously by executing the Sector Unlock command and must not be locked by the Sector Lock Range command.

When the Embedded Erase algorithm is complete, the bank returns to idle state 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 the Status Register. See *Status Register* on page 31 for information on these status bits.

Once the sector erase operation has begun, only reading from outside the erase bank, read of Status Register, and the Erase Suspend command are 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 must be reinitiated once the device has returned to idle state, to ensure data integrity.

See *Program/Erase Operations* on page 35 for parameters and timing diagrams.

Software Functions and Sample Code

Table 8.25 Sector Erase

(LLD Function = lld_SectorEraseCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Setup Command	Write	Sector Address + AAAh	Sector Address + 555h	0080h
2	Sector Erase Command	Write	Sector Address + 555h	Sector Address + AAAh	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. Byte address space is used. Refer to the *Spansion Low Level Driver User's Guide* (available on www.spansion.com) for general information on Spansion flash memory software development guidelines.

```
/* Example: Sector Erase Command */
*( (UINT16 *)base_addr + 0x555 ) = 0x0080;   /* write setup command             */
*( (UINT16 *)sector_address + 0x2AA)    = 0x0030;   /* write sector erase command      */
```

8.9.4 Chip Erase

The chip erase function erases the complete memory array. (See Table 12.1 on page 63). 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 chip erase, all locations within the device contain FFFFh. The system is not required to provide any controls or timings during these operations. Chip Erase requires 2 commands. Each of the Sector Addresses must match, the lower addresses must be correct, and Sector 0 must be unlocked previously by executing the Sector Unlock command. If any sector has been locked by the Sector Lock Range command, the Chip Erase command will not start.

When the Embedded Erase algorithm is complete, the device returns to idle state and addresses are no longer latched. Note that while the Embedded Erase operation is in progress, the system can not read data from the device. The system can determine the status of the erase operation by reading the Status Register. See *Status Register* on page 31 for information on these status bits.

Once the chip erase operation has begun, only a Status Read, Hardware RESET or Power cycle are valid. All other commands are ignored. However, note that a Hardware Reset or Power Cycle immediately terminates the erase operation. If that occurs, the chip erase command sequence must be reinitiated once the device has returned to idle state, to ensure data integrity.

See *Program/Erase Operations* on page 65 for parameters and timing diagrams.

Software Functions and Sample Code

Table 8.26 Chip Erase

(LLD Function = lld_ChipEraseCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Setup Command	Write	Base + AAAh	Base + 555h	0080h
2	Chip Erase Command	Write	Base + 555h	Base + AAAh	0010h

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

```
/* Example: Chip Erase Command */
/* Note: Cannot be suspended   */
*( (UINT16 *)base_addr + 0x555 ) = 0x0080;   /* write setup command             */
*( (UINT16 *)base_addr + 0x2AA ) = 0x0010;   /* write chip erase command        */
```

8.9.5 Erase Suspend/Erase Resume Commands

The Erase Suspend command allows the system to interrupt a sector erase operation and then read data from, or program data to, the device. This command is valid only during the sector erase operation. The Erase Suspend command is ignored if written during the chip erase operation.

When the Erase Suspend command is written during the sector erase operation, the device requires a maximum of t_{ESL} (erase suspend latency) to suspend the erase operation and update the status bits.

After the erase operation has been suspended, the bank enters the erase-suspend mode. The system can read data from or program data to the device. Reading at any address within erase-suspended sectors produces undetermined data. The system can read the Status Register to determine if a sector is actively erasing or is erase-suspended. Refer to *Status Register* on page 31 for information on these status bits.

After an erase-suspended program operation is complete, the bank returns to the erase-suspend mode. The system can determine the status of the program operation by reading the Status Register, just as in the standard program operation.

To resume the sector erase operation, the system must write the Erase Resume command. The device will revert to erasing and the status bits will be updated. Further writes of the Resume command are ignored. Another Erase Suspend command can be written after the chip has resumed erasing.

Software Functions and Sample Code

Table 8.27 Erase Suspend

(LLD Function = lld_EraseSuspendCmd)

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

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

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

Table 8.28 Erase Resume

(LLD Function = lld_EraseResumeCmd)

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

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

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

8.9.6 Accelerated Program/Sector Erase

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

The accelerated write buffer program and sector erase functions must not be used more than 50 times per sector. In addition, accelerated write buffer program and sector erase should be performed at room temperature (30°C 10°C).

If the system asserts V_{HH} on ACC, the device automatically uses the higher voltage on the input to reduce the time required for program and erase operations. Removing V_{HH} from the ACC input, upon completion of the embedded program or erase operation, returns the device to normal operation.

- Simultaneous operations are not supported while ACC is at V_{HH}. The ACC pin must not be at V_{HH} for operations other than accelerated write buffer programming, accelerated sector erase, and status register read or device damage may result.
- The ACC pin must not be left floating or unconnected; inconsistent behavior of the device may result.
- There is a minimum of 100 ms required between accelerated write buffer programming and a subsequent accelerated sector erase.

8.10 Handshaking

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

When either CE# input is Low, the RDY output signal is actively driven. When both of the CE# inputs are High the RDY output is high-impedance. When either CE# input and OE# input is Low, the DQ15-DQ0 output signals are actively driven. When both of the CE# inputs are High, or the OE# input is High, the DQ15-
DQ0 outputs are high-impedance.

When the device is operated in synchronous mode, and OE# is low (active), the initial word of burst data becomes available after the rising edge of the RDY. CR.8 in the Configuration Register allows the host to specify whether RDY is active at the same time that data is ready, or one cycle before data is ready (see Table 8.12 on page 30).

When the device is operated in asynchronous mode, RDY will be high when CE# is low (active).

8.11 Hardware Reset

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

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.

See Figure 11.6 for timing diagrams

8.12 Software Reset

Software reset is part of the command set (see Table 12.1 on page 63) that also returns the device to idle state and must be used for the following conditions:

1. Exit ID/CFI mode
2. Exit Secure Silicon Region mode
3. Exit Configuration Register mode
4. Exit SSR Lock mode

Reset commands are ignored once programming/erasure has begun until the operation is complete.

Software Functions and Sample Code

Table 8.29 Reset

(LLD Function = lld_ResetCmd)

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

Note:
Base = Base Address.

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

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

9. Sector Protection/Unprotection

The Sector Protection/Unprotection feature disables or enables programming or erase operations in one or multiple 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.

9.1 Sector Lock/Unlock Command

The Sector Lock/Unlock command sequence allows the system to protect all sectors from accidental writes or, unprotect one sector to allow programming or erasing of the sector. When the device is first powered up, all sectors are unlocked. To lock all sectors (enter protected mode), a Sector Lock/Unlock command must be issued to any Sector Address. Once this command is issued, only one sector at a time can be unlocked until power is cycled. To unlock a sector, the system must write the Sector Lock/Unlock command sequence. Two cycles are first written: addresses are xAAAh and x555 with data 60h. During the third cycle, the sector address (SLA) and unlock command (60h) are written, while specifying with address A6 whether that sector should be locked (A6 = V_{IL}) or unlocked (A6 = V_{IH}).

A Program or Erase operation will check the unlocked Sector Address only at the beginning of the Program or Erase operation. It is not necessary to keep the sector being Programmed or Erased locked during the operation. The system can change the unlocked Sector after programming or erasing the sector has begun. An Erase Resume or Program Resume command does not check the value of the unlocked Sector.

If A6 is set to V_{IL},then all sectors in the array will be locked. Only one sector at a time can be unlocked.

If a Sector Lock/Unlock command is issued to a sector that is protected by the Sector Lock Range command, all sectors in the part will be locked.

9.2 Sector Lock Range Command

This command allows a range of sectors to be protected from program or erase (locked) until a hardware reset or power is removed from the device. Once this command is issued, all sectors are protected and the Sector Lock/Unlock command is ignored for the selected the range of sectors. Sectors outside of the selected range must be unlocked one sector at a time using the Sector Unlock command in order to be erased/programmed.

Two cycles are first written: addresses are xAAAh and data is 60h. During the third cycle, the sector address (SLA) and load sector address command (61h) is written. This cycle sets the lower sector address of the range. During the fourth cycle, the sector address (SLA) and load sector address command (61h) is written. This cycle sets the upper sector address of the range. The addresses reference a large sector address range (128 KB). If a sector address matches the location of the four small sectors, all of the small sectors will be protected as a group. The sectors selected by the lower and upper address, as well as all sectors between these sectors, are protected from program and erase until a hardware reset or power is removed. If the lower and upper sector addresses are for the same sector then only that one sector is locked. flash address input A6 (system byte address bit a7) during both address cycles must be zero (A6 = V_{IL}) for the addresses to be accepted as valid.

If the first sector address cycle contains an address which is higher than the second sector address cycle, then the command sequence will be invalid. If A6 is set to one (A6 = V_{IH}) on either address cycle, the command sequence will disable subsequent Sector Lock Range commands.

A valid Sector Lock Range command sequence is accepted only once after a Hardware Reset or initial power up. Additional Sector Lock Range commands will be ignored.

If a Sector Unlock command tries to unlock a Sector within the Sector Lock Range, the Sector will remain in locked state. Similarly, if a Sector that is currently unlocked by the Sector Unlock command is overlapped by a subsequent Sector Lock Range, that sector will be locked and program erase operations to that region will be ignored.

This command is generally used by trusted boot code. After power on reset boot code has the option to check for any need to update sectors before locking them for the remainder of power on time. Once boot code is satisfied with the content of sectors to be protected the Sector Lock Range command is used to lock sectors against any program or erase during normal system operation. This adds an extra layer of protection for critical data that must be protected against accidental or malicious corruption. Yet, maintains flexibility for trusted boot code to perform occasional updates of the data. It is important to issue the Sector Lock Range command even if no sectors are to be protected so that sectors that should remain available for update cannot be later locked by accidental or malicious code behavior.

9.3 Hardware Data Protection Methods

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

9.3.1 ACC Method

Once the **ACC** input is set to V_{IL}, all program and erase functions are disabled and hence all Sectors (including the Secure Silicon Region and SSR Lock) are protected. ACC does not prevent programming (writing) of the configuration register.

9.3.2 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. 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}.

9.3.3 Write Pulse Glitch Protection

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

9.3.4 Power-Up Write Inhibit

If CE# = RESET# = V_{IL} and OE# = V_{IH} during power up, the device does not accept write commands. The internal state machine is automatically reset to the idle state on power-up.

9.4 SSR Lock

The SSR Lock consists of two bits. The Customer Secure Silicon Region Protection Bit is in location 0. The Factory Secure Silicon Region Protection Bit is in location 1. All other bits in this location return “1.”

If the Secure Silicon Region Protection Bit is set to “0,” the Customer Secure Silicon Region is protected and can not be programmed. If this bit is set to “1,” the Customer Secure Silicon Region is available for programming. Once this area has been programmed, the SSR Lock bit 0 should be programmed to “0.”

Similarly the Factory Secure Silicon Region Protection Bit is set to “0” when protected. This area is always programmed and protected at the Spansion factory.

9.5 Secure Silicon Region

The Secure Silicon Region provides an extra flash memory region that may be programmed once and permanently protected from further programming or erase.

- Reads can be performed in the Asynchronous or Synchronous mode.
- Sector address supplied during the Secure Silicon Entry command selects the flash memory array sector that is overlaid by the Secure Silicon Region address map.
- Continuous burst mode reads within Secure Silicon Region wrap from address FFh back to address 00h.
- Reads outside of the overlaid sector return memory array data.
- The Secure Silicon Region is not accessible when the device is executing an Embedded Algorithm (nor during Program Suspend, Erase Suspend, or while another AOS is active).
- See the Secure Silicon address map for address range of this area.

9.5.1 Factory Secure Silicon Region

The Factory Secure Silicon Region is always protected when shipped from the factory and has the Factory SSR Lock Bit (bit 1) permanently set to a *Zero*. 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.

9.5.2 Customer Secure Silicon Region

The Customer Secure Silicon Region is shipped unprotected, Customer SSR Lock Bit (bit 0) set to a *One*. allowing customers to utilize that sector in any manner they choose.

