Please note that Cypress is an Infineon Technologies Company.

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

Continuity of document content

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

Continuity of ordering part numbers

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

Downloaded from Arrow.com.

S29VS256R
S29VS128R
S29XS256R
S29XS128R

256/128-Mbit (32/16 Mbyte), 1.8 V, 16-bit Data Bus, Multiplexed MirrorBit® Flash

Features

- Single 1.8 V supply for read/program/erase (1.70–1.95 V)
- 65nm MirrorBit® Technology
- Address and Data Interface Options
  - Address and Data Multiplexed for reduced I/O count (ADM) S29VS-R
  - Address-High, Address-Low, Data Multiplexed for minimum I/O count (AADM) S29XS-R
- Simultaneous Read/Write operation
- 32-word Write Buffer
- Bank architecture
  - Eight-bank
- Four 32-KB sectors at the top or bottom of memory array 255/127 of 128-KB sectors
- Programmable linear (8/16-word) with wrap around and continuous burst read modes
- Secured Silicon Sector region consisting of 128 words each for factory and customer
- 10-year data retention (typical)
- Cycling Endurance: 100,000 cycles per sector (typical)
- RDY output indicates data available to system
- Command set compatible with JEDEC (42.4) standard
- Hardware sector protection via V_{PP} pin
- Handshaking by monitoring RDY
- Offered Packages
  - 44-ball FBGA (6.2 mm × 7.7 mm × 1.0 mm)
- Low V_{CC} write inhibit
- Write operation status bits indicate program and erase operation completion
- Suspend and Resume commands for Program and Erase operations
- Asynchronous program operation, independent of burst control register settings
- V_{PP} input pin to reduce factory programming time
- Support for Common Flash Interface (CFI)

General Description

The Cypress S29VS256/128R and S29XS256/128R are MirrorBit® Flash products fabricated on 65nm process technology. These burst mode Flash devices are capable of performing simultaneous read and write operations with zero latency on two separate banks using multiplexed data and address pins. These products can operate up to 108 MHz and use a single V_{CC} of 1.7 V to 1.95 V that makes them ideal for the demanding wireless applications of today that require higher density, better performance, and lowered power consumption. The S29VS256/128R operates in ADM mode, while the S29XS256/128R can operate in the AADM mode.

Performance Characteristics

Read Access Times	
Speed Option (MHz)	108
Max. Synch. Latency, ns (t_{IA})	72.34
Max. Synch. Burst Access, ns (t_{BACC})	6.75
Max. Asynch. Access Time, ns (t_{ACC})	80
Max OE# Access Time, ns (t_{OE})	15

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

Typical Program & Erase Times	
Single Word Programming	170 µs
Effective Write Buffer Programming (V_{CC}) Per Word	14.1 µs
Effective Write Buffer Programming (V_{PP}) Per Word	9 µs
Sector Erase (16 Kword Sector)	350 ms
Sector Erase (64 Kword Sector)	800 ms

Downloaded from Arrow.com.

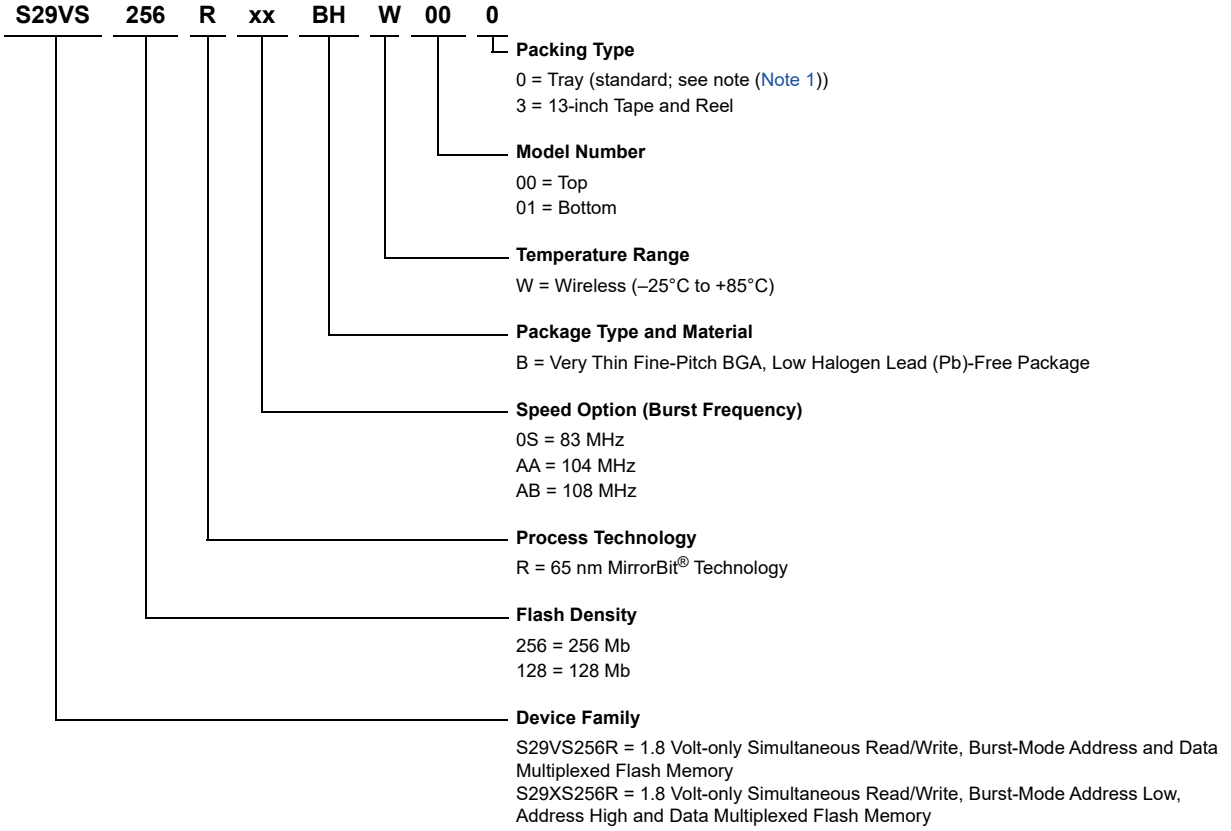
Contents

Ordering Information **3**
Valid Combinations 3
Input/Output Descriptions and Logic Symbol **4**
Block Diagram **5**
Physical Dimensions/Connection Diagrams **6**
Related Documents 6
Special Handling Instructions for FBGA Package 6
Product Overview **8**
Address Space Maps **9**
Data Address and Quantity Nomenclature 10
Flash Memory Array 11
Address/Data Interface 13
Bus Operations 14
Device ID and CFI (ID-CFI) 15
Device Operations **17**
Asynchronous Read 17
Synchronous (Burst) Read Mode and Configuration Register 18
Status Register 25
Blank Check 29
Simultaneous Read/Write 29
Writing Commands/Command Sequences 30
Program/Erase Operations 30
Handshaking 37
Hardware Reset 37
Software Reset 37
Sector Protection/Unprotection **39**
Sector Lock/Unlock Command 39
Sector Lock Range Command 39
Hardware Data Protection Methods 40
SSR Lock 40
Secure Silicon Region 40
Power Conservation Modes **42**
Standby Mode 42
Automatic Sleep Mode 42
Output Disable (OE#) 42
Electrical Specifications **43**
Absolute Maximum Ratings 43
Operating Ranges 44
DC Characteristics 44
Capacitance 45
AC Test Conditions 46
Key to Switching Waveforms 46
V_{CC} Power-Up and Power Down 47
CLK Characterization 48
AC Characteristics 49
Appendix **57**
Command Definitions 57
Device ID and Common Flash Memory Interface Address Map 59
Revision History **72**
Document History Page **72**
Sales, Solutions, and Legal Information **75**
Worldwide Sales and Design Support 75
Products 75
PSoC® Solutions 75
Cypress Developer Community 75
Technical Support 75

Downloaded from Arrow.com.

1. Ordering Information

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



1.1 Valid Combinations

Valid Combination list configurations are 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.

S29VS-R Valid Combinations (1) (2)				Model Numbers	Package Type (2)
Base Ordering Part Number	Speed Option	Package Type, Material, and Temperature Range	Packing Type		
S29VS256R	0S, AA, AB	BHW (3)	0, 3 (1)	00, 01	6.2 mm x 7.7 mm, 44-ball
S29VS128R					
S29XS256R					
S29XS128R					

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. *Industrial Temperature Range is also available. For device specification differences, please refer to the Specification Supplement with Publication Number S29VS_XS-R_SP.*

Downloaded from Arrow.com.

2. Input/Output Descriptions and Logic Symbol

Table 1 identifies the input and output package connections provided on the device.

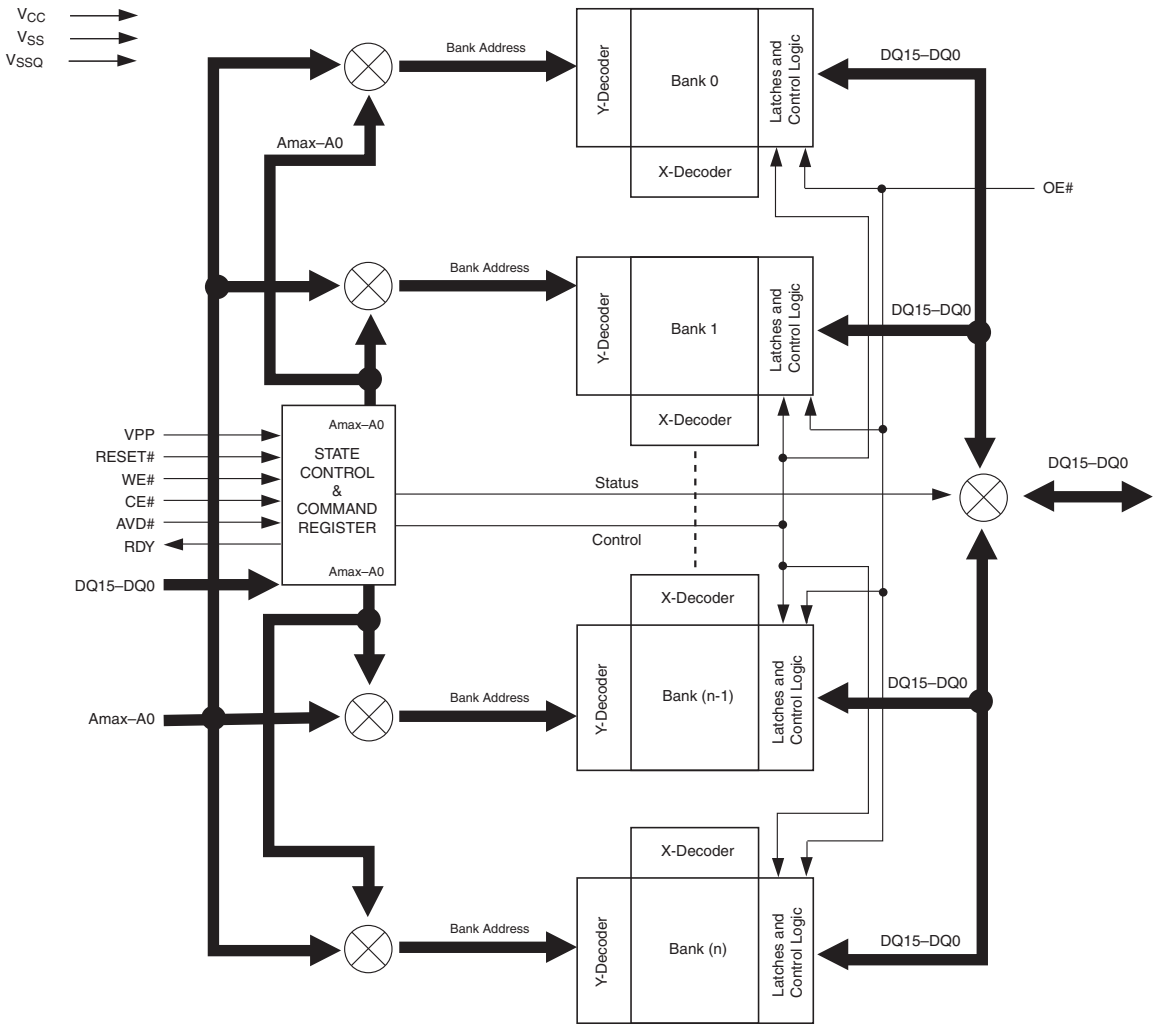
Table 1. Input/Output Descriptions

Symbol	Type	Description
Amax – A16		Higher order address lines. Amax = A23 for VS256R, A22 for VS128R. On the XS256R and XS128R, these inputs can be left unconnected in AADM mode.
A/DQ15 – A/DQ0	I/O	Multiplexed Address/Data input/output
CE#	Input	Flash Chip Enable. Asynchronous relative to CLK.
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}	I/O	Ground
V_{SSQ}	I/O	Input/Output 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 (address bits A15 – A0 are multiplexed, address bits Amax – A16 are address only). 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.
V_{PP}	Input	Accelerated input. At V_{HH}, accelerates programming; automatically places device in unlock bypass mode. At V_{IL}, disables all program and erase functions. Should be at V_{IH} for all other conditions.
RFU	Reserved	Reserved for future use

Downloaded from Arrow.com.