The Customer Secure Silicon Region can be read any number of times, but each word can be programmed only once and the region locked only once. The Customer Secure Silicon Region lock must be used with caution as once locked, there is no procedure available for unlocking the Customer Secure Silicon Region area and none of the bits in the Customer Secure Silicon Region memory space can be modified in any way. The Customer Indicator Bit is located in the SSR Lock at bit location 0.

Once the Customer Secure Silicon Region area is protected, any further attempts to program in the area will fail with status indicating the area being programmed is protected.

9.5.3 Secure Silicon Region Entry and Exit Command Sequences

The system can access the Secure Silicon Region region by issuing the one-cycle Enter Secure Silicon Region Entry command sequence from the IDLE State. The device continues to have access to the Secure Silicon Region region until the system issues the Exit Secure Silicon Region command sequence, performs a Hardware RESET, or until power is removed from the device.

See Command Definition Table [Secure Silicon Region Command Table, Appendix Table 12.1 on page 63 for address and data requirements for both command sequences.

The Secure Silicon Region Entry Command allows the following commands to be executed

- Read customer and factory Secure Silicon Regions
- Program the customer Secure Silicon Region
- Read data out of all sectors not re-mapped to Secure Silicon Region
- Secure Silicon Region Exit

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

Table 9.1 Secured Silicon Region Entry

(LLD Function = lld_SecSiSectorEntryCmd)

Cycle	Operation	Byte Address	Word Address	Data
Entry Cycle	Write	Base + AAAh	Base + 555h	0088h

Note:
Base = Base Address.

```
/* Example: SecSi Sector Entry Command Byte Address*/
*( (UINT16 *)base_addr + 0x555 ) = 0x0088;   /* write Secsi Sector Entry Cmd    */
```

Table 9.2 Secured Silicon Region Program

(LLD Function = lld_ProgramCmd)

Cycle	Operation	Byte Address	Word Address	Data
Program Setup	Write	Base + AAAh	Sector Address + 555h	0025h
Write Word Count	Write	Program Address + 555h	Sector Address + 2AA	Word Count (N–1)h

Table 9.2 Secured Silicon Region Program

(LLD Function = lld_ProgramCmd)

Cycle	Operation	Byte Address	Word Address	Data
Number of words (N) loaded into the write buffer can be from 1 to 32 words.				
Load Buffer Word N	Write	Program Address, Word N		Word N
Write Buffer to Flash	Write	Sector Address + AAAh	Sector Address + 555h	0029h

Note:
Base = Base Address.

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

Table 9.3 Secured Silicon Region Exit

(LLD Function = lld_SecSiSectorExitCmd)

Cycle	Operation	Byte Address	Word Address	Data
Exit Cycle	Write	Any Address	Any Address	00F0h

Note:
Base = Base Address.

```
/* Example: SecSi Sector Exit Command */
*( (UINT16 *)XXX ) = 0x00F0;   /* write SecSi Sector Exit cycle */
```

10. Power Conservation Modes

10.1 Standby Mode

In the standby mode current consumption is greatly reduced, and the outputs (DQ15-DQ0) 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} or t_{IA}) 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 48 represents the standby current specification

10.2 Automatic Sleep Mode

The automatic sleep mode minimizes flash device energy consumption while in asynchronous mode and while the device is not in a suspended state. 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 (t_{ACC} or t_{PACC}) 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. I_{CC6} in *DC Characteristics* on page 48 represents the automatic sleep mode current specification.

10.3 Output Disable (OE#)

When the OE# input is at V_{IH}, output (DQ15-DQ0) from the device is disabled and placed in the high impedance state. RDY is not controlled by OE#.

11. Electrical Specifications

11.1 Absolute Maximum Ratings

Table 11.1

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

Notes

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

Figure 11.1 Maximum Negative Overshoot Waveform

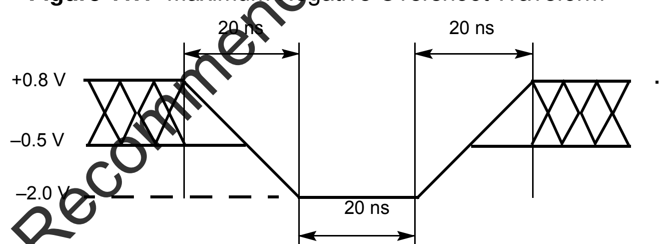
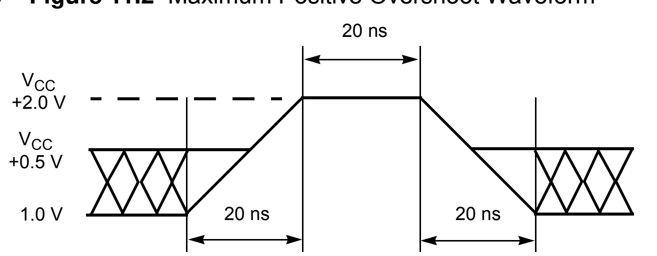


Figure 11.2 Maximum Positive Overshoot Waveform



11.2 Operating Ranges

Table 11.2

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

Table 11.2

V_{IO} Supply Voltages	+1.70 V to +1.95 V	
	$V_{CC(min)} \geq V_{IO(min)}$ - 200 mV	

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

11.3 DC Characteristics

11.3.1 CMOS Compatible

Parameter	Description	Test Conditions (Notes 1 & 2)		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	66 MHz		24	33	mA
			83 MHz		26	36	mA
			104 MHz		33	44	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 16	66 MHz		24	35	mA
			83 MHz		26	38	mA
			104 MHz		30	40	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = Continuous	66 MHz		28	39	mA
			83 MHz		30	42	mA
			104 MHz		32	44	mA
I_{IO1}	V_{IO} Non-active Output	OE# = V_{IH} , RDY = Tri-State			20	30	µA
I_{IO2}	V_{IO} Standby	CE# = RESET# = V_{CC} ± 0.2V			2	3	µA
I_{CC1}	V_{CC} Active Asynchronous Read Current	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}	10 MHz		40	80	mA
			5 MHz		20	40	mA
			1 MHz		10	20	mA
I_{CC2}	V_{CC} Active Write Current (3) (7)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}	V_{ACC}		1	5	µA
			V_{CC}		20	40	mA
I_{CC3}	V_{CC} Standby Current	CE# = RESET# = V_{CC} ± 0.2 V	V_{ACC}		1	5	µA
			V_{CC}		20	50	µA
I_{CC4}	V_{CC} Reset Current	RESET# = V_{IL}, CLK = V_{IL}			150	250	µA
I_{CC5}	V_{CC} Active Current Simultaneous Operation (Read While Write) (6)	CE# = V_{IL}, OE# = V_{IH}, ACC = V_{IH}	66 MHz, Continuous Burst		50	60	mA
			83 MHz, Continuous Burst		50	60	mA
			104 MHz, Continuous Burst		52	60	mA
I_{CC6}	V_{CC} Sleep Current (4)	CE# = V_{IL}, OE# = V_{IH}			20	50	µA
I_{CC7}	V_{CC} Active Page Read Current	OE# = V_{IH}, 8 word Page Read @ 10 MHz			10	15	mA
I_{ACC}	Accelerated Program Current (5)	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	V_{IO} = 1.8 V		–0.2		0.4	V
V_{IH}	Input High Voltage	V_{IO} = 1.8 V		V_{IO} – 0.4		V_{IO} + 0.4	
V_{OL}	Output Low Voltage	I_{OL} = 100 µA, $V_{CC} = V_{CC\ min} = V_{IO}$				0.1	V
V_{OH}	Output High Voltage	I_{OH} = –100 µA, $V_{CC} = V_{CC\ min} = V_{IO}$		V_{IO} – 0.1			V
V_{HH}	Voltage for Accelerated Program			8.5		9.5	V
V_{LKO}	Low V_{CC} Lock-out Voltage			1.0		1.1	V

Notes
1. *Maximum I_{CC} specifications are tested with $V_{CC} = V_{CC}max$.*
2. *$V_{CC} = V_{IO}$*
3. *I_{CC} active while Embedded Erase or Embedded Program is in progress.*
4. *Device enters automatic sleep mode when addresses are stable for t_{ACC} + 20 ns. Typical sleep mode current is equal to I_{CC3}.*
5. *Total current during accelerated programming is the sum of V_{ACC} and V_{CC} currents.*
6. *I_{CC5} apples while reading the status register during program and erase operations*
7. *Effect of status register polling during write not included.*

11.4 Capacitance

Symbol	Description	Test Condition		Min.	Typ.	Max.	Unit
C_{IN}	Input Capacitance (Address, CE#, OE#, WE#, AVD#, WE#, CLK, RESET#)	$V_{IN} = 0$	Single CE	2.0	4.5	6.0	pF
			Dual CE	4.0	9.0	12.0	pF
C_{OUT}	Output Capacitance (DQ, RDY)	$V_{OUT} = 0$	Single CE	2.0	4.5	6.0	pF
			Dual CE	4.0	9.0	12.0	pF

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

11.5 AC Test Conditions

Operating Range		
Input level		0.0 to V_{IO}
Input comparison level		$V_{IO}/2$
Output data comparison level		$V_{IO}/2$
Load capacitance (C_L)		30 pF
Transition time (t_T) (input rise and fall times)	66 MHz	3.00 ns
	83 MHz	2.40 ns
	104 MHz	1.85 ns
Transition time (t_T) (CLK input rise and fall times)	66 MHz	3.00 ns
	83 MHz	2.40 ns
	104 MHz	1.85 ns

Figure 11.3 Input Pulse and Test Point

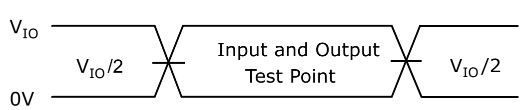
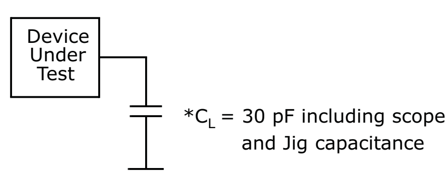
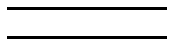
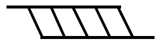
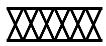
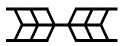


Figure 11.4 Output Load



Output Load

11.6 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)

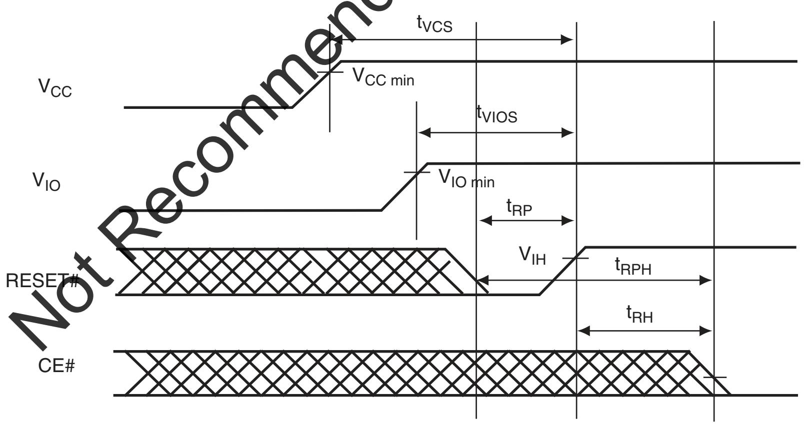
11.7 V_{CC} Power Up

Table 11.3 V_{CC} Power-up

Parameter	Description	Test Setup	Speed	Unit
t_{VCS}	V_{CC} Setup Time	Min	300	µs
t_{VIOS}	V_{IO} Setup Time	Min	300	µs

Notes
1. *RESET# must be high after V_{CC} and V_{IO} are higher than V_{CC} minimum.*
2. *$V_{CC} \geq V_{IO} - 200$ mV during power-up.*
3. *V_{CC} and V_{IO} ramp rate could be non-linear.*
4. *V_{CC} and V_{IO} are recommended to be ramped up simultaneously.*
5. *Time between a short power off to power on should be >10 ms when system is at 0°C or below.*