3. Block Diagram

Figure 1. Simultaneous Operation Circuit



Notes:

1. *Amax = A23 for S29VS/XS256R, A22 for S29VS/XS128R.*
2. *Bank(n) = 8 (S29VS/XS256/128R).*

Downloaded from Arrow.com.

4. Physical Dimensions/Connection Diagrams

This section shows the I/O designations and package specifications for the S29VS-R/S29XS-R.

4.1 Related Documents

The following document contains information relating to the S29VS-R/S29XS-R devices. Click on the title or go to www.cypress.com, or request a copy from your sales office.

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

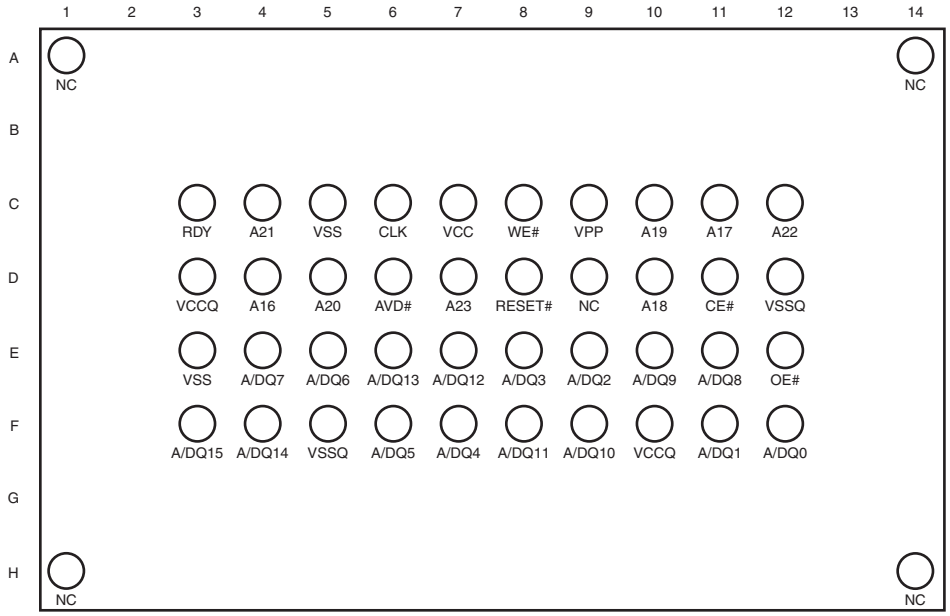
4.2 Special Handling Instructions for FBGA Package

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

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

4.2.1 44-Ball Very Thin Fine-Pitch Ball Grid Array, S29VS256R/S29XS256R/ S29VS128R/S29XS128R

Figure 2. 44-Ball Very Thin Fine-Pitch Ball Grid Array, Top View, Balls Facing Down



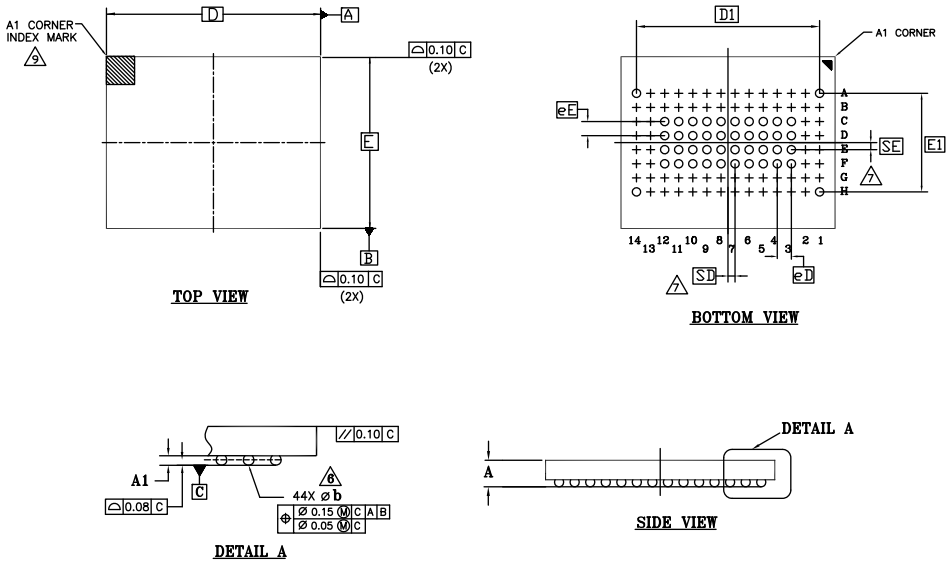
Notes:

1. *Ball D7 is NC for S29VS128R.*
2. *Balls D7, C12, C4, D5, C10, D10, C11, D4 are NC for S29XS256R and S29XS128R*

Downloaded from Arrow.com.

4.2.2 VDJ044-44-Ball Very Thin Fine-Pitch Ball Grid Array, 6.2 mm x 7.7 mm

Figure 3. VDJ044—44-Ball Very Thin Fine-Pitch Ball Grid Array



SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	-	-	1.00
A1	0.18	-	-
D	7.70 BSC		
E	6.20 BSC		
D1	6.50 BSC		
E1	3.50 BSC		
MD	14		
ME	8		
n	44		
Ø b	0.25	0.30	0.35
eD	0.50 BSC		
eE	0.50 BSC		
SD	0.25 BSC		
SE	0.25 BSC		

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING METHODS PER ASME Y14.5M-1994.
3. BALL POSITION DESIGNATION PER JEP95, SECTION 3, SPP-020.
4. [e] REPRESENTS THE SOLDER BALL GRID PITCH.
5. SYMBOL "MD" IS THE BALL MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL MATRIX SIZE IN THE "E" DIRECTION.
   n IS THE NUMBER OF POPULATED SOLDER BALL POSITIONS FOR MATRIX SIZE MD X ME.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
7. "SD" AND "SE" ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" OR "SE" = 0.
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" = eD/2 AND "SE" = eE/2.
8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
9. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.
10. JEDEC SPECIFICATION NO. REF. : N/A

002-24745 **

Downloaded from Arrow.com.

5. Product Overview

The S29VS/XS-R family is 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
S29VS128R/S29XS128R	128	16	8	8	2
S29VS256R/S29XS256R	256	32	16	8	4

The Flash memory array is divided into banks. 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. Most of the sectors are 128 KBytes each. Depending on the option ordered, either the top-4 sectors or the bottom-4 sectors are 32 KBytes each. 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.

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 S29VS/XS 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.

Downloaded from Arrow.com.

6. 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 accessed by asynchronous or burst read operations.
- 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 be directly read by asynchronous or burst accesses from the host system bus. The 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.

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.

Downloaded from Arrow.com.

6.1 Data Address and 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 boundary. A Word is a group of 16 bits aligned on a 16 bit boundary.

Throughout this document **quantities of data are generally expressed in terms of byte units**. Example: most sectors have 128 Kilo Bytes 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 a16 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. The host system byte address uses system address 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-maximum to a1 are connected to Flash A-maximum to A0 (the documentation convention here is to use lower case for system address signal numbering and upper case for Flash address signals). 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.

Downloaded from Arrow.com.

6.2 Flash Memory Array

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

Table 2. 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 3. S29VS/XS256R 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	224	128	0	SA000-SA031	000000h–1FFFFFh	000000h–3FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA032–SA063	⋮	⋮	
			2	SA064–SA095	⋮	⋮	
			3	SA096–SA127	⋮	⋮	
			4	SA128–SA159	⋮	⋮	
			5	SA160–SA191	⋮	⋮	
			6	SA192–SA223	⋮	⋮	
	31	128	7	SA224–SA254	E00000h–FEFFFFh	1C00000h–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.

Downloaded from Arrow.com.

Table 4. S29VS/XS256R 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 – Sector Ending Address
				SA001	004000h–007FFFh	008000h–00FFFFh	
				SA002	008000h–00BFFFh	010000h–017FFFh	
				SA003	00C000h–00FFFFh	018000h–01FFFFh	
	31	128		SA004–SA034	010000h–1FFFFFh	020000h–3FFFFFh	
	224	128	1	SA035–SA066	⋮	⋮	
			2	SA067–SA098	⋮	⋮	
			3	SA099–SA130	⋮	⋮	
			4	SA131–SA162	⋮	⋮	
			5	SA163–SA194	⋮	⋮	
			6	SA195–SA226	⋮	⋮	
			7	SA227–SA258	E00000h–FFFFFFh	1C00000h–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 5. S29VS/XS128R 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	112	128	0	SA000-SA015	000000h–0FFFFFh	000000h–1FFFFFh	Sector Starting Address – Sector Ending Address
			1	SA016–SA031	⋮	⋮	
			2	SA032–SA047	⋮	⋮	
			3	SA048–SA063	⋮	⋮	
			4	SA064–SA079	⋮	⋮	
			5	SA080–SA095	⋮	⋮	
			6	SA096–SA111	⋮	⋮	
	15	128	7	SA112–SA126	700000h–7EFFFFh	E00000h–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.

Downloaded from Arrow.com.

Table 6. S29VS/XS128R 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–SA018	010000h–0FFFFFh	020000h–1FFFFFh	
	112	128	1	SA019–SA034	⋮	⋮	
			2	SA035–SA050	⋮	⋮	
			3	SA051–SA066	⋮	⋮	
			4	SA067–SA082	⋮	⋮	
			5	SA083–SA098	⋮	⋮	
			6	SA099–SA114	⋮	⋮	
			7	SA115–SA130	700000h–7FFFFFh	E00000h–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.

6.3 Address/Data Interface

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

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

The two options allow use with the traditional address/data multiplexed NOR interface (S29NS family), or an address multiplexed/data multiplexed interface with the lowest signal count.

6.3.1 ADM Interface (S29VS256R and S29VS128R)

A number of processors use ADM interface as a way to reduce pin count. The system permanently connects the upper address bits (A[MAX:16] to the device. When AVD# is LOW it connects A[15:0] to DQ[15:0]. The address is latched on the rising edge of AVD#. When AVD# is HIGH, the system connects the data bus to DQ[15:0]. This results in 16-pin savings from the traditional Address and Data in Parallel (ADP) interface.

6.3.2 AADM Interface (S29XS256R and S29XS128R)

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

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

Downloaded from Arrow.com.

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

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

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

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

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

6.3.3 Default Access Mode

Upon power-up or hardware reset, the device defaults to the Asynchronous Access mode.

6.4 Bus Operations

Table 7 describes the required state of each input signal for each bus operation.

Table 7. Device Bus Operations

Operation	CE#	OE#	WE#	CLK	AVD#	A28-A16	A/DQ 15-A/DQ0	RESET#
Standby & Reset								
Standby (CE# deselect)	H	X	X	X	X	X	High-Z	H
Hardware Reset	X	X	X	X	X	X	High-Z	L
Asynchronous Mode Operations								
Asynchronous Address Latch (S29VS256R and S29VS128R)	L	H	X	X	↑	Addr In	Addr In	H
Asynchronous Upper Address Latch (S29XS256R and S29XS128R Only)	L	L	H	X	↑	X	Addr In	H
Asynchronous Lower Address Latch (S29XS256R and S29XS128R Only)	L	H	X	X	↑	X	Addr In	H
Asynchronous Read	L	L	H	X	H	X	Data Output Valid	H
Asynchronous Write Latched Data	L	H	↑	X	H	X	Data Input Valid	H

Downloaded from Arrow.com.

Table 7. Device Bus Operations (Continued)

Operation	CE#	OE#	WE#	CLK	AVD#	A28-A16	A/DQ 15-A/DQ0	RESET#
Synchronous Mode Operations								
Latch Starting Burst Address by CLK - ADM mode	L	H	H	↑	L	Addr In	Addr In	H
Latch Upper Starting Burst Address by CLK (S29XS256R and S29XS128R Only)	L	L	H	↑	L	X	Addr In	H
Latch Lower Starting Burst Address by CLK (S29XS256R and S29XS128R Only)	L	H	H	↑	L	X	Addr In	H
Burst Read and advance to next address (1)	L	L	H	↑	H	X	Data Output Valid	H
Terminate current Burst cycle	↑	X	X	X	X	X	High-Z	H

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

Note:
1. Data is delivered by a read operation only after the burst initial wait state count has been satisfied.

6.5 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 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 remove 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 8. 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 h	CFI data structure	174	Read Only

For the complete address map, see Tables in Section 11.2, Device ID and Common Flash Memory Interface Address Map on page 59.

Downloaded from Arrow.com.

6.5.1 JEDEC Device ID

The Joint Electron Device Engineering Council (JEDEC) standard JEP106T 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 code values are always byte wide but are located at bus width address boundaries such that incrementing the device address inputs will read successive byte, word, or double word locations with the codes always located in the least significant byte location of the data bus. 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.

6.5.2 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 11.2, Device ID and Common Flash Memory Interface Address Map on page 59.

Like the JEDEC Device ID information, CFI information is structured to work with any memory data bus width e.g. x8, x16, x32. The code values are always byte wide but are located at data bus width address boundaries such that incrementing the device address reads successive byte, word, or double word locations with the codes always located in the least significant byte location of the data bus. 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 Cypress CFI Version 1.4 (or later) Specification and the Cypress CFI Publication 100 (see also JEDEC publications JEP137-A and JESD68.01). Please contact JEDEC (http://www.jedec.org) for their standards and the Cypress CFI Publications may be found at the Cypress Web site (http://www.cypress.com/appnotes/CFI_v1.4_VendorSpec_Ext_A1.pdf at the time of this document's publication).