Figure 11.5 V_{CC} Power-up Diagram

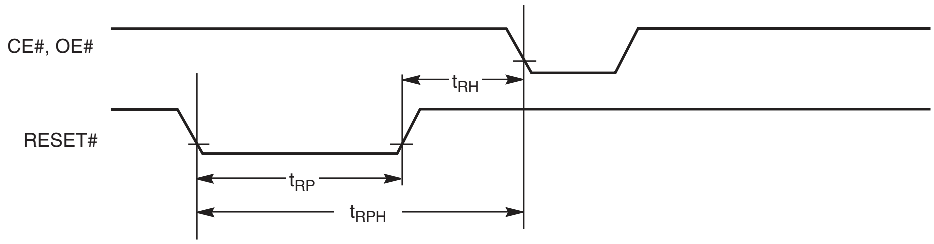


11.7.1 Hardware Reset (Reset#)

Table 11.4 Warm-Reset

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

Figure 11.6 Reset Timings



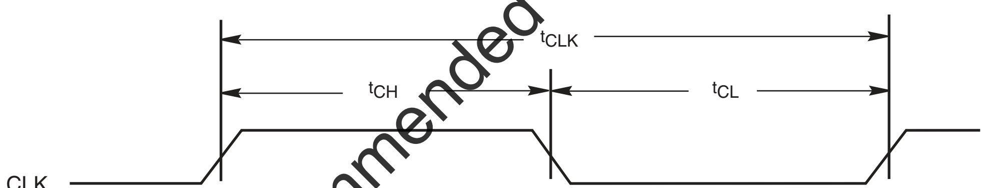
11.8 CLK Characterization

Parameter	Description		104 MHz	Unit
f_{CLK}	CLK Frequency	Max	104	MHz
		Min	DC (Note 1)	
t_{CLK}	CLK Period	Min	9.6	ns
t_{CL}/t_{CH}	CLK Low/High Time	Min	0.45 t_{CLK}	ns

Note

1. *DC for operations other than continuous and 16 word (32 byte) synchronous burst read. See Section 11.9.1, AC Characteristics–Synchronous Burst Read on page 52.*
2. *Clock jitter of +/- 5% is permitted.*

Figure 11.7 CLK Characterization



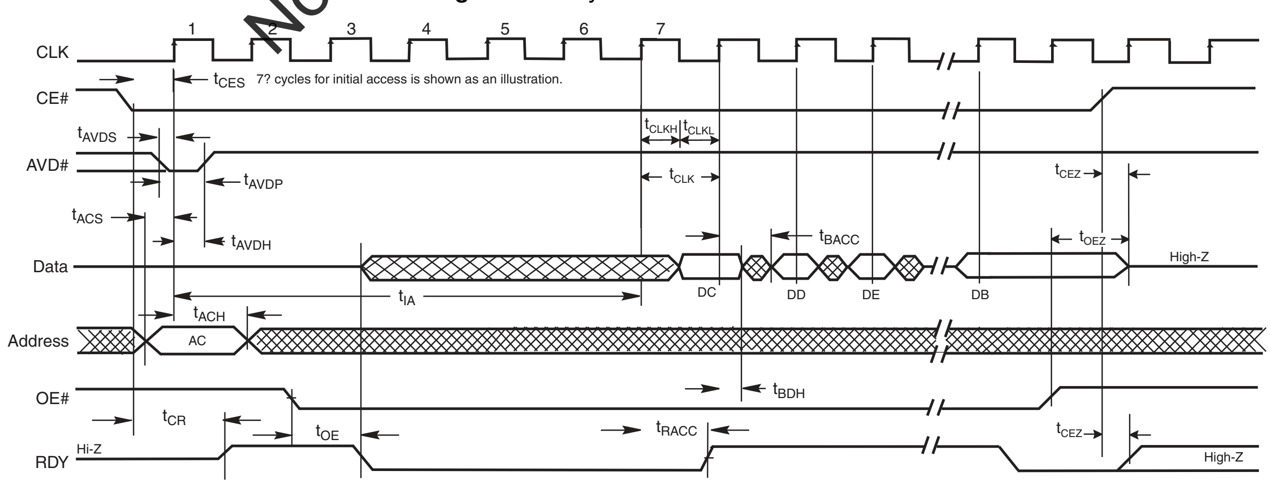
11.9 AC Characteristics

11.9.1 AC Characteristics–Synchronous Burst Read

Parameter (Notes)	Symbol		66	83	104	Unit
Clock Frequency	CLK	Min	DC (0) for operations other than continuous and 32 byte synchronous burst. 120 in 32 Byte burst 1000 in continuous burst			KHz
Clock Cycle	t_{CLK}	Min	15	12.0	9.62	ns
CLK High or Low Time	$t_{CLKH/L}$	Min	0.45 t_{CLK}			ns
Internal Access Time	t_{IA}	Max	75	75	75	ns
Burst Access Time Valid Clock to Output Delay	t_{BACC}	Max	11.2	9	7.6	ns
AVD# Setup Time to CLK	t_{AVDS}	Min	5	5	5	ns
AVD# Hold Time from CLK	t_{AVDH}	Min	3	3	3	ns
Address Setup Time to CLK	t_{ACS}	Min	4	4	4	ns
Address Hold Time from CLK	t_{ACH}	Min	6	5	5	ns
Data Hold Time from Next Clock Cycle	t_{BDH}	Min	3	3	2	ns
Output Enable to Data	t_{OE}	Max	15	15	15	ns
CE# Disable to Output High-Z (2)	t_{CEZ}	Max	10	10	10	ns
OE# Disable to Output High-Z (2)	t_{OEZ}	Max	10	10	10	ns
CE# Setup Time to CLK	t_{CES}	Min	4	4	4	ns
CLK to RDY valid	t_{RACC}	Max	11.2	9	7.6	ns
CE# low to RDY valid	t_{CR}	Max	10	10	10	ns
AVD# Pulse	t_{AVDP}	Min	6	6	6	ns

Notes
1. *Not 100% tested.*
2. *If OE# is disabled before CE# is disabled, the output goes to High-Z by t_{OEZ}.*
   If CE# is disabled before OE# is disabled, the output goes to High-Z by t_{CEZ}.
   If CE# and OE# are disabled at the same time, the output goes to High-Z by t_{OEZ}.
3. *Synchronous Access Time is calculated using the formula (# of WS - 1)*(clock period) + (t_{BACC}).*
4. *AVD can not be low for 2 subsequent CLK cycles.*

Figure 11.8 Synchronous Read Mode



11.9.2 AC Characteristics–Asynchronous Read

Parameter	Symbol	Min	Max	Unit
Access Time from CE# Low	t_{CE}	–	80	
Asynchronous Access Time from address valid	t_{ACC}	–	80	
Read Cycle Time	t_{RC}	80	–	
AVD# Low Time	t_{AVDP}	6	–	
Address Setup to rising edge of AVD#	t_{AAVDS}	5	–	
Address Hold from rising edge of AVD#	t_{AAVDH}	3.5	–	
Output Enable to Output Valid	t_{OE}	–	15	
CE# Setup to AVD# falling edge	t_{CAS}	4	–	ns
CE# Disable to Output & RDY High-Z (Note 1)	t_{CEZ}	–	10	
OE# Disable to Output High-Z (Note 1)	t_{OEZ}	–	10	
AVD# High to OE# Low	t_{AVDO}	0	–	
CE# low to RDY valid	t_{CR}	–	10	
WE# Disable to AVD# Enable	t_{WEA}	t_{CLK}	–	
WE# Disable to OE# Enable	t_{OEH}	4	–	
Intra Page Access Time	t_{PACC}	–	15	

Notes

1. *Not 100% tested.*
2. *If OE# is disabled before CE# is disabled, the output goes to High-Z by t_{OEZ}.*
   If CE# is disabled before OE# is disabled, the output goes to High-Z by t_{CEZ}.
   If CE# and OE# are disabled at the same time, the output goes to High-Z by t_{OEZ}.

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

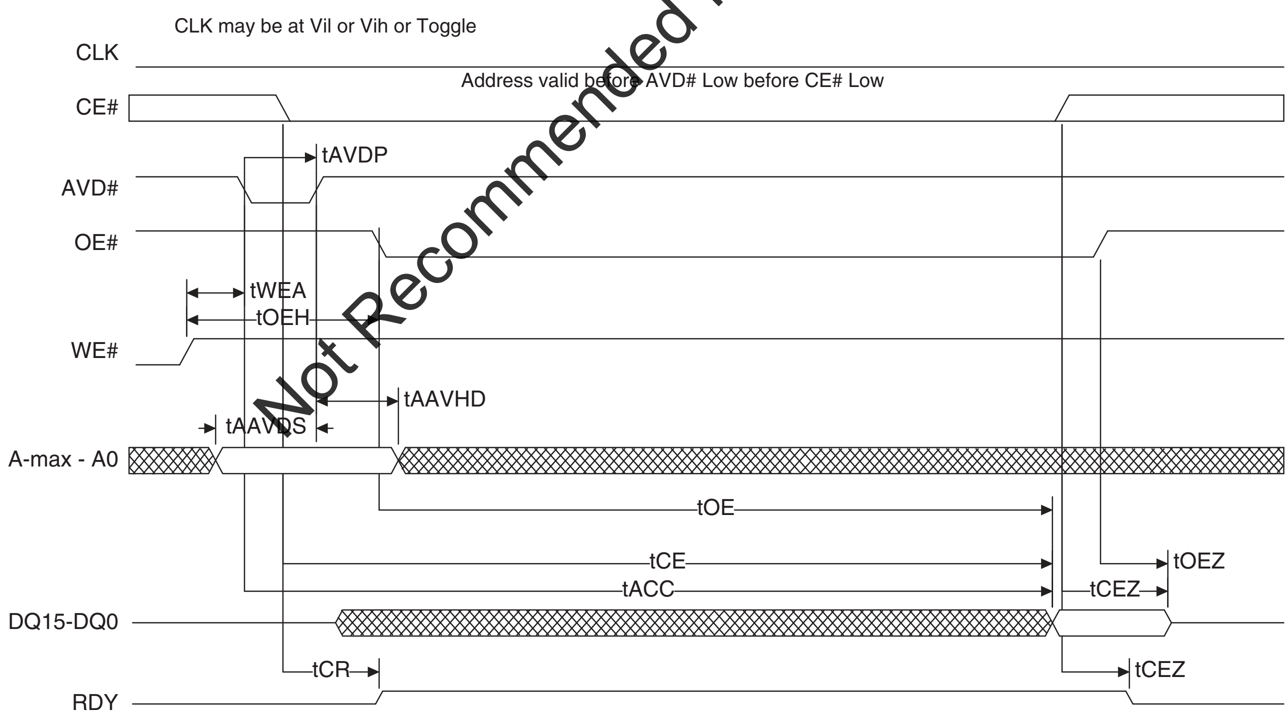


Figure 11.10 Asynchronous Read Mode (AVD# Toggling - Case 2)

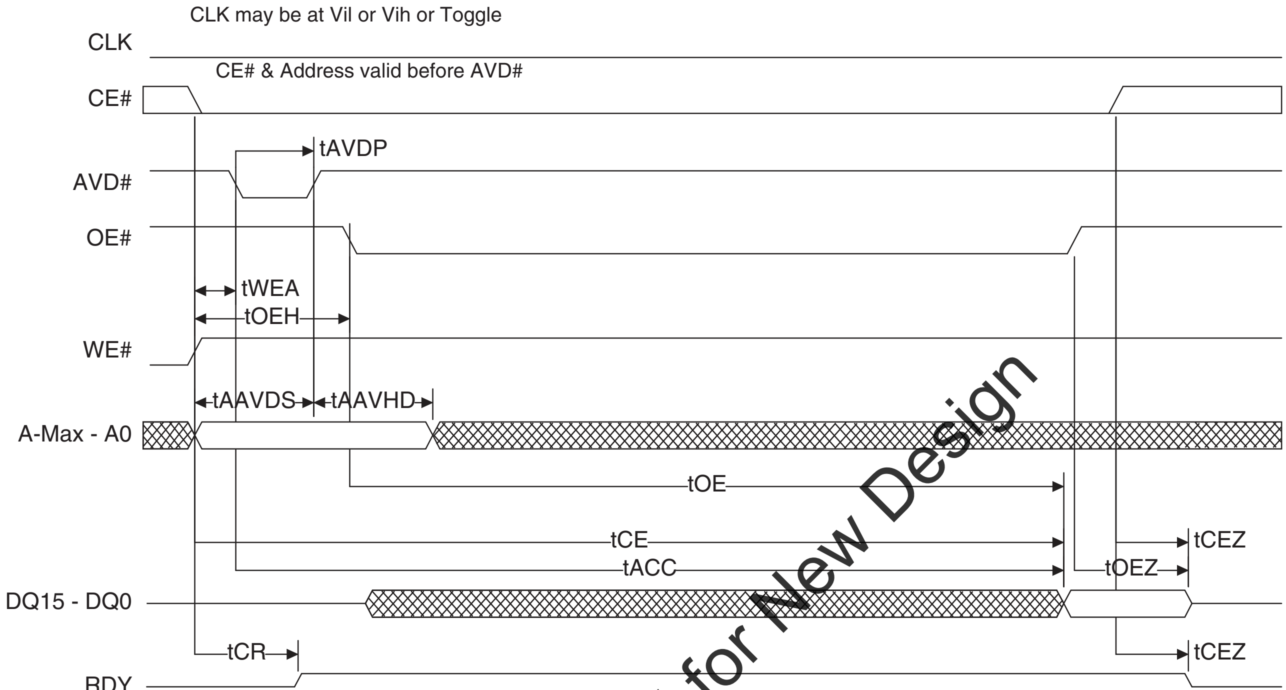


Figure 11.11 Asynchronous Read Mode (AVD# Toggling - Case 3)