6.5.3 Secured Silicon Region

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

The Secured Silicon Region (SSR) is overlaid in the sector address specified by the SSR enter command.

Table 9. Secured Silicon Region

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

6.5.4 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 the configuration register operation description. In configuration register mode, a write of 00F0h to any address will return the sector to Read mode.

Downloaded from Arrow.com.

7. 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 43 on page 57). 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.

7.1 Asynchronous Read

The device defaults to reading array data asynchronously after device power-up or hardware reset. The device is in the Asynchronous mode when Bit 15 of the Configuration register is set to '1'. To read data from the memory array, the system must first assert CE# and AVD# to V_{IL} with WE# at V_{IH} and a valid address.

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 10.9.2, AC Characteristics–Asynchronous Read on page 50. Any input on CLK is ignored while in Asynchronous mode.

Downloaded from Arrow.com.

7.1.1 S29VS-R ADM Access

With CE# at V_{IL}, WE# at V_{IH}, and OE# at V_{IH}, the system presents the address to the device and drives AVD# to V_{IL}. AVD# is kept at V_{IL} for at least t_{AVDP} ns. The address is latched on the rising edge of AVD#.

7.1.2 S29XS-R AADM Access

With CE# at V_{IL}, WE# at V_{IH}, and OE# at V_{IL}, the system presents the upper address bits to DQ and drives AVD# to V_{IL}. The upper address bits are latched when AVD# transitions to V_{IH}. The system then drives AVD# to V_{IL} again, with OE# at V_{IH} and the lower address bits on the DQ signals. The lower address bits are latched on the next rising edge of AVD#.

7.2 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 23 for further details.

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

7.2.1 S29VS-R ADM Access

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

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

7.2.2 S29XS-R AADM Access

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

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

The device has a fixed internal address boundary that occurs every 256 Bytes (128 words). A boundary crossing latency of one or two additional wait states may be required. The device also reads data in 16 byte (8 word) aligned and length groups. When the initial address is not aligned at the beginning of a 16 byte boundary, additional wait states may be needed when crossing the first 16 byte boundary. The number of additional wait states depends on the clock frequency and starting address location.

Downloaded from Arrow.com.

Table 10 through Table 18 provide the latency for initial and boundary crossing wait state operation (note that ws = wait state).

Table 10. Initial Wait State vs. Frequency

Wait State	Frequency (Maximum MHz)
3	27
4	40
5	54
6	66
7	80
8	95
9	104
10	120

Note:
The default initial wait state delay after power on or reset is 13 wait states.

Table 11. Address Latency for 10 -13 Wait States

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

Note:
1. This column applies to the 256 Byte boundary only.

Table 12. Address Latency for 9 Wait States

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

Note:
1. This column applies to the 256 Byte boundary only.

Downloaded from Arrow.com.

Table 13. Address Latency for 8 Wait States

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

Table 14. Address Latency for 7 Wait States

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

Table 15. Address Latency for 6 Wait States

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

Downloaded from Arrow.com.

Table 16. Address Latency for 5 Wait States

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

Table 17. Address Latency for 4 Wait States

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

Table 18. Address Latency for 3 Wait States

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

Downloaded from Arrow.com.

7.2.3 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 7, Device Bus Operations on page 14.

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 might be 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).

7.2.4 8-, 16-Word Linear Burst with Wrap Around

Table 19. Burst Address Groups

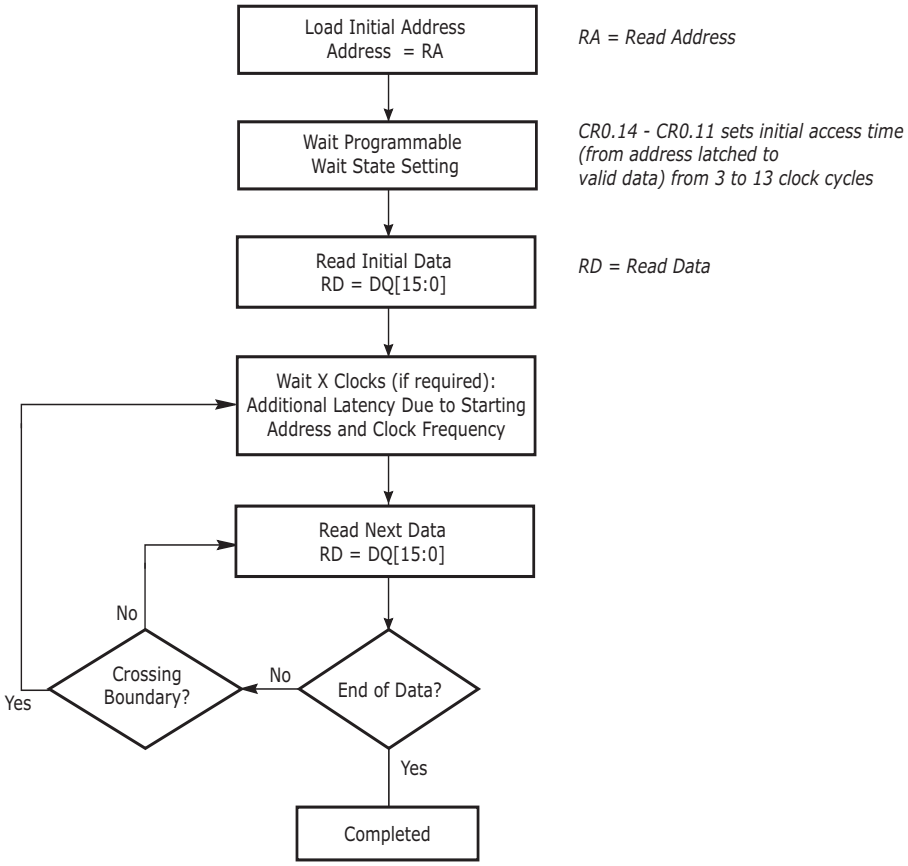
Mode	Group Size	Group Byte Address Ranges
8-word	16 bytes	0-Fh, 10-1Fh, 20-2Fh,...
16-word	32 bytes	0-1Fh, 20-3Fh, 30-4Fh,...

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 19).

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. **Note that in these two burst read modes the address pointer does not cross the boundary that occurs every 128 words; thus, no 128-word address boundary crossing wait states are inserted for linear burst with wrap.**

Downloaded from Arrow.com.

Figure 4. Synchronous Read



7.2.5 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 43 on page 57). 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.

Downloaded from Arrow.com.

Table 20. 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 = Initial data is valid on the 4th rising CLK edge after addresses are latched
		0011 = Initial data is valid on the 5th rising CLK edge after addresses are latched
		⋮
		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)

7.2.5.1 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.

Downloaded from Arrow.com.

7.2.5.2 Wait States

Configuration Register bits 14 to 11 (CR.[14..11]) define the number of delay cycles after the AVD# Low cycle that captures the initial address until the cycle that read data is valid. The bits from 14 to 11 are in most to least significant order. The random address access at the beginning of each read burst takes longer than the subsequent read cycles. The memory bus interface must be told how many cycles to wait before driving valid data then advancing to the next data word. The number of initial wait cycles will vary with the memory clock rate. The number of wait states is found in the wait state table information above. The minimum number of wait cycles is three. The maximum is 13. The default after power-on or hardware reset is 13 cycles.

When the appropriate number of Wait States have occurred, data is output after the rising edge of the CLK. Subsequent words are output t_{BACC} after the rising edge of each successive clock cycle, which automatically increments the internal address counter.

7.2.5.3 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.

7.2.5.4 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.

7.2.5.5 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.

7.2.5.6 Burst Length

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

7.3 Status Register

The status of program and erase operations is provided by a status register. A status register read command is written followed by a read of the status register for each access of the status register information. The Clear Status Register Command will reset the status register. The status register can be read in synchronous or asynchronous mode.

Table 21. 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 higher than Bit 7 are undefined.*
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.*

Downloaded from Arrow.com.

Table 22. 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 23. 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 24. 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.*

Downloaded from Arrow.com.

Table 25. 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 26. 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 27. 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.*

Downloaded from Arrow.com.

Table 28. 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 29. 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

Note:

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

Downloaded from Arrow.com.

7.4 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 555h 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.

7.5 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 20, Back-to-Back Read/Write Cycle Timings - ADM Interface on page 55 shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to the DC Characteristics on page 44 table for read-while-program and read-while-erase current specification.

Downloaded from Arrow.com.

7.6 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. While an address is valid, AVD# must be driven to V_{IL}. Addresses are latched on the rising edge of AVD#, data is latched on the rising edge of WE#.

All writes to the memory are single word length and follow asynchronous timing. However, it is allowed to leave the host and memory interfaces in synchronous mode as long as the host synchronous timing for a single word synchronous write can meet the timing requirements of the memory device write cycle. Generally a synchronous write would include Clock toggling during the write but, it is also allowed for Clock to be at V_{IL} during the write.

If the device is in the Synchronous Read Mode (CR.15 = 0), the addresses are latched on the rising edge of CLK when AVD# is at V_{IL}, while data is latched on the rising edge of WE#. If CLK is held at V_{IL}, addresses are latched on the rising edge of AVD#. CLK should not be held at V_{IH} when writing commands while the device is in Synchronous Read Mode. See the Table 7, Device Bus Operations on page 14 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 in which 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 43, Command Definitions on page 57 for a listing of the commands accepted by the device. I_{CC2} in DC Characteristics on page 44 represents the active current specification for an Embedded Algorithm operation.

7.7 Program/Erase Operations

- 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 25 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 Program Suspend, and Status Read command. Any commands written to the device during the Embedded Erase Algorithm are ignored except Erase Suspend and Status Read command. Reading from a bank that is not programming or erasing is allowed.
- 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.

Downloaded from Arrow.com.

7.7.1 Write Buffer Programming

Write Buffer Programming allows the system to write 1 to 64 bytes 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 + 555h in which programming occurs. Next, the system writes the number of *word locations minus 1* at the Sector Address + 2AAh. This tells the device how many write buffer addresses are loaded with data and therefore when to expect the *Program Buffer to Flash* confirm command. The 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 + 555h. 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 25 for information on these status bits.

The Write Buffer Programming Sequence can be Aborted in the following ways:

- 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.

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

Downloaded from Arrow.com.

Software Functions and Sample Code

Table 30. Write Buffer Program

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Write Buffer Load Command	Write	Sector Address + AAAh	Sector Address + 555h	0025h
2	Write Word Count	Write	Sector Address + 555h	Sector Address + 2AA	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 + AAAh	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 *Cypress Low Level Driver User's Guide* (available on www.cypress.com) for general information on Cypress Flash memory software development guidelines.

```
/* Example: Write Buffer Programming Command            */
/* NOTES: Write buffer programming limited to 32 words. */
/*        All addresses to be written to the flash in   */
/*        one operation must be within the same flash   */
/*        page. A flash page begins at addresses        */
/*        evenly divisible by 0x20.                     */
  UINT16 *src = source_of_data;               /* address of source data          */
  UINT16 *dst = destination_of_data;          /* flash destination address       */
  UINT16 wc   = words_to_program -1;          /* word count (minus 1)            */
*( (UINT16 *)sector_address + 0x555 )     = 0x0025;  /* write write buffer load command */
  *( (UINT16 *)sector_address + 0x2AA)    = wc;      /* write word count (minus 1)      */
do{
  *dst = *src; /* ALL dst MUST BE SAME PAGE */ /* write source data to destination */
  dst++;                                      /* increment destination pointer    */
  src++;                                      /* increment source pointer         */
  wc--;                                       /* decrement word count             */
  }while ( wc >= 0 ); /* do it again                      */

*( (UINT16 *)sector_address + 0x555)    = 0x0029;   /* write confirm command           */
  /* poll for completion */
```


Downloaded from Arrow.com.

7.7.2 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 25 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 31. Program Suspend

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

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

Table 32. Program Resume

Cycle	Operation	Byte Address	Word Address	Data
1	Write	Sector Address + 000h	Sector Address + 000h	0050h

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

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


Downloaded from Arrow.com.

7.7.3 Sector Erase

The sector erase function erases one sector in the memory array (see Table 43 on page 57). 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 25 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 30 for parameters and timing diagrams.

Software Functions and Sample Code

Table 33. Sector Erase

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 + 2AA	0030h

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

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


Downloaded from Arrow.com.

7.7.4 Chip Erase

The chip erase function erases the complete memory array. (See Table 43 on page 57). 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 25 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 30 for parameters and timing diagrams.

Software Functions and Sample Code

Table 34. Chip Erase

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 + 2AA	0010h

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

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


Downloaded from Arrow.com.

7.7.5 Erase Suspend/Erase Resume Commands

The Erase Suspend command allows the system to interrupt a sector erase operation and then read data from, or program data to, 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 25 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 35. Erase Suspend

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

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

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

Table 36. Erase Resume

Cycle	Operation	Byte Address	Word Address	Data
1	Write	Sector Address + 000h	Sector Address + 000h	0030h

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

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


Downloaded from Arrow.com.

7.7.6 Accelerated Program/Sector Erase

Accelerated write buffer programming, and sector erase operations are enabled through the V_{PP} 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 V_{PP}, 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 V_{PP} input, upon completion of the embedded program or erase operation, returns the device to normal operation.