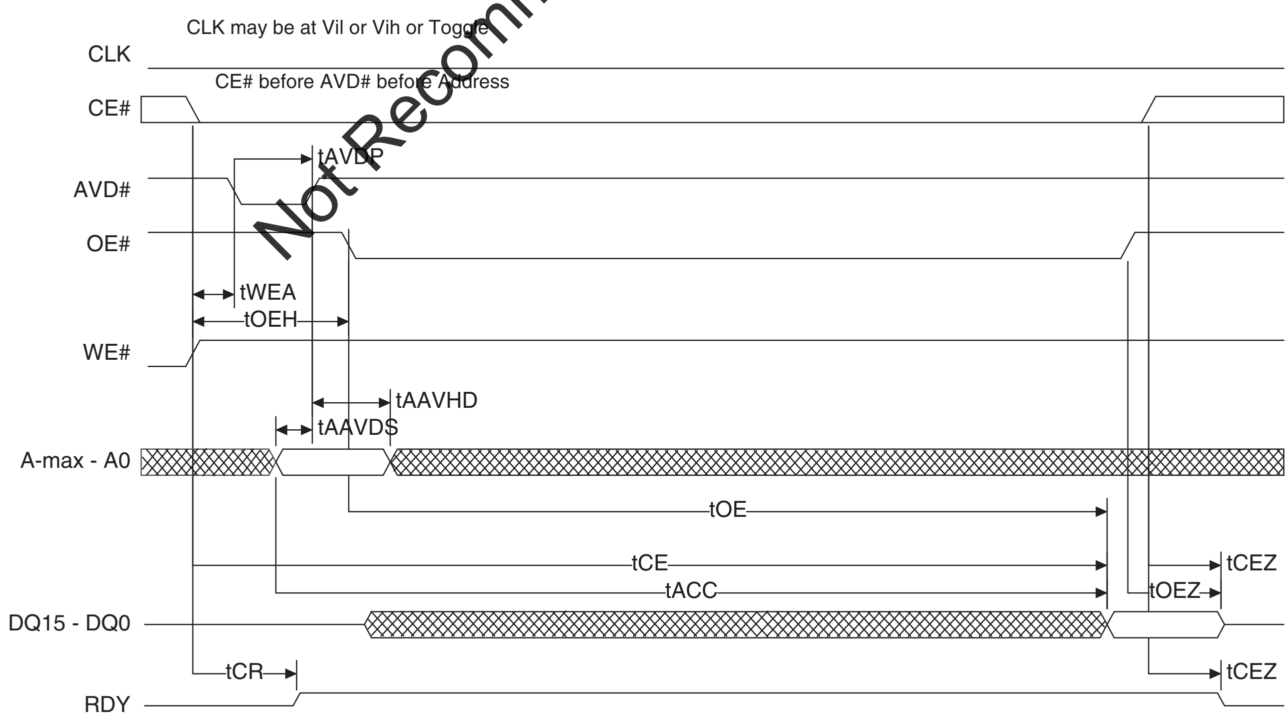
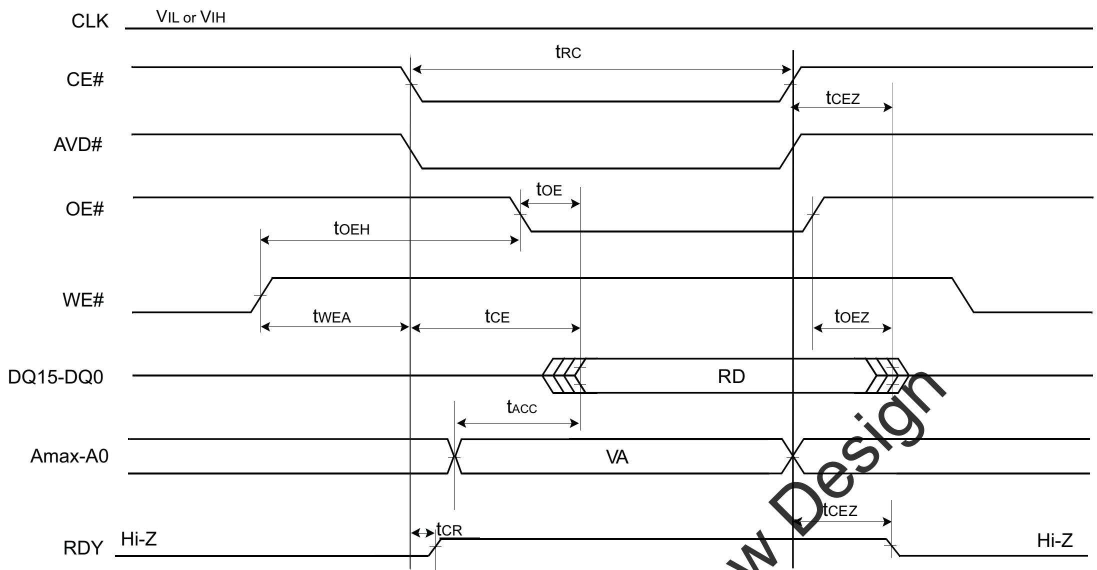


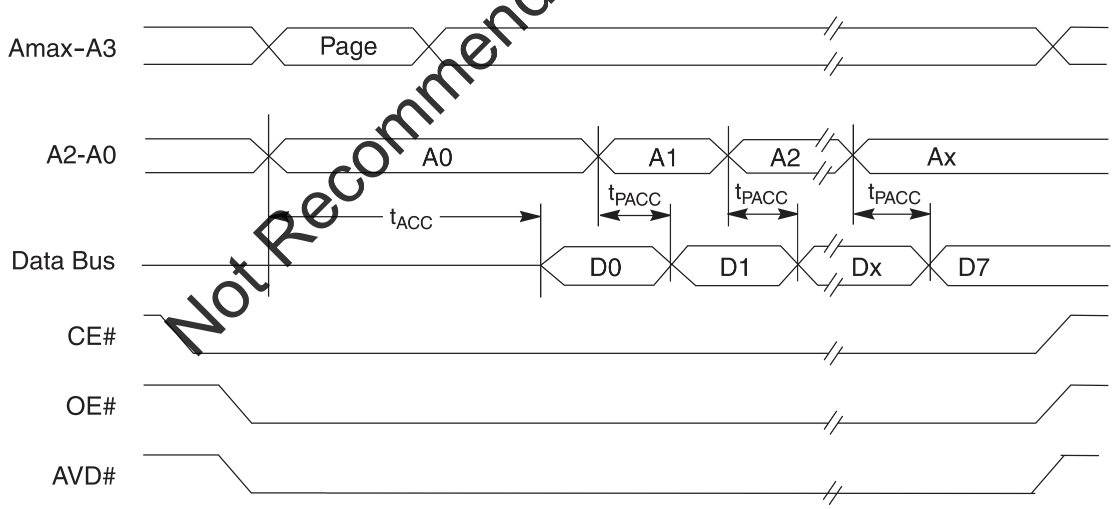
Figure 11.12 Asynchronous Read Mode (AVD# tied to CE#)



Notes

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

Figure 11.13 Asynchronous Page Mode Read



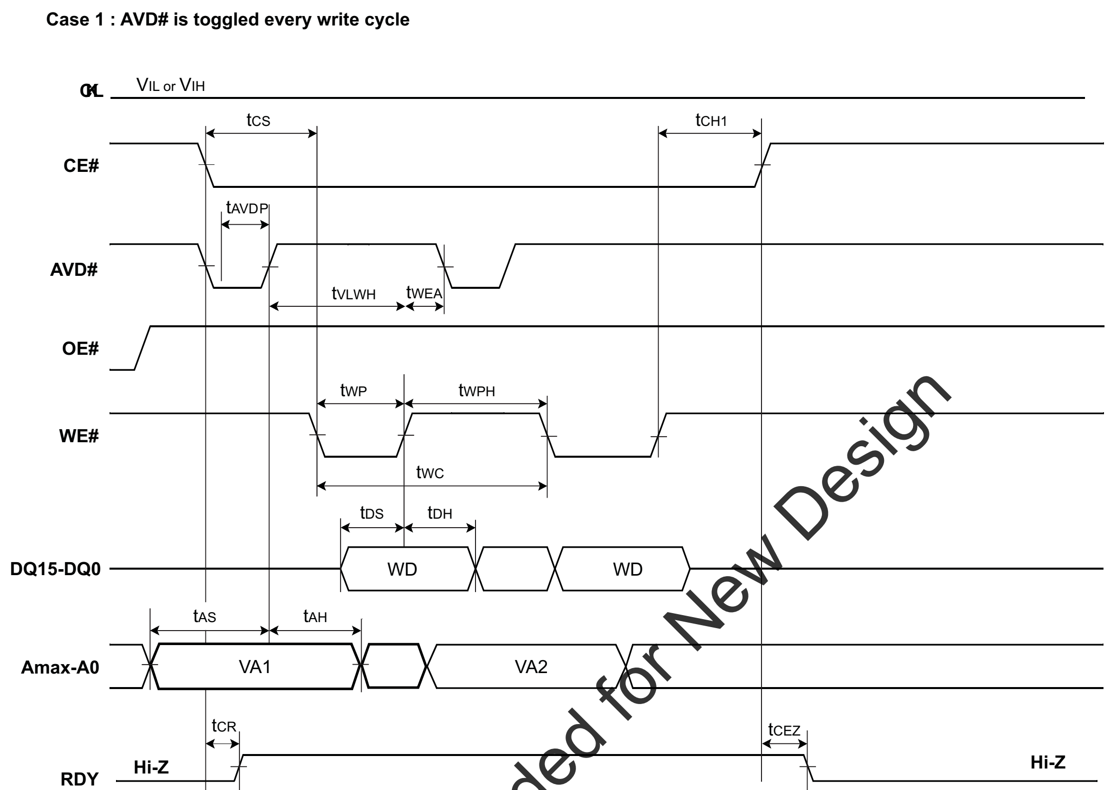
Note

RA = Read Address, RD = Read Data.

11.9.3 AC Characteristics–Asynchronous Write Operation

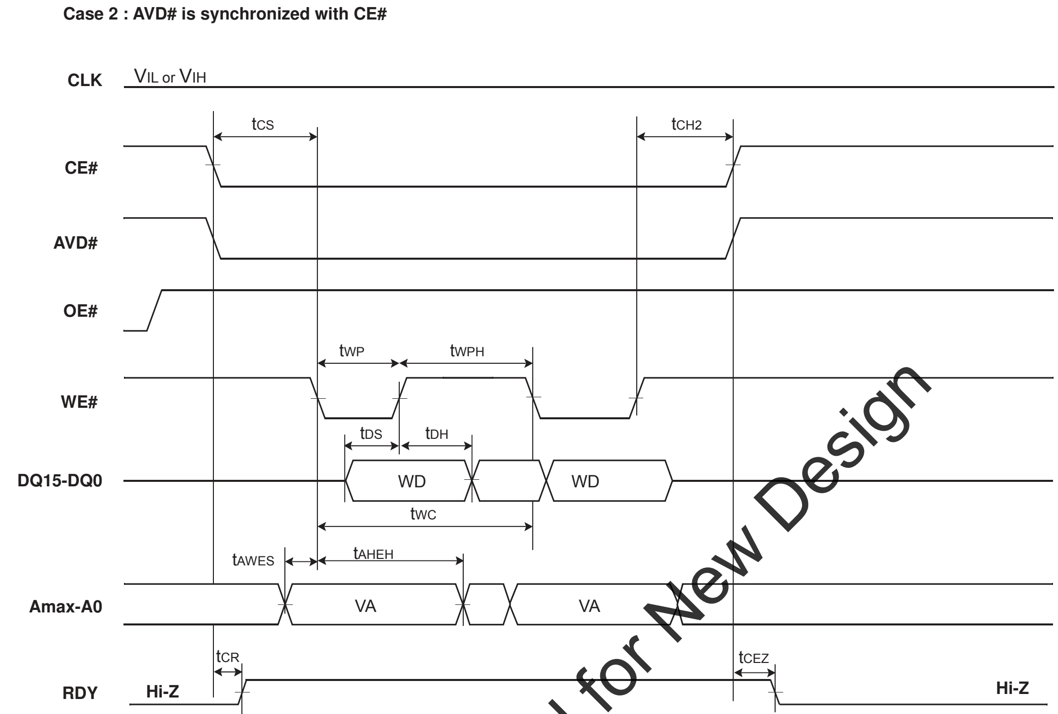
Parameter		Symbol	Min	Typ	Max	Unit
WE# Cycle Time		t_{WC}	60	–	–	ns
AVD# low pulse width		t_{AVDP}	6	–	–	ns
Address Setup to rising edge of AVD#		t_{AAVDS}	5	–	–	ns
Address Setup to falling edge of WE#		t_{AWES}	5	–	–	ns
Address Hold from rising edge of AVD#		t_{AAVDH}	3.5	–	–	ns
Address Hold from falling edge of WE#		t_{AWEH}	3.5	–	–	ns
Read Recovery time before Write		t_{GHWL}	0	–	–	ns
Data Setup to rising edge of WE#		t_{DS}	20	–	–	ns
Data Hold from rising edge of WE#		t_{DH}	0	–	–	ns
CE# Setup to falling edge of WE#		t_{CS}	4	–	–	ns
CE# Hold from rising edge of WE#	AVD# toggled	t_{CH1}	0	–	–	ns
CE# Hold from rising edge of WE#	AVD# = CE#	t_{CH2}	0	–	–	ns
WE# Pulse Width		t_{WP}	25	–	–	ns
WE# Pulse Width High		t_{WPH}	20	–	–	ns
AVD# Disable to WE# Disable		t_{VLWH}	23.5	–	–	ns
WE# Disable to AVD# Enable		t_{WEA}	7.5	–	–	ns
CE# low to RDY valid		t_{CR}	–	–	10	ns
CE# Disable to Output High Z		t_{CEZ}	–	–	10	ns
OE# Disable to WE# Enable		t_{WEH}	4	–	–	ns
Erase Suspend Latency		t_{ESL}	–	–	30	µs
Program Suspend Latency		t_{PSL}	–	–	30	µs
Latency between Read and Write Operations		t_{SRW}	0	–	–	ns
Erase Resume to Erase Suspend		t_{ERS}	30	–	–	µs
Program Resume to Program Suspend		t_{PRS}	30	–	–	µs

Figure 11.14 Latched Asynchronous Write Mode (AVD# Toggling)



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

Figure 11.15 Asynchronous Write Mode (AVD# Tied to CE#)

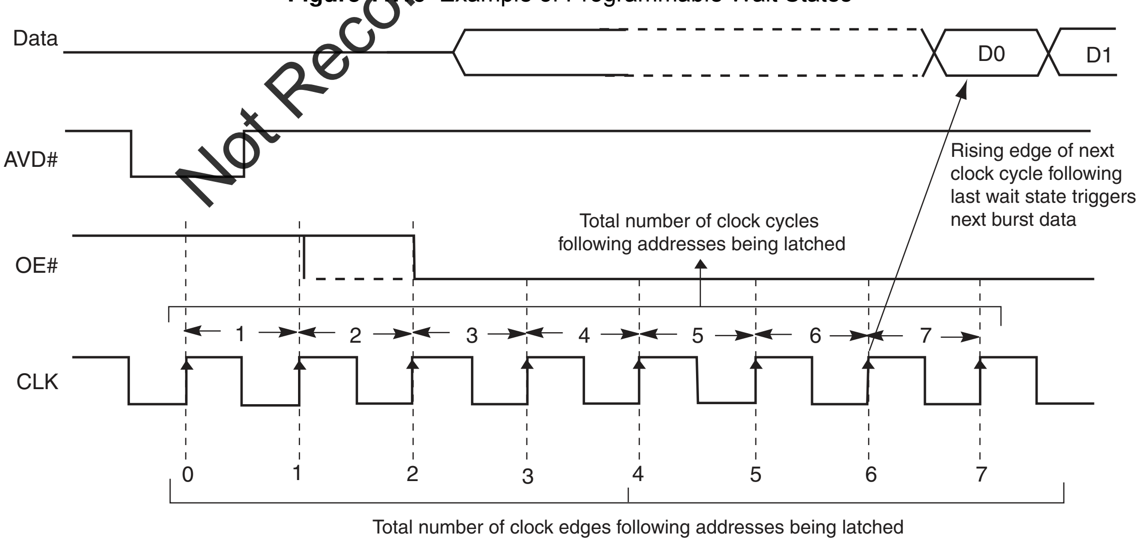


Notes

1. *VA = Valid Read Address, WD = Write Data.*
2. *Addresses latched by falling edge of WE#.*

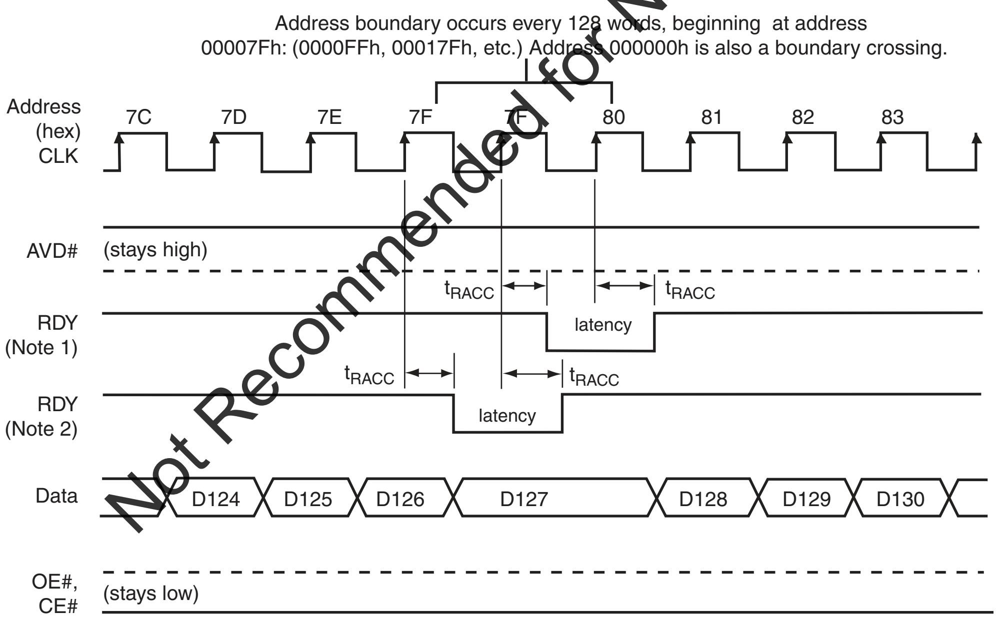
11.9.4 Wait State Configuration Register Setup

Figure 11.16 Example of Programmable Wait States



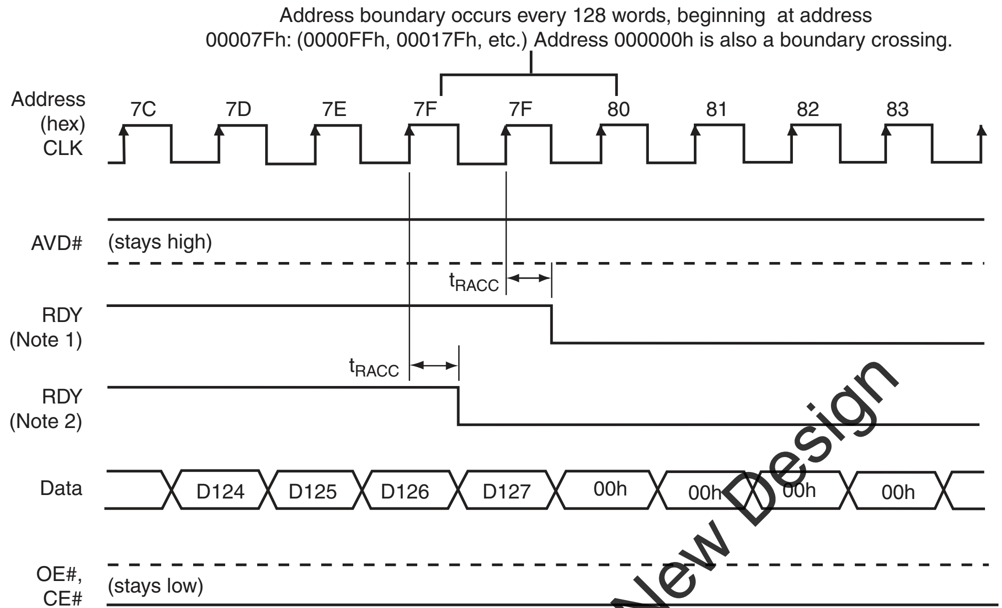
Configuration Register	Programmable Wait States			
CR.14 CR.13 CR.12 CR.11	0000 =	initial data is valid on the		rising CLK edge after addresses are latched
	0001 =		3rd	
	0010 =		4th	
	0011 =		5th	
	0100 =		6th	
	0101 =		7th	
	0110 =		8th	
	0111 =		9th	
	1000 =		10th	
	. . .		. . .	
	1011 =		13th	
	1100 =	Reserved		
	1111 =			

Figure 11.17 Latency with Boundary Crossing



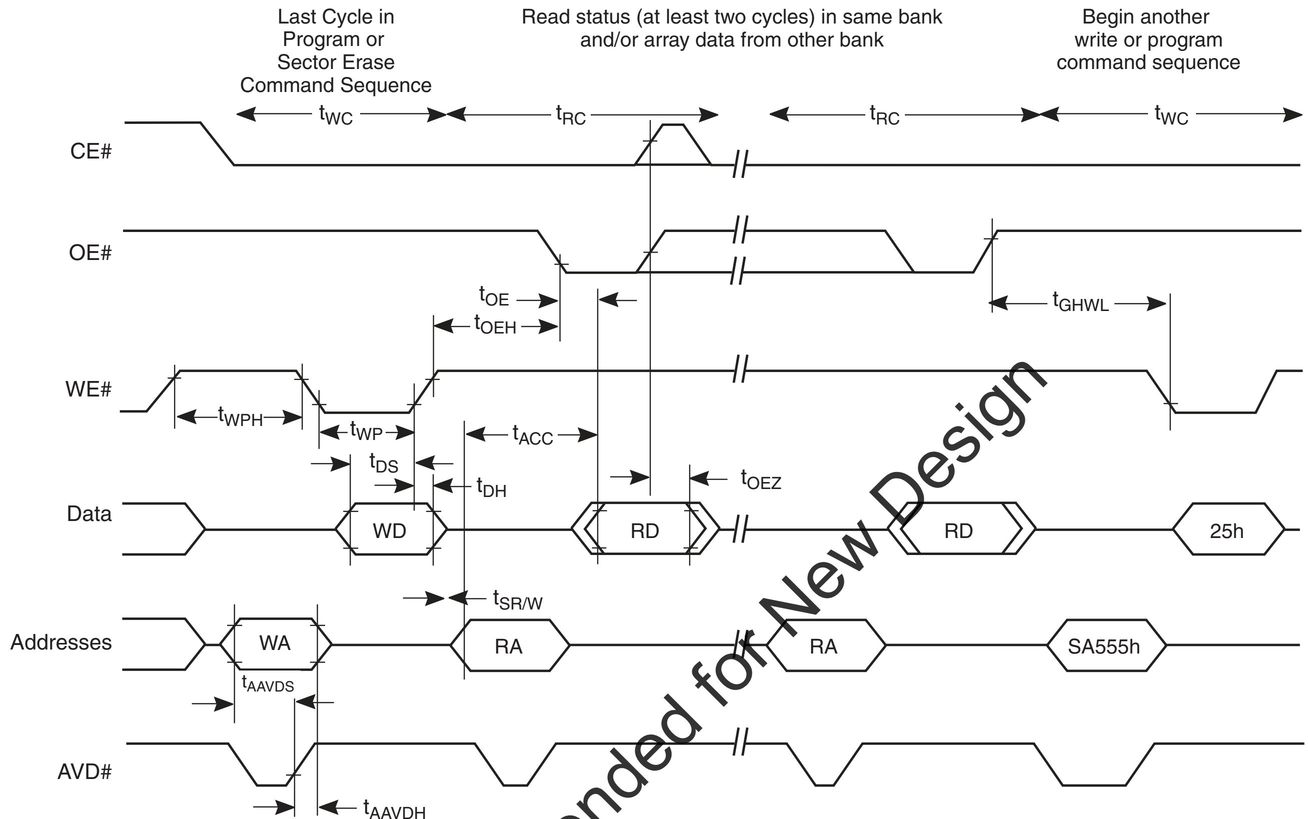
Notes
1. *RDY active with data (CR.8 = 1 in the Configuration Register).*
2. *RDY active one clock cycle before data (CR.8 = 0 in the Configuration Register).*
3. *Figure shows the device not crossing a bank in the process of performing an erase or program.*