- Simultaneous operations are not supported while V_{PP} is at V_{HH}. The V_{PP} 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 V_{PP} 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.

7.8 Handshaking

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

When 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 CE# input and OE# input is Low, the A/DQ15-A/DQ0 output signals are actively driven. When both of the CE# inputs are High, or the OE# input is High, the A/DQ15-A/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 20 on page 24).

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

7.9 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, tristates 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. Hardware Reset clears the AADM upper address register to zero.

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 16 for timing diagrams

7.10 Software Reset

Software reset is part of the command set (see Table 43 on page 57) 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.

Downloaded from Arrow.com.

Software Functions and Sample Code

Table 37. Reset

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

Note:
Base = Base Address.

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

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


Downloaded from Arrow.com.

8. 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.

8.1 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 x555h and x2AAh, and data is 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 unlocked 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.

8.2 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 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 x555h and x2AAh, 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.

Downloaded from Arrow.com.

8.3 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:

8.3.1 V_{PP} Method

Once V_{PP} input is set to V_{IL}, all program and erase functions are disabled and hence all Sectors (including the Secure Silicon Region) are protected.

8.3.2 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}.

8.3.3 Write Pulse Glitch Protection

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

8.3.4 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.

8.4 SSR Lock

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

8.5 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.

8.5.1 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.

Downloaded from Arrow.com.

8.5.2 Customer Secure Silicon Region

The Customer Secure Silicon Region is typically 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 CL 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.

8.5.3 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 43 on page 57 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 *Cypress Low Level Driver User's Guide* (available on www.cypress.com) for general information on Cypress Flash memory software development guidelines.

Table 38. Secured Silicon Region Entry

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

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

Table 39. Secured Silicon Region Program

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

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


Downloaded from Arrow.com.

Table 40. Secured Silicon Region Exit

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

```
/* Example: SecSi Sector Exit Command */
*( (UINT16 *)base_addr + 0x000 ) = 0x00F0;   /* write SecSi Sector Exit cycle */
```

9. Power Conservation Modes

9.1 Standby Mode

In the standby mode current consumption is greatly reduced, and the outputs (A/DQ15-A/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 44 represents the standby current specification

9.2 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 44 represents the automatic sleep mode current specification.

9.3 Output Disable (OE#)

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

Downloaded from Arrow.com.

10. Electrical Specifications

10.1 Absolute Maximum Ratings

Storage Temperature Plastic Packages	–65°C to +150°C
Ambient Temperature with Power Applied	–65°C to +125°C
Voltage with Respect to Ground: All Inputs and I/Os except as noted below (Note 1)	–0.5 V to VIO + 0.5 V
V_{CC} (Note 1)	–0.5 V to +2.5 V
V_{IO}	–0.5 V to +2.5 V
V_{PP} (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 5. 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 .*
2. *Minimum DC input voltage on pin V_{PP} is -0.5V. During voltage transitions, V_{PP} may overshoot V_{SS} to –2.0 V for periods of up to 20 ns. See Figure 5. Maximum DC voltage on pin V_{PP} is +9.5 V, which may overshoot to 10.5 V for periods up to 20 ns.*
3. *No more than one output may be shorted to ground at a time. Duration of the short circuit should not be greater than one second.*
4. *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this data sheet is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.*

Figure 5. Maximum Negative Overshoot Waveform

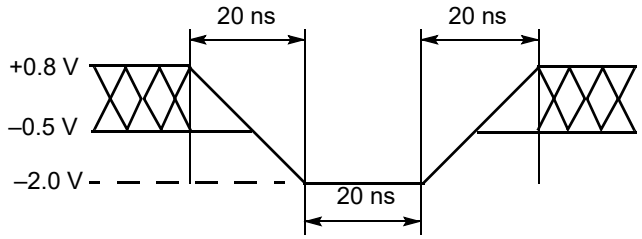
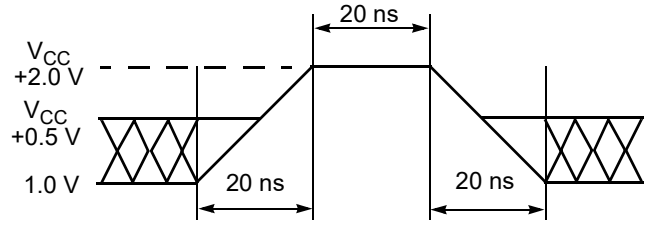


Figure 6. Maximum Positive Overshoot Waveform



Downloaded from Arrow.com.

10.2 Operating Ranges

Wireless (W) Devices	
Ambient Temperature (T_A)	–25°C to +85°C
Industrial (I) Devices	
Ambient Temperature (T_A) (Refer to Publication Number S29VS_XS-R_SP for Industrial Temperature specific differences)	–40°C to +85°C
Supply Voltages	
V_{CC} Supply Voltages	+1.70 V to +1.95 V
V_{IO} Supply Voltages	+1.70 V to +1.95 V
	$V_{CC(min)} \geq V_{IO(min)}$ - 200mV

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

10.3 DC Characteristics

10.3.1 CMOS Compatible

Table 41. 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	83 MHz		35	38	mA
			104 MHz 108 MHz		39	44	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 16	83 MHz		28	30	mA
			104 MHz 108 MHz		32	36	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = Continuous	83 MHz		28	30	mA
			104 MHz 108 MHz		32	36	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	60	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}, $V_{PP} = V_{IH}$	V_{PP}		1	5	µA
			V_{CC}		30	40	mA
I_{CC3}	V_{CC} Standby Current	CE# = RESET# = V_{CC} ± 0.2 V	V_{PP}		1	5	µA
			V_{CC}		30	40	µA
I_{CC4}	V_{CC} Reset Current	RESET# = V_{IL}, CLK = V_{IL}			150	250	µA
I_{CC5}	V_{CC} Active Current (Read While Write) (Continuous Burst) (6)	CE# = V_{IL}, OE# = V_{IH}, $V_{PP} = V_{IH}$	83 MHz		65	70	mA
			104 MHz 108 MHz		71	76	

Downloaded from Arrow.com.

Table 41. CMOS Compatible (Continued)

Parameter	Description	Test Conditions (Notes 1 & 2)		Min	Typ	Max	Unit
I_{CC6}	V_{CC} Sleep Current (4)	CE# = V_{IL}, OE# = V_{IH}			20	40	µA
I_{PP}	Accelerated Program Current (5)	CE# = V_{IL}, OE# = V_{IH}, V_{PP} = 9.5 V	V_{PP}		7	10	mA
			V_{CC}		25	28	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_{PP} and V_{CC} currents.*
6. *I_{CC5} applies while reading the status register during program and erase operations.*
7. *Effect of status register polling during write not included.*

10.4 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 Die	2.0	4.5	6.0	pF
			Dual Die	4.0	9.0	12.0	pF
C_{OUT}	Output Capacitance (DQ, RDY)	V_{OUT} = 0	Single Die	2.0	4.5	6.0	pF
			Dual Die	4.0	9.0	12.0	pF

Notes:

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

Downloaded from Arrow.com.

10.5 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)	83 MHz	2.50 ns
	104 MHz	1.85 ns
	108 MHz	1.85 ns
Transition time (t_T) (CLK input rise and fall times)	83 MHz	2.50 ns
	104 MHz	1.85 ns
	108 MHz	1.85 ns

Figure 7. Input Pulse and Test Point

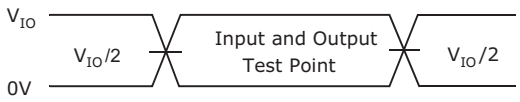
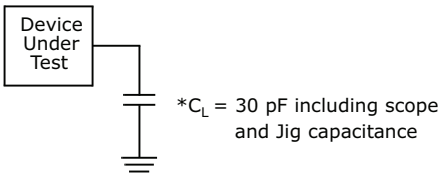
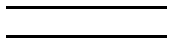
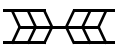


Figure 8. Output Load



10.6 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)

Downloaded from Arrow.com.

10.7 V_{CC} Power-Up and Power Down

During power-up or power-down, V_{CC} must always be greater than or equal to V_{IO} ($V_{CC} \geq V_{IO}$).

The device ignores all inputs until a time delay of t_{VCS} has elapsed after the moment that V_{CC} and V_{IO} both rise above, and stay above the minimum V_{CC} and V_{IO} thresholds. During t_{VCS}, the device is performing power-on reset operations.

During power-down or voltage drops below V_{CC} Lockout maximum (V_{LKO}), the V_{CC} and V_{IO} voltages must drop below V_{CC} Reset (V_{RST}) minimum for a period of t_{PD} for the part to initialize correctly when V_{CC} and V_{IO} again rise to their operating ranges. If during a voltage drop the V_{CC} stays above V_{LKO} maximum, the part will stay initialized and will work correctly when V_{CC} is again above V_{CC} minimum. If the part locks up from improper initialization, a hardware reset can be used to initialize the part correctly.

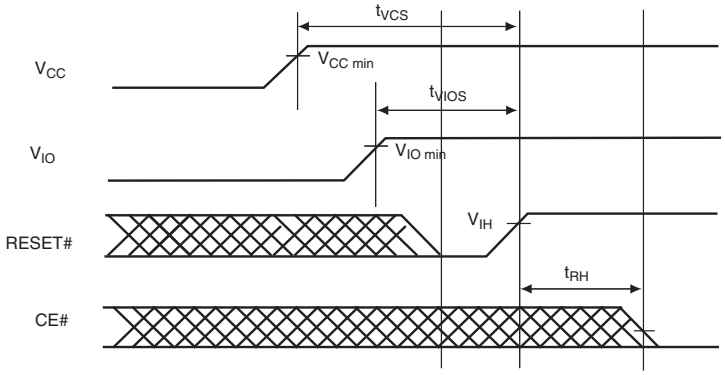
Normal precautions must be ensured for supply decoupling to stabilize the V_{CC} and V_{IO} power supplies. Each device in a system should have the V_{CC} and V_{IO} power supplies decoupled by a suitable capacitor close to the package connections (this capacitor is generally in the order of 0.1 µF). At no time should V_{IO} be greater then 200 mV above V_{CC} ($V_{CC} \geq V_{IO}$ - 200 mV).

Parameter	Description	Test Setup	Value	Unit
t_{VCS}	V_{CC} Setup Time (Note 5)	Min	300	µs
t_{VIOS}	V_{IO} Setup Time	Min	300	µs
t_{RH}	Time between RESET# (high) and CE# (low)	Min	200	ns
V_{RST}	V_{CC} and V_{IO} Low voltage needed to ensure initialization will occur	Min	0.7	V
t_{PD}	Duration of $V_{CC} \leq V_{RST\ (min)}$ (Note 5)	Min	10	µs

Notes:

1. *RESET# must be high after V_{CC} and V_{IO} are higher than V_{CC} minimum.*
2. *$V_{CC} \geq V_{IO}$ – 200 mV during power-up.*
3. *V_{CC} and V_{IO} ramp rate could be non-linear*
4. *V_{CC} and V_{IO} are recommended to be ramped up simultaneously.*
5. *Not 100% tested.*

Figure 9. V_{CC} Power-up Diagram.



Downloaded from Arrow.com.

Figure 10. Power-up

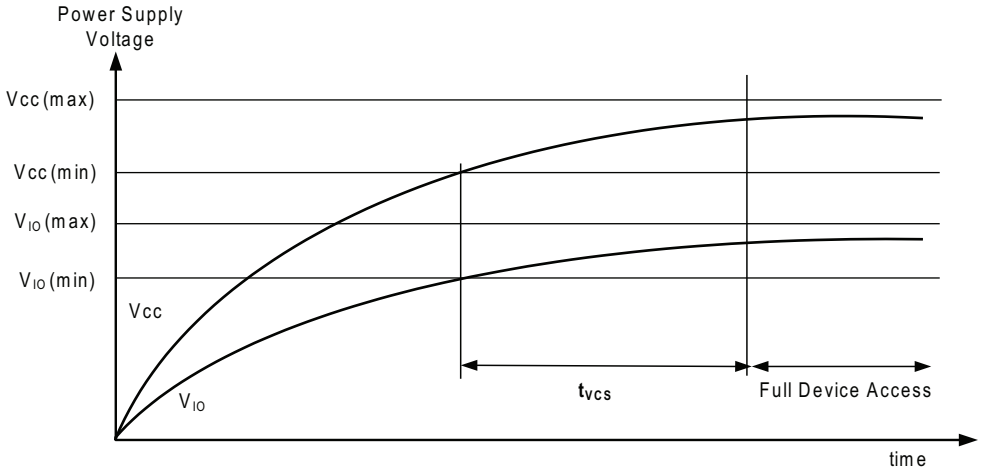
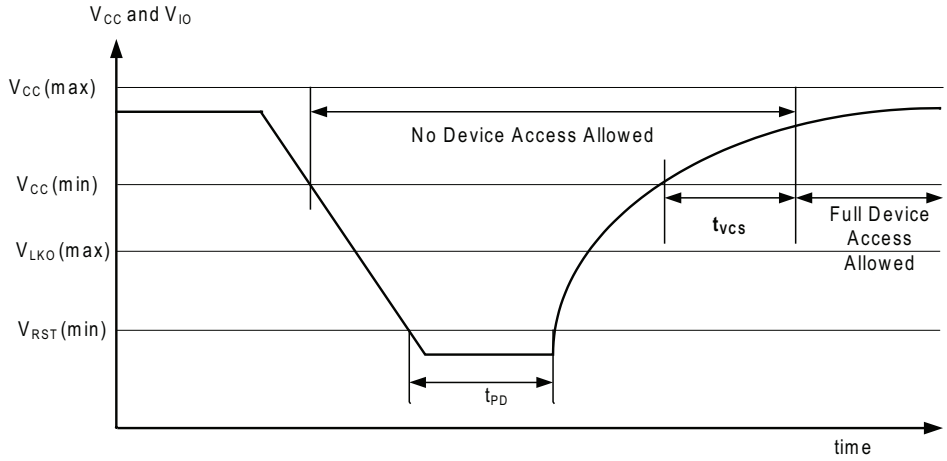


Figure 11. V_{CC} Power-Down and Voltage Drop

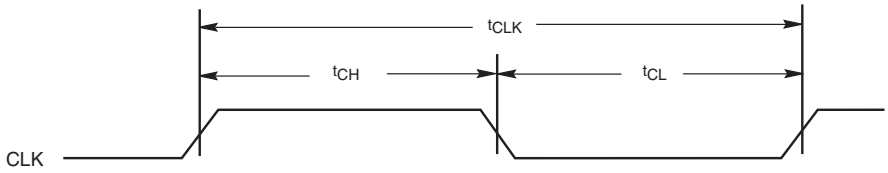


10.8 CLK Characterization

Parameter	Description		108 MHz	Unit
f_{CLK}	CLK Frequency	Max	108	MHz
		Min	DC (1)	
t_{CLK}	CLK Period	Min	9.26	ns
t_{CL}/t_{CH}	CLK Low/High Time	Min	0.40 t_{CLK}	ns

Note:
1. DC for operations other than continuous and 16 word (32 byte) synchronous burst read. See AC Characteristics Table.

Figure 12. CLK Characterization



Downloaded from Arrow.com.

10.9 AC Characteristics

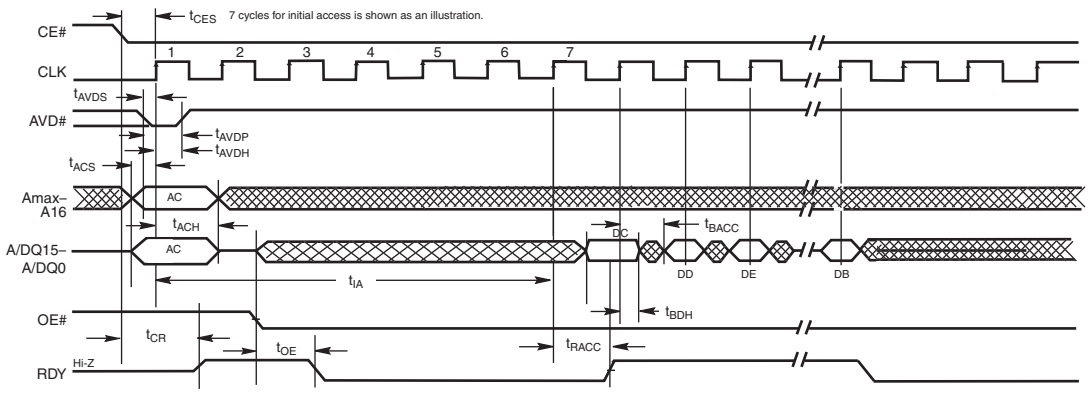
10.9.1 AC Characteristics–Synchronous Burst Read

Parameter (Notes)	Symbol		83 MHz	104 MHz	108 MHz	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	12	9.6	9.26	ns
CLK Rise Time	t_{CLKR}	Max	2.5	1.92	1.852	ns
CLK Fall Time	t_{CLKF}					
CLK High or Low Time	$t_{CLKH/L}$	Min	5	4	3.86	ns
Internal Access Time	t_{IA}	Max	75		72.34	ns
Burst Access Time Valid Clock to Output Delay	t_{BACC}	Max	9	7.6	6.75	ns
AVD# Setup Time to CLK	t_{AVDS}	Min	4		3.38	ns
AVD# Hold Time from CLK	t_{AVDH}	Min	3		2.89	ns
Address Setup Time to CLK	t_{ACS}	Min	4		2.89	ns
Address Hold Time from CLK	t_{ACH}	Min	5		4.82	ns
Data Hold Time from Next Clock Cycle	t_{BDH}	Min	3	2	2	ns
Output Enable to Data	t_{OE}	Max	15			ns
CE# Disable to Output High-Z (2)	t_{CEZ}	Max	10			ns
OE# Disable to Output High-Z (2)	t_{OEZ}	Max	10			ns
CE# Setup Time to CLK	t_{CES}	Min	4		3.38	ns
CLK to RDY valid	t_{RACC}	Max	9	7.6	6.75	ns
CE# low to RDY valid	t_{CR}	Max	10			ns
AVD# Pulse Width	t_{AVDP}	Min	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. *AVD can not be low for 2 subsequent CLK cycles.*

Figure 13. Synchronous Read Mode - ADM Interface



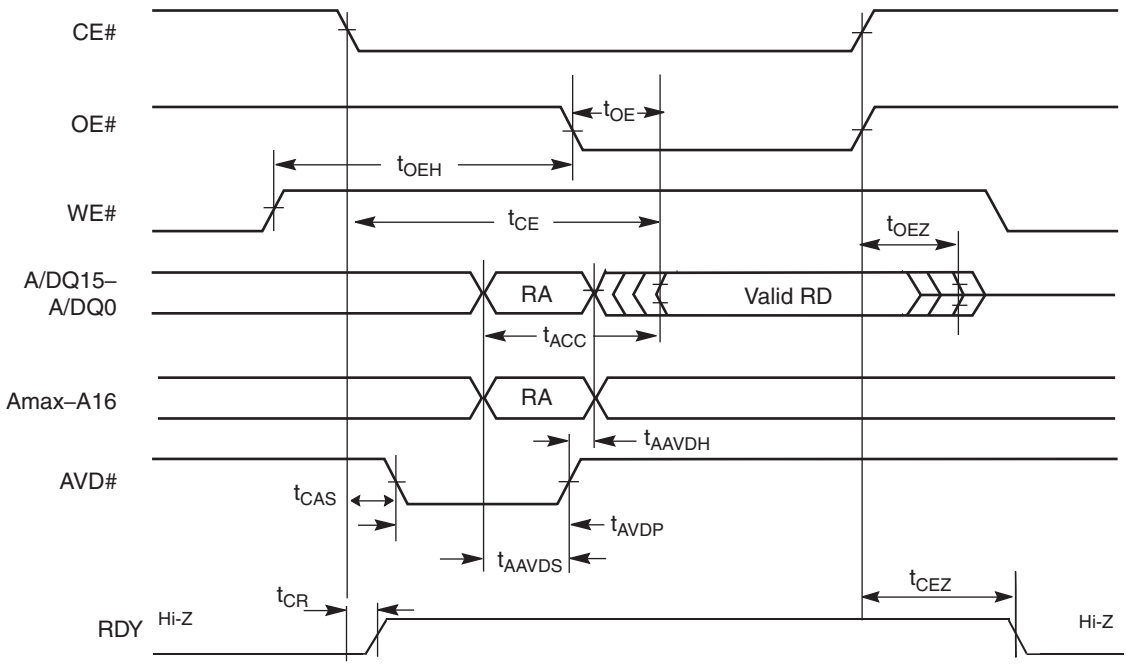
Downloaded from Arrow.com.

10.9.2 AC Characteristics–Asynchronous Read

Parameter	Symbol	Min	Max	Unit
Access Time from CE# Low	t_{CE}	–	80	ns
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}	4	–	
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}	0	–	
CE# Disable to Output & RDY High-Z (1)	t_{CEZ}	–	10	
OE# Disable to Output High-Z (1)	t_{OEZ}	–	10	
AVD# High to OE# Low	t_{AVDO}	4	–	
CE# low to RDY valid	t_{CR}	–	10	
WE# Disable to AVD# Enable	t_{WEA}	9.6	–	
WE# Disable to OE# Enable	t_{OEH}	4	–	

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 14. Asynchronous Mode Read - ADM Interface



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

Downloaded from Arrow.com.

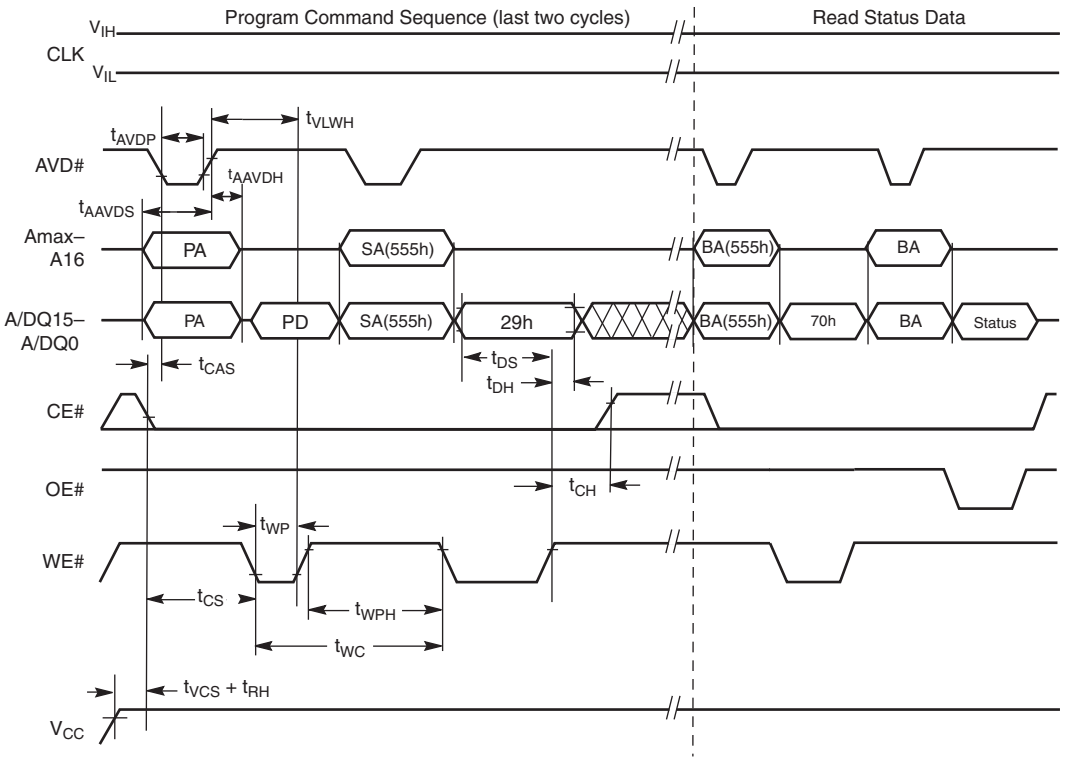
10.9.3 AC Characteristics–Erase/Program Timing

Parameter	Symbol	Min	Typ	Max	Unit
WE# Cycle Time (1)	t_{WC}	60	–	–	ns
AVD# low pulse width	t_{AVDP}	6	–	–	ns
Address Setup to rising edge of AVD#	t_{AAVDS}	4	–	–	ns
Address Hold from rising edge of AVD#	t_{AAVDH}	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#	t_{CH}	0	–	–	ns
WE# Pulse Width	t_{WP}	25	–	–	ns
WE# Pulse Width High	t_{WPH}	20	–	–	ns
Latency Between Read and Write Operations	t_{SRW}	0	–	–	ns
AVD# Disable to WE# Disable	t_{VLWH}	23.5	–	–	ns
WE# Disable to AVD# Enable	t_{WEA}	9.6	–	–	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
Erase Resume to Erase Suspend	t_{ERS}	30	–	–	µs
Program Resume to Program Suspend	t_{PRS}	30	–	–	µs

Note:
1. Sampled, not 100% tested.

Downloaded from Arrow.com.