Figure 11.18 Latency with Boundary Crossing into Bank Performing Embedded Operation



Notes
1. *RDY active with data (CR.8 = 1 in the Configuration Register).*
2. *RDY active one clock cycle before data (CR.8 = 0 in the Configuration Register).*
3. *Figure shows the device crossing a bank in the process of performing an erase or program.*

Figure 11.19 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.*

11.9.5 Erase and Programming Performance

Parameter			Typ (Note 1)	Max (Note 2)	Unit	Comments
Sector Erase Time (7)	128 Kbyte	V_{CC}	0.8/1.3 (6)	3.5	s	(Note 3)
	32 Kbyte	V_{CC}	0.35/0.6 (6)	2		
	128 Kbyte	ACC	0.8 / 1.3 (6)	3.5		
	32 Kbyte	ACC	0.35 / 0.6 (6)	2		
Chip Erase Time (7)		V_{CC} (6)	78/126 (128 Mbit) 155/251 (256 Mbit) 308/500 (512 Mbit)	154/250 (128 Mbit) 308/500 (256 Mbit) 612/998 (512 Mbit)		
		ACC	78/126 (128 Mbit) 155/251 (256 Mbit) 308/500 (512 Mbit)	154/250 (128 Mbit) 308/500 (256 Mbit) 612/998 (512 Mbit)		
Single Word Program Time		V_{CC}	130	400	µs	Excludes system level overhead (Note 4)
Effective Word Programming Time using Program Write Buffer		V_{CC}	12.5	94		
		ACC	8	60		
Total 32-Word Buffer Programming Time		V_{CC}	400	3000		
		ACC	256	1920		
Chip Programming Time (using 32 word buffer)		V_{CC}	105 (128 Mbit) 210 (256 Mbit) 420 (512 Mbit)	210 (128 Mbit) 420 (256 Mbit) 840 (512 Mbit)	s	Excludes system level overhead (Note 4)
		ACC	68 (128 Mbit) 135 (256 Mbit) 269 (512 Mbit)	136 (128 Mbit) 270 (256 Mbit) 538 (512 Mbit)		
Erase Suspend/Erase Resume (t_{ESL})				30	µs	
Program Suspend/Program Resume (t_{PSL})				30	µs	
Blank Check				1	ms	

Notes

1. *Typical program and erase times assume the following conditions: 25°C, 1.8 V V_{CC}, 10,000 cycles. Additionally, programming typically assumes a checkerboard pattern.*
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 bus-cycle sequence for the program command. See Table 12.1 on page 63 for further information on command definitions.*
5. *The device has a minimum erase and program cycle endurance of 100,000 cycles.*
6. *The first value is excluding pre-programming time, while the second value is inclusive of pre-programming time for the FFFFh pattern, with status polling rate as 400 ns.*
7. *The erase time is calculated from the time of issuing erase command to the completion of erase operation (indicated by status register).*

12. Appendix

This section contains information relating to software control or interfacing with the flash device.

12.1 Command Definitions

Where appropriate, addresses are listed in both 16-bit word mode for the flash device address space and byte mode for the system address space. The byte address is listed above the word address.

Table 12.1 Command Definitions (Sheet 1 of 2)

Command Sequence	Cycles	Bus Cycles (Notes 1–5)							
		First		Second		Third		Fourth	
		Addr Byte Word	Data	Addr Byte Word	Data	Addr Byte Word	Data	Addr Byte Word	Data
Read	1	RA	RD						
Reset	1	XX	F0						
Write Buffer Load (9)	4-35	(SA) AAA (SA) 555	25	(SA) 555 (SA) AAA	WC	(SA) PA (12)	PD	(SA) PA (13)	PD
Buffer to Flash	1	(SA) AAA (SA) 555	29						
Chip Erase	2	(SA) AAA (SA) 555	80	(SA) 555 (SA) AAA	10				
Sector Erase	2	(SA) AAA (SA) 555	80	(SA) 555 (SA) AAA	30				
Status Register Read	2	(SA) AAA (SA) 555	70	(SA)	RR				
Status Register Clear	1	(SA) AAA (SA) 555	71						
Program Suspend (6)	1	XXX	51						
Program Resume (6)	1	XXX	50						
Erase Suspend (7)	1	XXX	B0						
Erase Resume (7)	1	XXX	30						
Blank Check (14)	1	(SA) AAA (SA) 555	33						
Sector Lock/Unlock	3	AAA 555	60	555 AAA	60	SLA	60		
Sector Lock Range	4	AAA 555	60	555 AAA	60	SLA	61	SLA	61
ID/CFI Command Definitions									
ID/CFI: ID/CFI Entry (8) (11)	1	(SA) XAA (SA) X55	90 or 98						
ID/CFI: ID/CFI Read	1	(SA) RA	data						
ID/CFI: ID/CFI Exit	1	XXX	F0						
Configuration Command Definitions									

Table 12.1 Command Definitions (Sheet 2 of 2)

Command Sequence		Cycles	Bus Cycles (Notes 1–5) First Addr Byte Word	First Data	Second Addr Byte Word	Second Data	Third Addr Byte Word	Third Data	Fourth Addr Byte Word	Fourth Data
Configuration Register	Configuration Register Entry (8) (11)	1	(SA) AAA (SA) 555	D0						
	Write Buffer Load	3	(SA) AAA (SA) 555	25	(SA) 555 (SA) AAA	0	(SA) X00(SA) X00	PD		
	Buffer to Flash (Configuration Register)	1	(SA) AAA (SA) 555	29						
	Configuration Register Read	1	(SA) X00(SA) X00	RR						
	Configuration Register Exit	1	XXX	F0						
SSR Lock Command Definitions										
SSR Lock	SSR Lock Entry (8) (11)	1	(SA) AAA (SA) 555	40						
	Write Buffer Load (9)	3	(SA) AAA (SA) 555	25	(SA) 555 (SA) AAA	0	(SA) 00	FE		
	Buffer to Flash	1	(SA) AAA (SA) 555	29						
	SSR Lock Read	1	(SA) XXX	RR						
	SSR Lock Exit	1	XXX	F0						
Secure Silicon Region Command Definitions										
Secure Silicon Region	Secure Silicon Region Entry (8) (11)	1	(SA) AAA (SA) 555	88						
	Write Buffer Load (9)	4-35	(SA) AAA (SA) 555	25	(SA) 555 (SA) AAA	WC	(SA) PA (12)	PD	(SA) PA (13)	PD
	Buffer to Flash	1	(SA) AAA (SA) 555	29						
	Secure Silicon Region Read	1	(SA) RA	RD						
	Secure Silicon Region Exit	1	XXX	F0						

Legend

X = Don't care

RA = Address of the location to be read.

RD = Read Data from location RA during read operation.

RR = Read Register value

PA = Address of the memory location to be programmed.

PD = Data to be programmed at location PA.

BA = Address bits sufficient to select a bank

SA = Address bits sufficient to select a sector

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 Section 8., Device Operations on page 23 for description of bus operations.*
2. *All values are in hexadecimal. All addresses are shown both in terms of system viewpoint byte address above the flash viewpoint word address where system address a-max to a1 is connected to flash A-max to A0. Low order address bit patterns in commands, that are below the bits that specify (BA) or (SA), are relevant only on system address signals a11 to a1 i.e. flash address signals A10 to A0. In commands using address bit patterns as part of the command recognition the address bits above system a11 (flash A10) and below (BA) or (SA) are don't care. System a0 is also don't care because it is not connected to the flash address inputs.*
3. *Except for the following, all bus cycles are write cycle: read cycle during Read, ID/CFI Read (Manufacturing ID, Device ID, Indicator Bits), Configuration Register read, Secure Silicon Region Read, SSR Lock Read, and 2nd cycle of Status Register Read*
4. *Data bits DQ15–DQ8 are don't care in command sequences, except for RD, PD, and WD.*
5. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data*

6. The Program Resume command is valid only during the Program Suspend mode/state.
7. The Erase Resume command is valid only during the Erase Suspend mode/state.
8. Command is valid when all banks are ready to read array data.
9. The total number of cycles in the command sequence is determined by the number of words written to the write buffer.
10. ACC must be at V_{HH} during the entire operation of this command
11. Entry commands are needed to enter a specific mode to enable instructions only available within that mode.
12. Must be the lowest word address of the words being programmed within the 32 word write buffer page. This is not necessarily the lowest address of the page. Data words are loaded into the write page buffer in sequential order from lowest to highest address.
13. Subsequent addresses must fall within the same Sector and Page as the initial starting address.
14. Blank Check is only functional in Asynchronous Read mode (Configuration Register - CR[15] = 1).

12.2 Device ID and Common Flash Memory Interface Address Map

The Device ID fields occupy the first 32 bytes of address space followed by the Common Flash Interface data structure. The Common Flash Interface (CFI) specification defines a standardized data structure containing device specific parameter, structure, and feature set information, which allows 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 driver software can be standardized for long-term compatibility.

This device enters the ID/CFI mode when the system writes the ID/CFI Query command, 90h or 98h, to address (SA)55h any time all banks are in read mode (the CU is in Idle State). The system can then read ID and CFI information at the addresses, within the selected sector, given in the following tables. To terminate reading ID/CFI, the system must write the reset command.