Figure 15. Asynchronous Program Operation Timings - ADM Interface

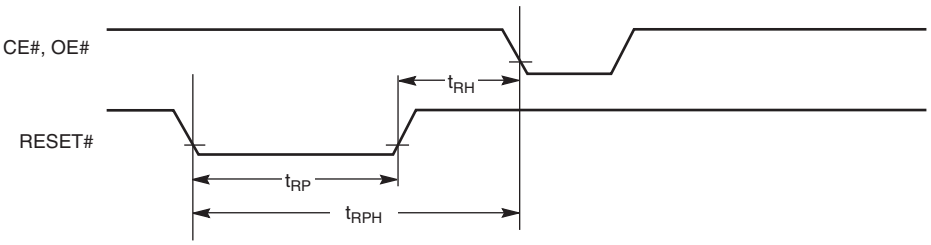


10.9.4 Hardware Reset (Reset#)

Table 42. Warm-Reset

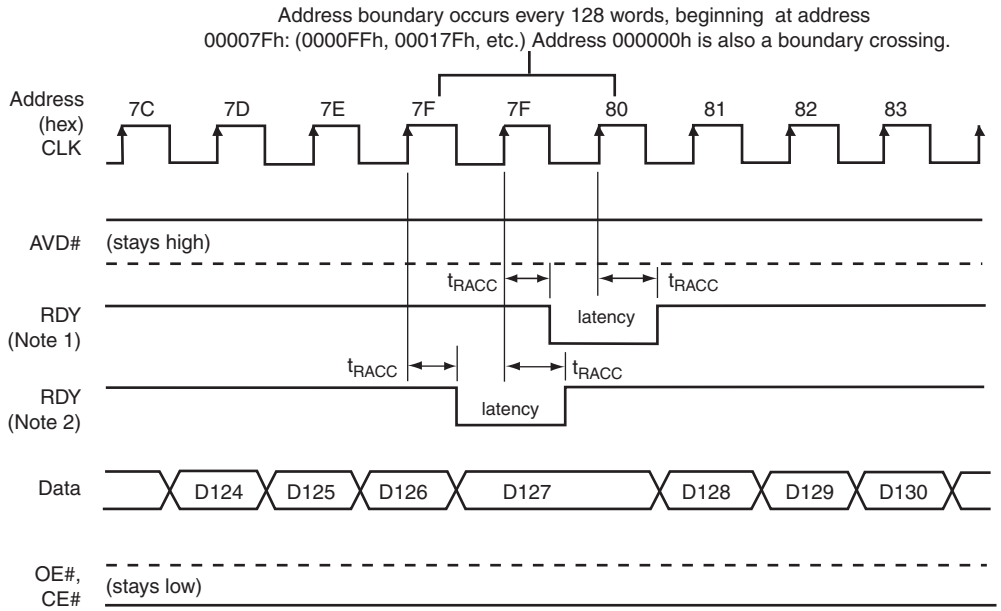
Parameter		Description		All Speed Options	Unit
JEDEC	Std				
	t_{RP}	RESET# Pulse Width	Min	50	ns
	t_{RH}	Reset High Time Before Read	Min	200	ns
	t_{RPH}	RESET# Low to CE# Low	Min	10	us

Figure 16. Reset Timings



Downloaded from Arrow.com.

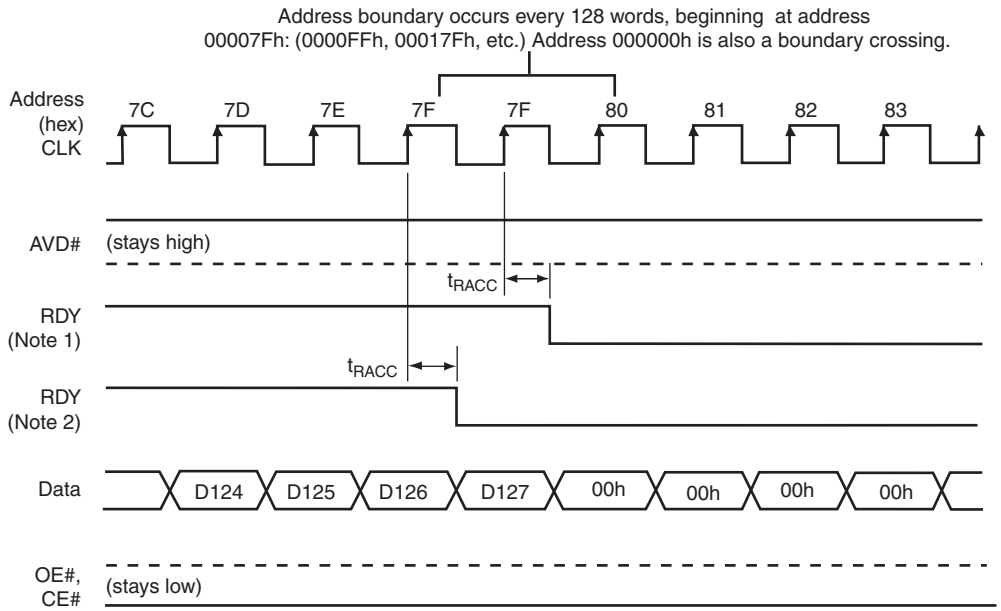
Figure 17. Latency with Boundary Crossing



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 18. Latency with Boundary Crossing into Bank Performing Embedded Operation



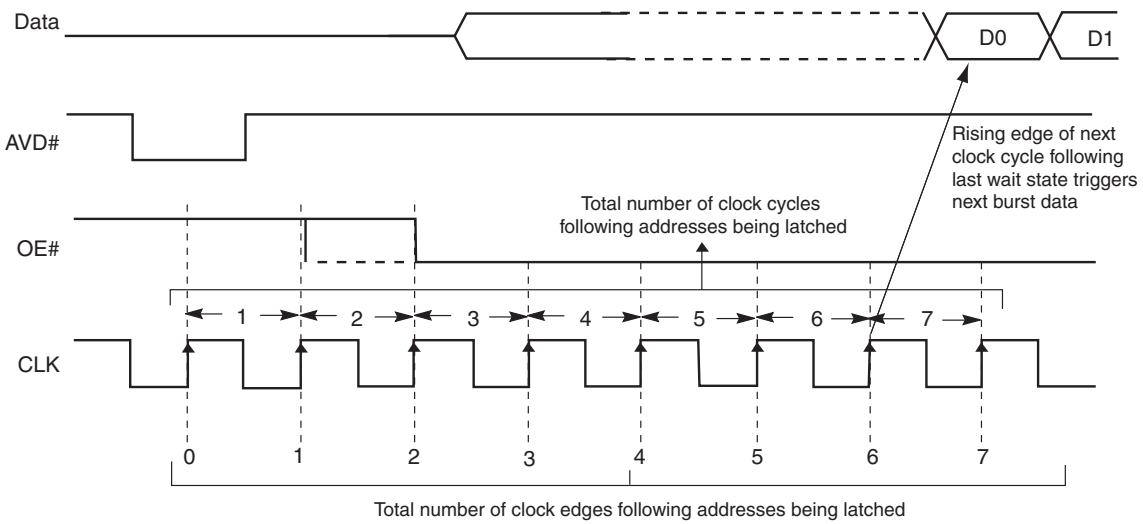
Notes:

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.*

Downloaded from Arrow.com.

10.9.5 Wait State Configuration Register Setup

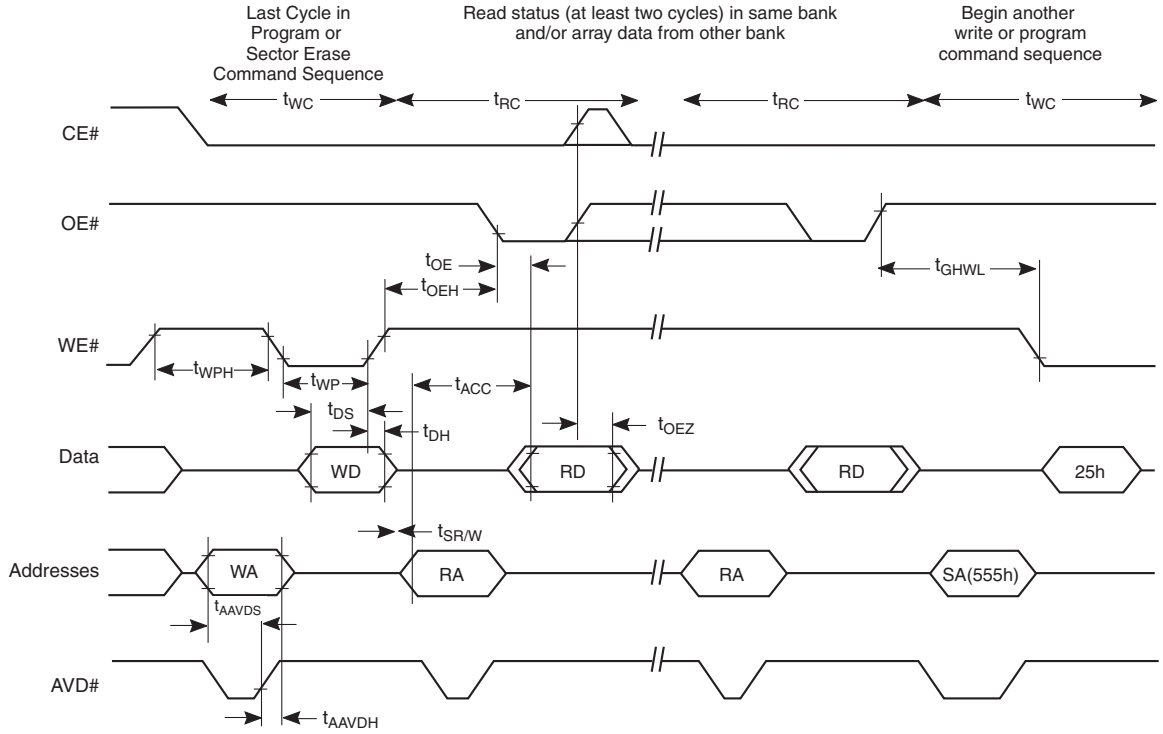
Figure 19. Example of Programmable Wait States



Configuration Register	Programmable Wait States			
CR.14 CR.13 CR.12 CR.11	0000 =	initial data is valid on the	Reserved	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 =			

Downloaded from Arrow.com.

Figure 20. Back-to-Back Read/Write Cycle Timings - ADM Interface



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

Downloaded from Arrow.com.

10.9.6 Erase and Programming Performance

Parameter			Typ (Note 1)	Max (Note 2)	Unit	Comments
Sector Erase Time (Note 6)	128 Kbyte	V_{CC}	0.8/1.3	3.5/5.5	s	(Note 3)
	32 Kbyte	V_{CC}	0.35/0.6	2.0/3.5		
	128 Kbyte	V_{PP}	0.8/1.3	3.5/5.5		
	32 Kbyte	V_{PP}	0.35/0.6	2.0/3.5		
Chip Erase Time (Note 6), (Note 7)		V_{CC}	78/126 (128 Mbit) 155/251 (256 Mbit)	200/325 (128 Mbit) 400/650 (256 Mbit)		
		V_{PP}	78/126 (128 Mbit) 155/251 (256 Mbit)	154/250 (128 Mbit) 308/500 (256 Mbit)		
Single Word Program Time (using Program Buffer)		V_{CC}	170	800	µs	Excludes system level overhead (Note 4)
Effective Word Programming Time using Program Write Buffer		V_{CC}	14.1	94		
		V_{PP}	9	48		
Total 32-Word Buffer Programming Time		V_{CC}	450	3000		
		V_{PP}	288	1540		
Chip Programming Time (using 32 word buffer)		V_{CC}	118 (128 Mbit) 236 (256 Mbit)	157 (128 Mbit) 315 (256 Mbit)	s	Excludes system level overhead (Note 4)
		V_{PP}	76 (128 Mbit) 151 (256 Mbit)	80 (128 Mbit) 160 (256 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 –25°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 43 on page 57 for further information on command definitions.*
5. *The device has a minimum erase and program cycle endurance of 10,000 cycles.*
6. *The first value excludes 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)*

Downloaded from Arrow.com.

11. Appendix

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

11.1 Command Definitions

All values are in hexadecimal. The S29VS-R family of devices are 16-bit word address oriented. Most system address buses, regardless of data bus size, are byte oriented. It is common practice for system designers to shift the address busses. That is, Flash Address A0 is connected to system Address A1, etc. To accommodate the system designers, addresses are listed in both word address and byte address where applicable. The flash address (word) is listed above the system address (byte).

Table 43. Command Definitions

Command Sequence		Cycles	Bus Cycles (Notes 1–4)							
			First		Second		Third		Fourth	
			Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read			RA	RD						
Reset		1	X	F0						
Write Buffer Load (8)		3-34	(SA) 555 (SA) AAA	25	(SA) 2AA (SA) 554	WC	(SA) PA (11)	PD	PA (12)	PD
Buffer to Flash		1	(SA) 555 (SA) AAA	29						
Chip Erase		2	(SA) 555 (SA) AAA	80	(SA) 2AA (SA) 554	10				
Sector Erase		2	(SA) 555 (SA) AAA	80	(SA) 2AA (SA) 554	30				
Read Status Register		2	(SA) 555 (SA) AAA	70	(SA)	RR				
Clear Status Register		1	(SA) 555 (SA) AAA	71						
Program Suspend (5)		1	XXX	51						
Program Resume (5)		1	(SA) 000	50						
Erase Suspend (6)		1	XXX	B0						
Erase Resume (6)		1	(SA) 000	30						
Blank Check (13)		1	(SA) 555 (SA) AAA	33						
Sector Lock/Unlock		3	555 AAA	60	2AA 554	60	SLA	60		
Sector Lock Range		4	555 AAA	60	2AA 554	60	SLA	61	SLA	61
ID/CFI Command Definitions										
ID/CFI	ID/CFI Entry (7) (10)	1	(SA) X55 (SA) XAA	90 or 98						
	ID/CFI Read	1	(SA) RA	data						
	ID/CFI Exit	1	XXX	FO						

Downloaded from Arrow.com.

Table 43. Command Definitions (Continued)

Command Sequence		Cycles	Bus Cycles (Notes 1–4)							
			First		Second		Third		Fourth	
			Addr	Data	Addr	Data	Addr	Data	Addr	Data
Configuration Command Definitions										
Configuration Register	Configuration Register Entry (7) (10)	1	(SA) 555 (SA) AAA	D0						
	Write Buffer Load	3	(SA) 555 (SA) AAA	25	(SA) 2AA (SA) 554	0	(SA) X00	PD		
	Buffer to Flash (Configuration Register)	1	(SA) 555 (SA) AAA	29						
	Configuration Register Read	1	(SA) X00	RR						
	Configuration Register Exit	1	XXX	FO						
SSR Lock Command Definitions										
SSR Lock	SSR Lock Entry (7) (10)	1	(SA) 555 (SA) AAA	40						
	Write Buffer Load (8)	3	(SA) 555 (SA) AAA	25	(SA) 2AA (SA) 554	0	(SA) 00	PD		
	Buffer to Flash	1	(SA) 555 (SA) AAA	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 (7) (10)	1	(SA) 555 (SA) AAA	88						
	Write Buffer Load (8)	3-34	(SA) 555 (SA) AAA	25	(SA) 2AA (SA) 554	WC	(SA) PA	PD	(SA) PA	PD
	Buffer to Flash	1	(SA) 555 (SA) AAA	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
SLA = Sector Lock Address
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.

Downloaded from Arrow.com.

Notes:

1. *See Section 7., Device Operations on page 17 for description of bus operations.*
2. *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.*
3. *Data bits DQ15–DQ8 are don't care in command sequences, except for RD, PD, and WD.*
4. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*
5. *The Program Resume command is valid only during the Program Suspend mode/state.*
6. *The Erase Resume command is valid only during the Erase Suspend mode/state.*
7. *Command is valid when all banks are ready to read array data.*
8. *The total number of cycles in the command sequence is determined by the number of words written to the write buffer.*
9. *V_{PP} must be at V_{HH} during the entire operation of this command.*
10. *Entry commands are needed to enter a specific mode to enable instructions only available within that mode.*
11. *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.*
12. *Subsequent addresses must fall within the same Sector and Page as the initial starting address.*
13. *Blank Check is only functional in Asynchronous Read mode (Configuration Register - CR [15] = 1).*

11.2 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.

Downloaded from Arrow.com.

Table 44. ID/CFI Data

	Word Offset Address	Byte Offset Address	DATA		Description
			VS256R/XS256R	VS128R/XS128R	
Device Identification	(SA) + 00h	(SA) + 00h	0001h		Cypress Manufacturer ID
	(SA) + 01h	(SA) + 02h	007Eh (Top/Bottom)	007Eh (Top/Bottom)	Device ID, Word 1 Extended ID address code. Indicates an extended two byte device ID is located at byte address 1Ch and 1Eh.
	(SA) + 02h	(SA) + 04h	Reserved		Reserved
	(SA) + 03h	(SA) + 06h	Reserved		Reserved
	(SA) + 04h	(SA) + 08h	Reserved		Reserved
	(SA) + 05h	(SA) + 0Ah	Reserved		Reserved
	(SA) + 06h	(SA) + 0Ch	0010h		ID Version
	(SA) + 07h	(SA) + 0Eh	DQ15 - DQ8 = Reserved DQ7 - Factory Lock Bit: 1 = Locked; 0 = Not Locked DQ6 - Customer Lock Bit: 1 = Locked; 0 = Not locked DQ5 - DQ0 = Reserved		Indicator Bits
	(SA) + 08h	(SA) + 10h	Reserved		Reserved
	(SA) + 09h	(SA) + 12h	Reserved		Reserved
	(SA) + 0Ah	(SA) + 14h	Reserved		Reserved
	(SA) + 0Bh	(SA) + 16h	Reserved		Reserved
	(SA) + 0Ch	(SA) + 18h	05h 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 - Reserved		Lower Software Bits
	(SA) + 0Dh	(SA) + 1Ah	Reserved		Upper Software Bits Reserved
	(SA) + 0Eh	(SA) + 1Ch	0064h/Top; 0066h/Bottom	0063h/Top; 0065h/Bottom	High Order Device ID, Word 2
	(SA) + 0Fh	(SA) + 1Eh	0001h (Top/Bottom)	0001h (Top/Bottom)	Low Order Device ID, Word 3

Downloaded from Arrow.com.

Table 44. ID/CFI Data (Continued)

	Word Offset Address	Byte Offset Address	DATA		Description
			VS256R/XS256R	VS128R/XS128R	
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 (Cypress = 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		
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 ms
	(SA) + 22h	(SA) + 44h	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)

Downloaded from Arrow.com.

Table 44. ID/CFI Data (Continued)

	Word Offset Address	Byte Offset Address	DATA		Description
			VS256R/XS256R	VS128R/XS128R	
Common Flash Interface	**Device Geometry Definition**				
	(SA) + 27h	(SA) + 4Eh	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	0002h		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 of identical size within the Erase Block Region. 00h = 1 sector; 01h = 2 sectors 02h = 3 sectors 03h = 4 sectors
			0003h (Bottom Boot)	0003h (Bottom Boot)	
	(SA) + 2Eh	(SA) + 5Ch	0000h		[upper byte]
	(SA) + 2Fh	(SA) + 5Eh	0000h (Top Boot)		[lower byte] - Sector Size in bytes divided by 256 (n [bytes]h = sector size / 256)
			0080h (Bottom Boot)		
	(SA) + 30h	(SA) + 60h	0002h (Top Boot)		[upper byte]
			0000h (Bottom Boot)		
	(SA) + 31h	(SA) + 62h	0003h (Top Boot)	0003h (Top Boot)	Erase block Region 2 Information
			00FEh (Bottom Boot)	007Eh (Bottom Boot)	
	(SA) + 32h	(SA) + 64h	0000h		[upper byte]
	(SA) + 33h	(SA) + 66h	0080h (Top Boot)		[lower byte] - Sector Size in bytes divided by 256 (n [bytes]h = sector size / 256)
			0000h (Bottom Boot)		
	(SA) + 34h	(SA) + 68h	0000h (Top Boot)		[upper byte]
			0002h (Bottom Boot)		

Downloaded from Arrow.com.

Table 44. ID/CFI Data (Continued)

	Word Offset Address	Byte Offset Address	DATA		Description
			VS256R/XS256R	VS128R/XS128R	
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	00E0h	0070h	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	0000h		Page Mode Type 00h = Not Supported 01h = 4-Word Page 02h = 8-Word Page 04h = 16-Word Page
	(SA) + 4Dh	(SA) + 9Ah	0085h		V_{PP} (Acceleration) Supply Minimum 00h = Not Supported D7-D4: Volt D3-D0: 100 millivolt
	(SA) + 4Eh	(SA) + 9Ch	0095h		V_{PP} (Acceleration) Supply Maximum 00h = Not Supported D7-D4: Volt D3-D0: 100 millivolt
	(SA) + 4Fh	(SA) + 9Eh	03h (Top Boot) 02h (Bottom Boot)		Top/Bottom Sector Flag 00h = Uniform 01h = Dual Boot 02h = Bottom boot 03h = Top boot
	(SA) + 50h	(SA) + A0h	0001h		Program Suspend 00h = Not Supported 01h= Supported

Downloaded from Arrow.com.

Table 44. ID/CFI Data (Continued)

	Word Offset Address	Byte Offset Address	DATA		Description
			VS256R/XS256R	VS128R/XS128R	
Common Flash Interface	(SA) + 51h	(SA) + A2h	0000h		Unlock Bypass 00h = Not Supported 01h = 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	0008h		Bank Organization: X= Number of banks
Common Flash Interface	(SA) + 58h	(SA) + B0h	0020h (Top Boot) 0023h (Bottom Boot)	0010h (Top Boot) 0013h (Bottom Boot)	Bank 0 Region Information. X= Number of sectors in bank
	(SA) + 59h	(SA) + B2h	0020h	0010h	Bank 1 Region Information. X= Number of sectors in bank
	(SA) + 5Ah	(SA) + B4h	0020h	0010h	Bank 2 Region Information. X= Number of sectors in bank
	(SA) + 5Bh	(SA) + B6h	0020h	0010h	Bank 3 Region Information. X= Number of sectors in bank
	(SA) + 5Ch	(SA) + B8h	0020h	0010h	Bank 4 Region Information. X= Number of sectors in bank
	(SA) + 5Dh	(SA) + BAh	0020h	0010h	Bank 5 Region Information. X= Number of sectors in bank
	(SA) + 5Eh	(SA) + BCh	0020h	0010h	Bank 6 Region Information. X= Number of sectors in bank
	(SA) + 5Fh	(SA) + BEh	0020h (Bottom Boot) 0023h (Top Boot)	0010h (Bottom Boot) 0013h (Top Boot)	Bank 7 Region Information. X= Number of sectors in bank

Downloaded from Arrow.com.

Figure 21. Asynchronous Read - AADM Interface

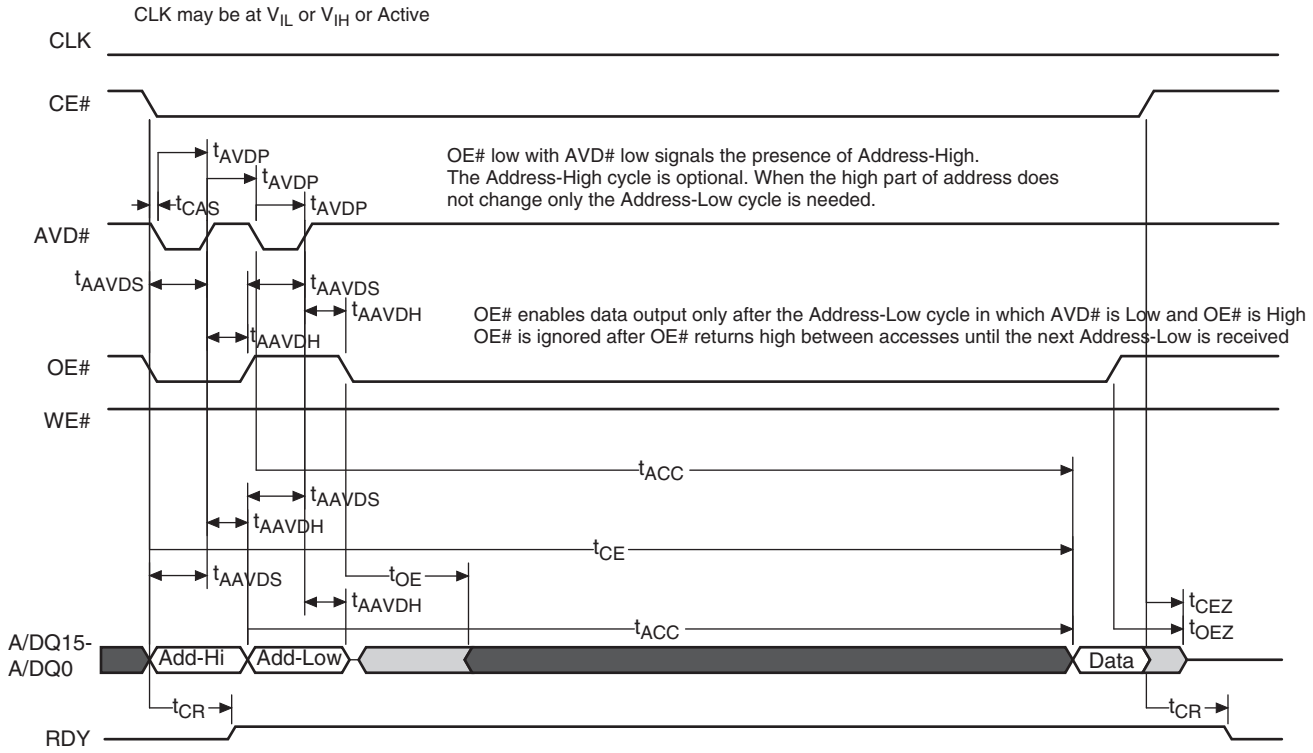
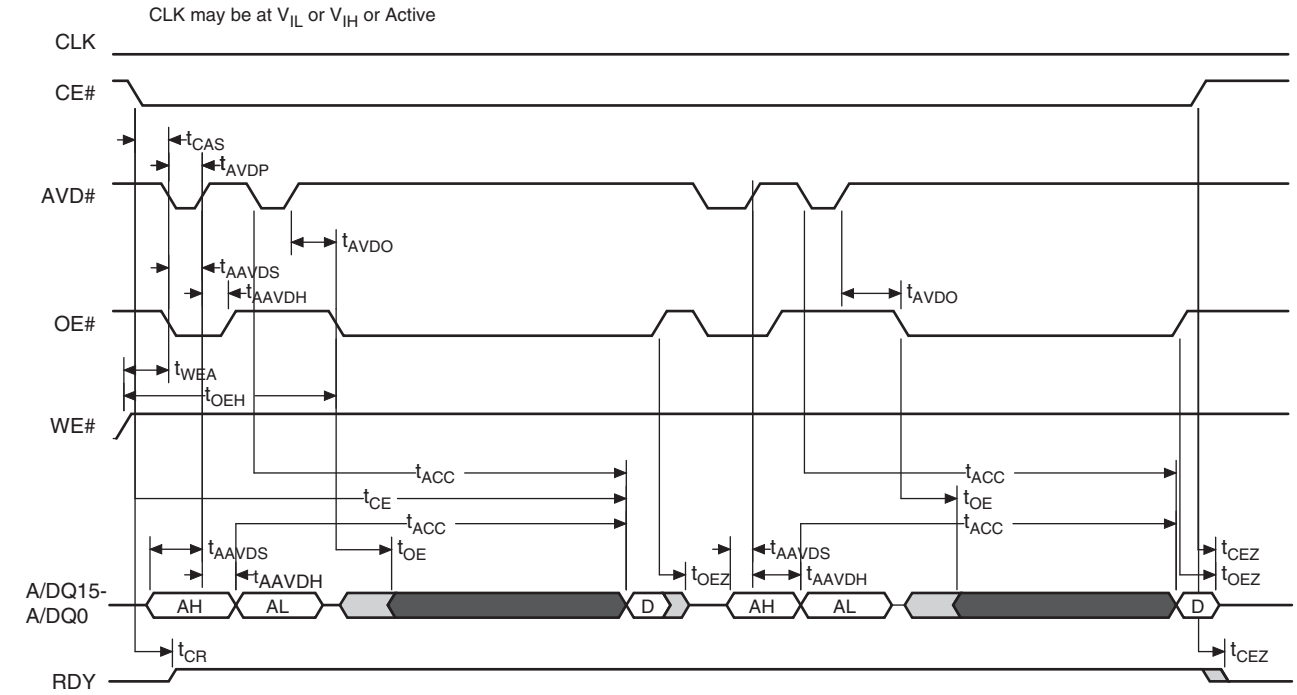


Figure 22. Asynchronous Read Followed By Read - AADM Interface



Downloaded from Arrow.com.