Table 12.2 ID/CFI Data (Sheet 1 of 5)

	Word Offset Address	Byte Offset Address	DATA WS512R	WS256R	WS128R	Description
Device Identification	(SA) + 00h	(SA) + 00h	0001h			Spansion Manufacturer ID
	(SA) + 01h	(SA) + 02h	(WS) 007Eh	(WS) 007Eh	(WS) 007Eh	Device ID, Word 1 Extended ID address code. Indicates an extended two byte device ID is located at byte address 1Ch and 1Eh.
	(SA) + 02h	(SA) + 04h	00FFh			Reserved
	(SA) + 03h	(SA) + 06h	00FFh			Revision ID
	(SA) + 04h	(SA) + 08h	00FFh			Reserved
	(SA) + 05h	(SA) + 0Ah	00FFh			Reserved
	(SA) + 06h	(SA) + 0Ch	00FFh			ID Version
	(SA) + 07h	(SA) + 0Eh	00BFh			Reserved
	(SA) + 08h	(SA) + 10h	00FFh			Reserved
	(SA) + 09h	(SA) + 12h	00FFh			Reserved
	(SA) + 0Ah	(SA) + 14h	00FFh			Reserved
	(SA) + 0Bh	(SA) + 16h	00FFh			Reserved
	(SA) + 0Ch	(SA) + 18h	00FF Bit 0 - Status Register Support 1 = Status Register Supported 0 = Status register not Supported Bit 1 - DQ Polling Support 1 = DQ bits polling supported 0 = DQ bits polling not supported Bit 3-2 - Command Set Support 11 = Reserved 10 = Reserved 01 = Reduced Command Set 00 = Old Command Set Bit 4-F - 00Fh - Reserved			Lower Software Bits
	(SA) + 0Dh	(SA) + 1Ah	00FFh			Upper Software Bits Reserved
	(SA) + 0Eh	(SA) + 1Ch	0025h	0026h	0027h	High Order Device ID, Word 2
	(SA) + 0Fh	(SA) + 1Eh	0003h	0003h	0003h	Low Order Device ID, Word 3
CFI	**CFI Query Identification String**					
	(SA) + 10h	(SA) + 20h	0051h			Query Unique ASCII string "QRY"
	(SA) + 11h	(SA) + 22h	0052h			
	(SA) + 12h	(SA) + 24h	0059h			
	(SA) + 13h	(SA) + 26h	0002h			Primary Algorithm Command Set (Spansion = 0002h)
	(SA) + 14h	(SA) + 28h	0000h			
	(SA) + 15h	(SA) + 2Ah	0040h			Address for Primary Extended Table
	(SA) + 16h	(SA) + 2Ch	0000h			
	(SA) + 17h	(SA) + 2Eh	0000h			Alternate Algorithm Command Set (00h = none exists)
	(SA) + 18h	(SA) + 30h	0000h			
	(SA) + 19h	(SA) + 32h	0000h			Address for Secondary Algorithm extended Query Table (00h = none exists)
	(SA) + 1Ah	(SA) + 34h	0000h			

Table 12.2 ID/CFI Data (Sheet 2 of 5)

	Word Offset Address	Byte Offset Address	DATA			Description
			WS512R	WS256R	WS128R	
Common Flash Interface	System Interface String					
	(SA) + 1Bh	(SA) + 36h	0017h			V_{CC} Logic Supply Minimum Program/Erase or Write voltage D7-D4: Volt D3-D0: 100 millivolt
	(SA) + 1Ch	(SA) + 38h	0019h			V_{CC} Logic Supply Maximum Program/Erase or Write voltage D7-D4: Volt D3-D0: 100 millivolt
	(SA) + 1Dh	(SA) + 3Ah	0085h			V_{PP} [Programming] Supply Minimum Program/Erase voltage (00h = no V_{PP} pin present)
	(SA) + 1Eh	(SA) + 3Ch	0095h			V_{PP} [Programming] Supply Maximum Program/Erase voltage (00h = no V_{PP} pin present)
	(SA) + 1Fh	(SA) + 3Eh	0008h			Typical Word Programming Time per single word 2^N µs (e.g. < or = 32 µs)
	(SA) + 20h	(SA) + 40h	0009h			Typical Program Time for programming the complete buffer 2^N µs (e.g. < or = 256 µs) (00h = not supported)
	(SA) + 21h	(SA) + 42h	000Ah			Typical Time for Sector Erase 2^N µs
	(SA) + 22h	(SA) + 44h	0013h	0012h	0011h	Typical Time for full chip erase 2^N µs (00h = not supported)
	(SA) + 23h	(SA) + 46h	0003h			Max. Program Time per single word [2^N times typical value]
	(SA) + 24h	(SA) + 48h	0003h			Max. Program Time using buffer [2^N times typical value]
	(SA) + 25h	(SA) + 4Ah	0003h			Max. Time for sector erase [2^N times typical value]
	(SA) + 26h	(SA) + 4Ch	0003h			Max. Time for full chip erase [2^N times typical value] (00h = not supported)

Table 12.2 ID/CFI Data (Sheet 3 of 5)

	Word Offset Address	Byte Offset Address	DATA WS512R	DATA WS256R	DATA WS128R	Description
Common Flash Interface	Device Geometry Definition					
	(SA) + 27h	(SA) + 4Eh	001Ah	0019h	0018h	Device Size = 2^N byte
	(SA) + 28h	(SA) + 50h	0001h			Flash Device Interface 0h = x8 1h = x16 2h = x8/x16 3h = x32 [lower byte]
	(SA) + 29h	(SA) + 52h	0000h			[upper byte] (00h = not supported)
	(SA) + 2Ah	(SA) + 54h	0006h			Max. number of bytes in multi-byte buffer write = 2^N [lower byte]
	(SA) + 2Bh	(SA) + 56h	0000h			[upper byte] (00h = not supported)
	(SA) + 2Ch	(SA) + 58h	0001h Uniform 0002h Top or Bottom			Number of Erase Block Regions within device (Number of regions within the device containing one or more contiguous Erase Blocks of the same size)
	(SA) + 2Dh	(SA) + 5Ah	00FEh (Top Boot)		007Eh (Top Boot)	Erase Block Region 1 information [lower byte] - Number of Erase sectors minus one, of identical size within the Erase Block Region. Example: 00h = 1 sector; 01h = 2 sectors
			0003h (Bottom Boot)			
			00FFh (Uniform)		007Fh (Uniform)	
	(SA) + 2Eh	(SA) + 5Ch	0001h (Top Boot or Uniform)	0000h (Top Boot or Uniform)		[upper byte]
			000h (Bottom Boot)	0000h (Bottom Boot)		
	(SA) + 2Fh	(SA) + 5Eh	0000h (Top Boot or Uniform) 0080h (Bottom Boot)			[lower byte] - Sector Size in bytes divided by 256 (n [bytes]h = sector size / 256)
	(SA) + 30h	(SA) + 60h	0000h (Bottom Boot) 0002h (Top Boot or Uniform)			[upper byte]
	(SA) + 31h	(SA) + 62h	0003h (Top Boot)	0003h (Top Boot)	0003h (Top Boot)	Erase block Region 2 Information
			00FEh (Bottom Boot)	00FEh (Bottom Boot)	007Eh (Bottom Boot)	
			00FFh (Uniform)			
	(SA) + 32h	(SA) + 64h	000h (Top Boot)	0000h (Top Boot)	0000h (Top Boot)	
			0001h (Bottom Boot)	0000h (Bottom Boot)		
			00FFh (Uniform)			
	(SA) + 33h	(SA) + 66h	0000h (Bottom Boot or Uniform) 0080h (Top Boot)			
	(SA) + 34h	(SA) + 68h	0002h (Bottom Boot) 0000h (Top Boot or Uniform)			

Table 12.2 ID/CFI Data (Sheet 4 of 5)

	Word Offset Address	Byte Offset Address	DATA WS512R	WS256R	WS128R	Description
Common Flash Interface			**Primary Algorithm-Specific Extended Query**			
	(SA) + 40h	(SA) + 80h	0050h			Query Unique ASCII string "PRI"
	(SA) + 41h	(SA) + 82h	0052h			
	(SA) + 42h	(SA) + 84h	0049h			
	(SA) + 43h	(SA) + 86h	0031h			Major CFI version number, ASCII
	(SA) + 44h	(SA) + 88h	0034h			Minor CFI version number, ASCII
	(SA) + 45h	(SA) + 8Ah	0020h			Address Sensitive Unlock (Bits 1-0): 00b = Required 01b = Not required Process Technology (Bits 5-2) 0011b = 130 nm Floating-Gate Technology 0100b = 110 nm MirrorBit Technology 0101b = 90 nm Floating-Gate Technology 0110b = 90 nm MirrorBit Technology 1000b = 65 nm MirrorBit Technology
	(SA) + 46h	(SA) + 8Ch	0002h			Erase Suspend 0= Not supported 1 = To Read Only 2 = To Read & Write
	(SA) + 47h	(SA) + 8Eh	0001h			Sector Protection per Group 0 = not Supported X = number of sectors in per group
	(SA) + 48h	(SA) + 90h	0000h			Sector Temporary Unprotect 00h = Not Supported 01h = Supported
	(SA) + 49h	(SA) + 92h	0009h			Sector Protect/Unprotect scheme 08h = Advanced Sector Protection 09h = Single-Sector Lock + Sector Lock Range
	(SA) + 4Ah	(SA) + 94h	0020h	0010h	0008h	Simultaneous Operations Number of Sectors in all banks except Boot Bank
	(SA) + 4Bh	(SA) + 96h	0001h			Burst Mode Type 00h = Not Supported 01h = Supported
	(SA) + 4Ch	(SA) + 98h	0002h (WS-R)			Page Mode Type 00h = Not Supported 01h = 4-Word Page 02h = 8-Word Page 04h = 16-Word Page
	(SA) + 4Dh	(SA) + 9Ah	0085h			ACC (Acceleration) Supply Minimum 00h = Not Supported D7-D4: Volt D3-D0: 100 millivolt
	(SA) + 4Eh	(SA) + 9Ch	0095h			ACC (Acceleration) Supply Maximum 00h = Not Supported D7-D4: Volt D3-D0: 100 millivolt
	(SA) + 4Fh	(SA) + 9Eh	0003h (Top Boot) 0002h (Bottom Boot) 0000h (Uniform or No Boot)			Top/Bottom Sector Flag 00h = No Boot 01h = Dual Boot 02h = Bottom boot 03h = Top boot 04h = Uniform Bottom Boot 05h = Uniform Top Boot;
	(SA) + 50h	(SA) + A0h	0001h			Program Suspend 00h = Not Supported 01h= Supported

Table 12.2 ID/CFI Data (Sheet 5 of 5)

	Word Offset Address	Byte Offset Address	DATA WS512R	WS256R	WS128R	Description
Common Flash Interface	(SA) + 51h	(SA) + A2h	0000h			Unlock Bypass 00h = Not Supported
	(SA) + 52h	(SA) + A4h	0008h			Secure Silicon Region (Customer SSR Area) Size 2^N bytes
	(SA) + 53h	(SA) + A6h	000Eh			Hardware Reset Low Time-out until reset is completed during an embedded algorithm - Maximum 2^N ns (e.g. 10 µs => n = E)
	(SA) + 54h	(SA) + A8h	000Eh			Hardware Reset Low Time-out until reset is completed not during an embedded algorithm - Maximum 2^N ns (e.g. 10 µs => n = E)
	(SA) + 55h	(SA) + AAh	0005h			Erase Suspend Time-out Maximum 2^N µs
	(SA) + 56h	(SA) + ACh	0005h			Program Suspend Time-out Maximum 2^N µs
	(SA) + 57h	(SA) + AEh	0010h			Bank Organization: X= Number of banks
	(SA) + 58h	(SA) + B0h	0023h (Bottom Boot) 0020h (Top Boot or Uniform)	0013h (Bottom Boot) 0010h (Top Boot or Uniform)	000Bh (Bottom Boot) 0008h (Top Boot or Uniform)	Bank 0 Region Information. X= Number of sectors in bank
	(SA) + 59h	(SA) + B2h	0020h	0010h	0008h	Bank 1 Region Information. X= Number of sectors in bank
	(SA) + 5Ah	(SA) + B4h	0020h	0010h	0008h	Bank 2 Region Information. X= Number of sectors in bank
	(SA) + 5Bh	(SA) + B6h	0020h	0010h	0008h	Bank 3 Region Information. X= Number of sectors in bank
	(SA) + 5Ch	(SA) + B8h	0020h	0010h	0008h	Bank 4 Region Information. X= Number of sectors in bank
	(SA) + 5Dh	(SA) + BAh	0020h	0010h	0008h	Bank 5 Region Information. X= Number of sectors in bank
	(SA) + 5Eh	(SA) + BCh	0020h	0010h	0008h	Bank 6 Region Information. X= Number of sectors in bank
	(SA) + 5Fh	(SA) + BEh	0020h	0010h	0008h	Bank 7 Region Information. X= Number of sectors in bank
	(SA) + 60h	(SA) + C0h	0020h	0010h	0008h	Bank 8 Region Information. X= Number of sectors in bank
	(SA) + 61h	(SA) + C2h	0020h	0010h	0008h	Bank 9 Region Information. X= Number of sectors in bank
	(SA) + 62h	(SA) + C4h	0020h	0010h	0008h	Bank 10 Region Information. X= Number of sectors in bank
	(SA) + 63h	(SA) + C6h	0020h	0010h	0008h	Bank 11 Region Information. X= Number of sectors in bank
	(SA) + 64h	(SA) + C8h	0020h	0010h	0008h	Bank 12 Region Information. X= Number of sectors in bank
	(SA) + 65h	(SA) + CAh	0020h	0010h	0008h	Bank 13 Region Information. X= Number of sectors in bank
	(SA) + 66h	(SA) + CCh	0020h	0010h	0008h	Bank 14 Region Information. X= Number of sectors in bank
	(SA) + 67h	(SA) + CEh	0023h (Top Boot) 0020h (Bottom Boot or Uniform)	0013h (Top Boot) 0010h (Bottom Boot or Uniform)	000Bh (Top Boot) 0008h (Bottom Boot or Uniform)	Bank 15 Region Information. X= Number of sectors in bank

13. Revision History

Section	Description
Revision 01 (March 28, 2007)	
	Initial release
Revision 02 (April 24, 2007)	
Device ID and Common Flash Memory Interface Address Map	Updated device ID for word offset 03H to 00FFh
Revision 03 (March 19, 2008)	
Effective Write Buffer Programming (V_{CC}) Per Byte	Changed Typical value from 4.7µs to 9.6 µs
Effective Write Buffer Programming (V_{ACC}) Per Byte	Changed Typical value from 2.5 µs to 6 µs
Typical Program & Erase Time	Change Sector Erase (32 KByte Sector) from 0.3s to 0.355s
Product Ordering Information	Added package, boot configuration and frequency information to ordering information
Valid Combination Table	Removed WS2901GR Added model # 00 and 02 for boot configuration
Block Diagram	Renamed V_{PP} to ACC
Input/Output Descriptions & Logic Symbol	Removed A/D mux from I/O description Removed Address multiplexing wording from AVD#
Pin Out Diagram	Removed WP# pin
Synchronous Read Wait State Table	Reformatted the frequency description
DC Characterization	Added 83 MHz value for I_{CCB} and I_{CC5} Changed all 108 MHz to 104 MHz Changed V_{LKO} Min/Max from 1.3V/1.4V to 1.0V/1.1V
AC Characterization	Synchronous Burst: Changed t_{BDH} min from 5 ns for 66 MHz/83 MHz to 3 ns, 2.5 ns for 108 MHz/ 133 Mhz to 2 ns for 104 Mhz/133 MHz Asynchronous Read: Changed t_{WEA} from 4 ns to t_{CLK} Asyncronous Write: Change t_{ESL} and t_{PSL} from 20 µs to 30 µs Added note 3 on AVD requirement to subsequent CLK cycle
Product Overview	Corrected Secure Silicon Region size to 512 Bytes
Synchronous Burst Read Mode	Deleted wait state 2 from Wait State Tables Figure 7.1 Changed 256 Byte Boundary Crossing Latency additional wait states up to 2
V_{CC} Power Up	Changed t_{VCS} value to 300 µs Changed V_{IOS} value to 300 µs Added t_{RPH} to timing diagram
Wait State Configuration Register Set-up	Corrected typo in table to wait state 13
Device ID and CFI Table	General update CFI values
Revision 04 (March 25, 2008)	
Device ID and Common Flash Memory Interface Table	Corrected CFI setting for 4Ah
Revision 05 (October 27, 2008)	
Global	Removed 133 MHz speed option
Ordering Information & Valid Combination Table	Added Low-Halogen Lead Free package option Removed Standard Lead Free package option
Flash Memory Array	Updated table 6.6 to correct sector range for bank 16 of S29WS512R
Table 7.1 Device Bus Operation	Deleted Asynchronous Write (WE# latched Address)

Section	Description
Write Buffer Programming Command	Changed to word offset
AC Characterization	Changed Clock High or Low time (t_{CLK} H/L) to .45 t_{CLK}
Program/Erase Operation	Deleted 7.9.2 Programming of a previously programmed location
Program and Erase Performance	Deleted ACC mode from single word programming Changed 32-word buffer programming performance V_{CC} mode 400 µs, ACC mode to 256 µs Change Chip Programming Time for V_{CC} mode as below: • 128 Mbit 105s typical 210s max • 256 Mbit 210s typical 420s max • 512 Mbit 420s typical 840s max Changed sector erase time as follows: • 128 Kbyte V_{CC}: 0.8/1.3 (Note 6) (typ), 3.5/5.5 (max) • 32 Kbyte V_{CC}: 0.35/0.6 (Note 6) (typ), 2.0/3.5 (max) • 128 Kbyte ACC: 0.8/1.3 (Note 6) (typ), 3.5/5.5 (max) • 32 Kbyte ACC: 0.35/0.6 (Note 6) (typ), 2.0/3.5 (max) Updated chip erase time (both VCC and VPP) as follows: • WS128R: 78/126 s (typ), 154/250 s (max) • WS256R: 155/251 s (typ), 308/500 s (max) • WS512R: 308/500 s (typ), 612/998 s (max) Added Note 6 to state, The first value is excluding pre-programming time, while the second value is inclusive of pre-programming for the FFFFh pattern, with status polling rate as 400 ns (typ). Added Note 7 to state, The erase time is calculated from the time of issuing erase command to the completion of erase operation (indicated by status register).
Revision 06 (February 27, 2009)	
Global	Remove 1G product option
Ordering Information and Valid Combinations	Add additional model numbers for uniform boot, 1 and 2CEs
Device Operation	Blank Check Command functional in Asynchronous Read Mode only
DC Characteristics	Change I_{CCB} for 83 MHz 16word burst to 26 mA
Programming Performance	Effect Word Programming time using Write Buffer change to 12.5 µs for V_{CC} mode and 8 µs for ACC mode
Revision 07 (May 7, 2009)	
Ordering Information and Valid Combinations	Removed model numbers 10, 30, 50
Input/Output Descriptions & Logic Symbol	Input/Output Descriptions table – Updated ACC description
V_{CC} Power Up	Removed t_{RH} from table
Device ID and Common Flash Memory Interface Address Map	ID/CFI Data table – Updated (SA) + 51h description
Revision 08 (June 1, 2010)	
Sector Lock Range Command	Clarified Sector Lock Range command behavior
Revision 09 (September 19, 2011)	
Global	Added product obsolescence information

Document History Page

Document Title: S29WS512R, S29WS256R, S29WS128R 512/256/128 Mb (32/16/8M x 16 bit), 1.8 V, S29WS-R MirrorBit® Flash Document Number: 002-01101				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	–	WIOB	03/28/2007	Initial release
*A	–	WIOB	04/24/2007	Device ID and Common Flash Memory Interface Address Map: Updated device ID for word offset 03H to 00FFh
*B	–	WIOB	03/19/2008	Effective Write Buffer Programming (VCC) Per Byte: Changed Typical value from 4.7µs to 9.6 µs Effective Write Buffer Programming (VACC) Per Byte: Changed Typical value from 2.5 µs to 6 µs Typical Program & Erase Time: Change Sector Erase (32 KByte Sector) from 0.3s to 0.355s Product Ordering Information: Added package, boot configuration and frequency information to ordering information Valid Combination Table: Removed WS2901GR Added model # 00 and 02 for boot configuration Block Diagram: Renamed VPP to ACC Input/Output Descriptions & Logic Symbol: Removed A/D mux from I/O description Removed Address multiplexing wording from AVD# Pin Out Diagram: Removed WP# pin Synchronous Read Wait State Table: Reformatted the frequency description DC Characterization: Added 83 MHz value for ICCB and ICC5 Changed all 108 MHz to 104 MHz Changed VLKO Min/Max from 1.3V/1.4V to 1.0V/1.1V AC Characterization: Synchronous Burst: Changed tBDH min from 5 ns for 66 MHz/83 MHz to 3 ns, 2.5 ns for 108 MHz/133 Mhz to 2 ns for 104 Mhz/133 MHz Asynchronous Read: Changed tWEA from 4 ns to tCLK Asyncronous Write: Change tESL and tPSL from 20 µs to 30 µs Added note 3 on AVD requirement to subsequent CLK cycle Product Overview: Corrected Secure Silicon Region size to 512 Bytes Synchronous Burst Read Mode: Deleted wait state 2 from Wait State Tables Figure 7.1Changed 256 Byte Boundary Crossing Latency additional wait states up to 2 VCC Power Up: Changed tVCS value to 300 µs Changed VIOS value to 300 µs Added tRPH to timing diagram Wait State Configuration Register Set-up: Corrected typo in table to wait state 13
*C	–	WIOB	03/25/2008	Device ID and Common Flash Memory Interface Table: Corrected CFI setting for 4Ah

Document History Page (Continued)

Document Title: S29WS512R, S29WS256R, S29WS128R 512/256/128 Mb (32/16/8M x 16 bit), 1.8 V, S29WS-R MirrorBit® Flash Document Number: 002-01101				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
*D	—	WIOB	10/27/2008	Global: Removed 133 MHz speed option Ordering Information & Valid Combination Table: Added Low-Halogen Lead Free package option Removed Standard Lead Free package option Flash Memory Array: Updated table 6.6 to correct sector range for bank 16 of S29WS512R Table 7.1 Device Bus Operation: Deleted Asynchronous Write (WE# latched Address) Write Buffer Programming Command: Changed to word offset AC Characterization: Changed Clock High or Low time (t_{CLK} H/L) to .45 t_{CLK} Program/Erase Operation: Deleted 7.9.2 Programming of a previously programmed location Program and Erase Performance: Deleted ACC mode from single word programming Changed 32-word buffer programming performance V_{CC} mode 400 µs, ACC mode to 256 µs Change Chip Programming Time for V_{CC} mode as below: • 128 Mbit 105s typical 210s max • 256 Mbit 210s typical 420s max • 512 Mbit 420s typical 840s max Changed sector erase time as follows: • 128 Kbyte V_{CC}: 0.8/1.3 (Note 6) (typ), 3.5/5.5 (max) • 32 Kbyte V_{CC}: 0.35/0.6 (Note 6) (typ), 2.0/3.5 (max) • 128 Kbyte ACC: 0.8/1.3 (Note 6) (typ), 3.5/5.5 (max) • 32 Kbyte ACC: 0.35/0.6 (Note 6) (typ), 2.0/3.5 (max) Updated chip erase time (both VCC and VPP) as follows: • WS128R: 78/126 s (typ), 154/250 s (max) • WS256R: 155/251 s (typ), 308/500 s (max) • WS512R: 308/500 s (typ), 612/998 s (max) Added Note 6 to state, The first value is excluding pre-programming time, while the second value is inclusive of pre-programming for the FFFFh pattern, with status polling rate as 400 ns (typ). Added Note 7 to state, The erase time is calculated from the time of issuing erase command to the completion of erase operation (indicated by status register).
*E	—	WIOB	02/27/2009	Global: Remove 1G product option Ordering Information and Valid Combinations: Add additional model numbers for uniform boot, 1 and 2CEs Device Operation: Blank Check Command functional in Asynchronous Read Mode only DC Characteristics: Change I_{CCB} for 83 MHz 16word burst to 26 mA Programming Performance: Effect Word Programming time using Write Buffer change to 12.5 µs for V_{CC} mode and 8 µs for ACC mode
*F	—	WIOB	05/07/2009	Ordering Information and Valid Combinations: Removed model numbers 10, 30, 50 Input/Output Descriptions & Logic Symbol: Input/Output Descriptions table – Updated ACC description V_{CC} Power Up: Removed t_{RH} from table Device ID and Common Flash Memory Interface Address Map: ID/CFI Data table – Updated (SA) + 51h description

Document History Page (Continued)

Document Title: S29WS512R, S29WS256R, S29WS128R 512/256/128 Mb (32/16/8M x 16 bit), 1.8 V, S29WS-R MirrorBit® Flash Document Number: 002-01101				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
*G	—	WIOB	06/01/2011	Sector Lock Range Command: Clarified Sector Lock Range command behavior
*H	—	WIOB	18/19/2011	Global: Added product obsolescence information
*I	4953774	WIOB	10/09/2015	Updated to Cypress template.

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

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

Products

Automotive .... cypress.com/go/automotive
Clocks & Buffers .... cypress.com/go/clocks
Interface .... cypress.com/go/interface
Lighting & Power Control .... cypress.com/go/powerpsoc
Memory .... cypress.com/go/memory
PSoC .... cypress.com/go/psoc
Touch Sensing .... cypress.com/go/touch
USB Controllers .... cypress.com/go/USB
Wireless/RF .... cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support