Figure 23. Asynchronous Read Followed By Write - AADM Interface

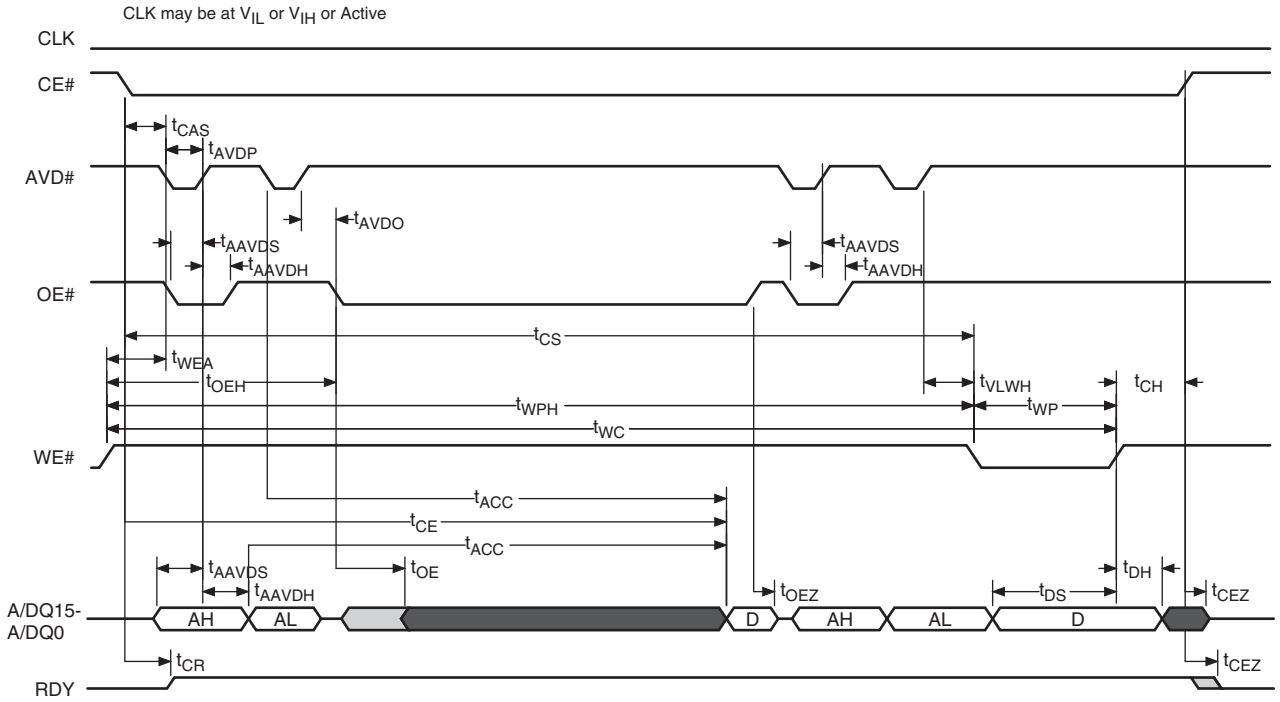
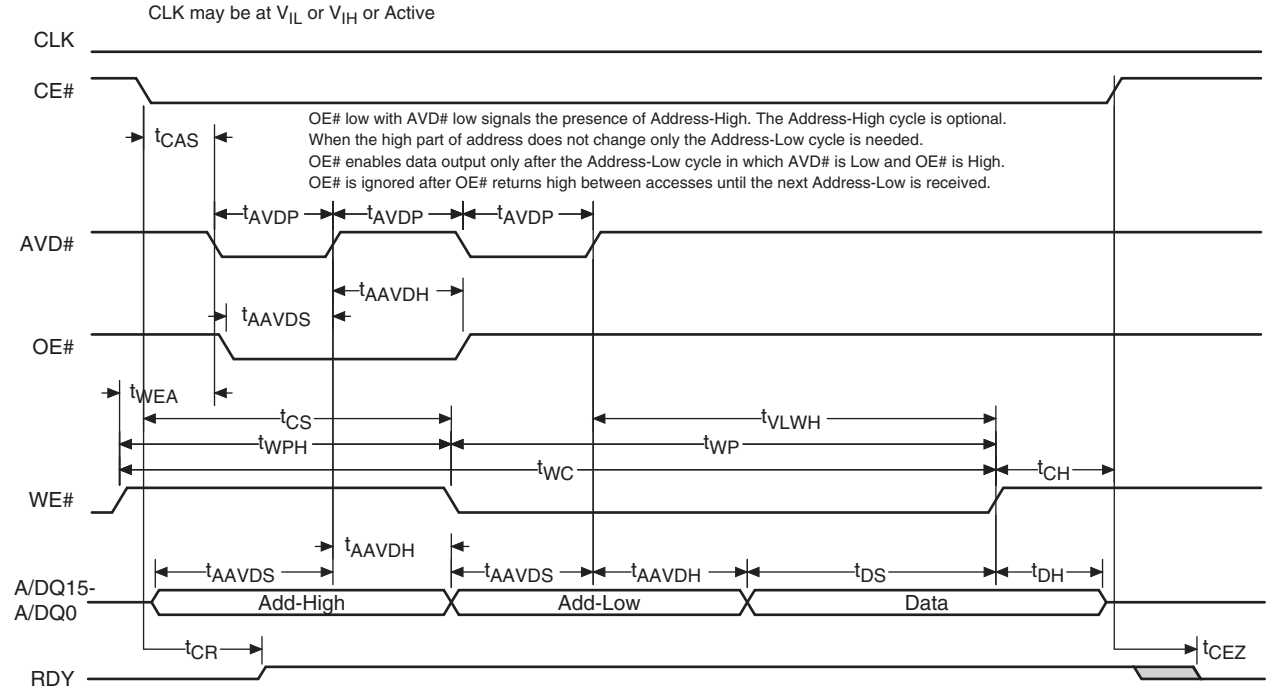


Figure 24. Asynchronous Write - AADM Interface



Downloaded from Arrow.com.

Figure 25. Asynchronous Write Followed By Read - AADM Interface

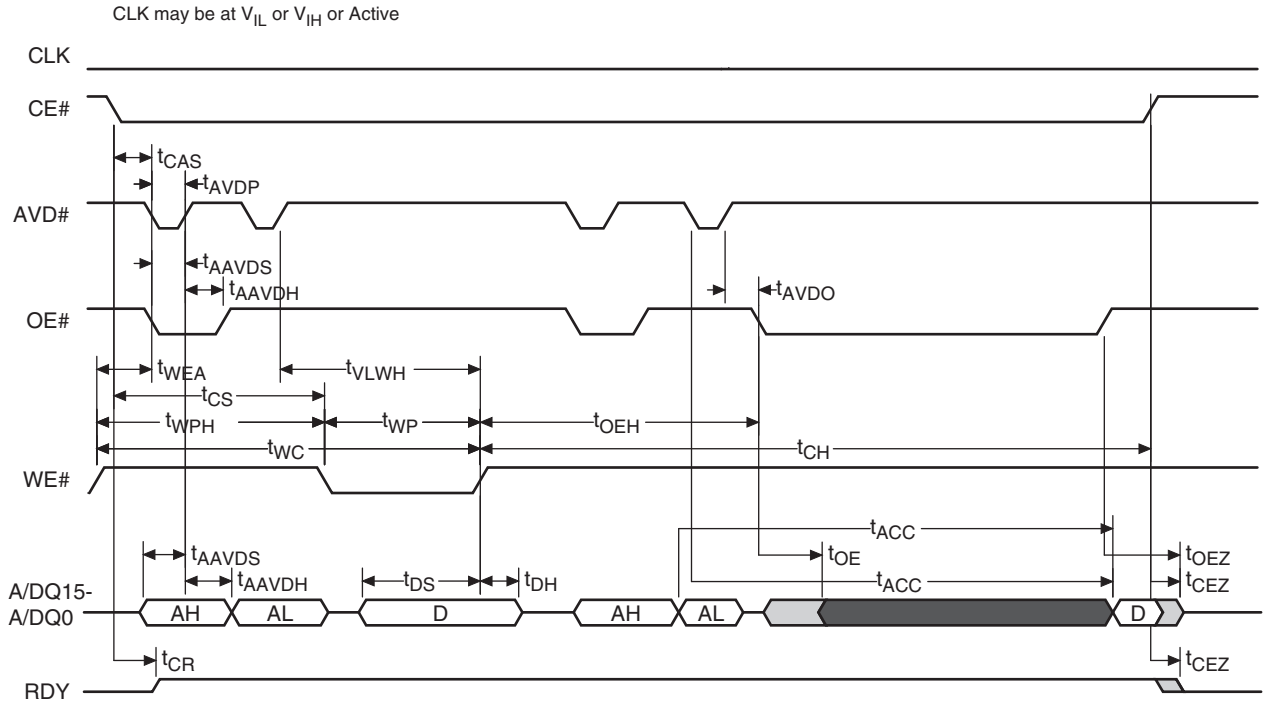
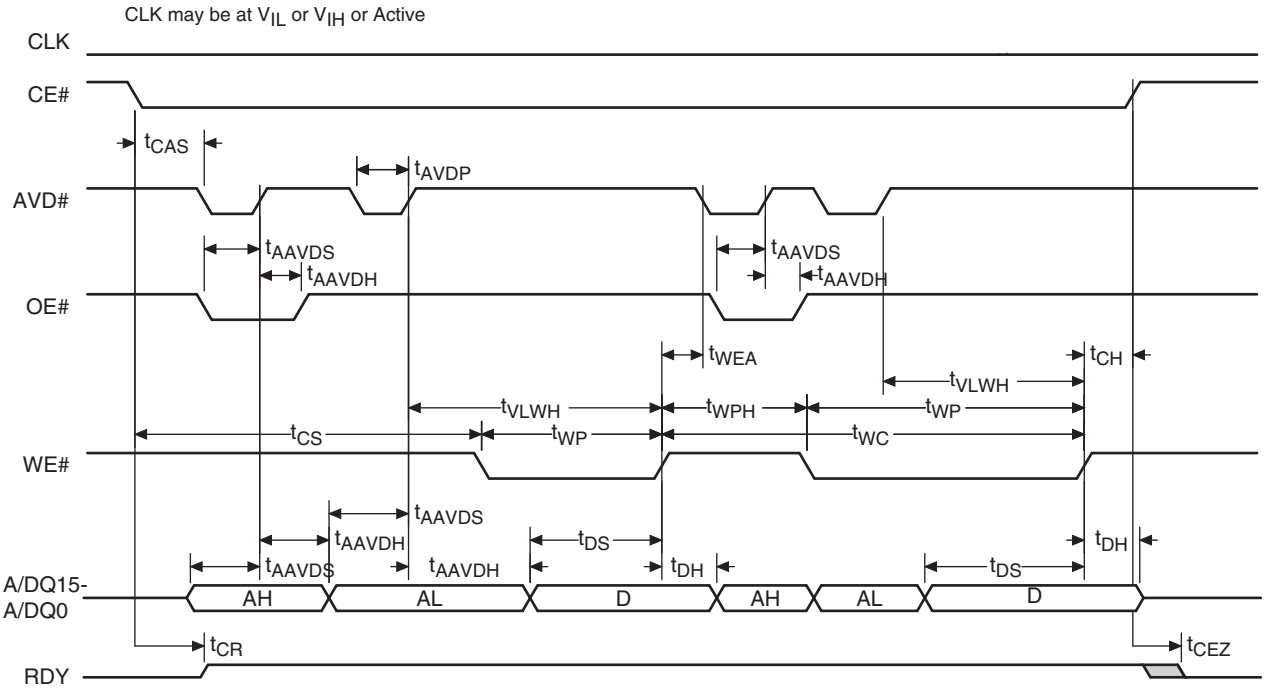


Figure 26. Asynchronous Write Followed By Write - AADM Interface



Downloaded from Arrow.com.

Figure 27. Synchronous Read Wrapped Burst Address Low Only - AADM Interface

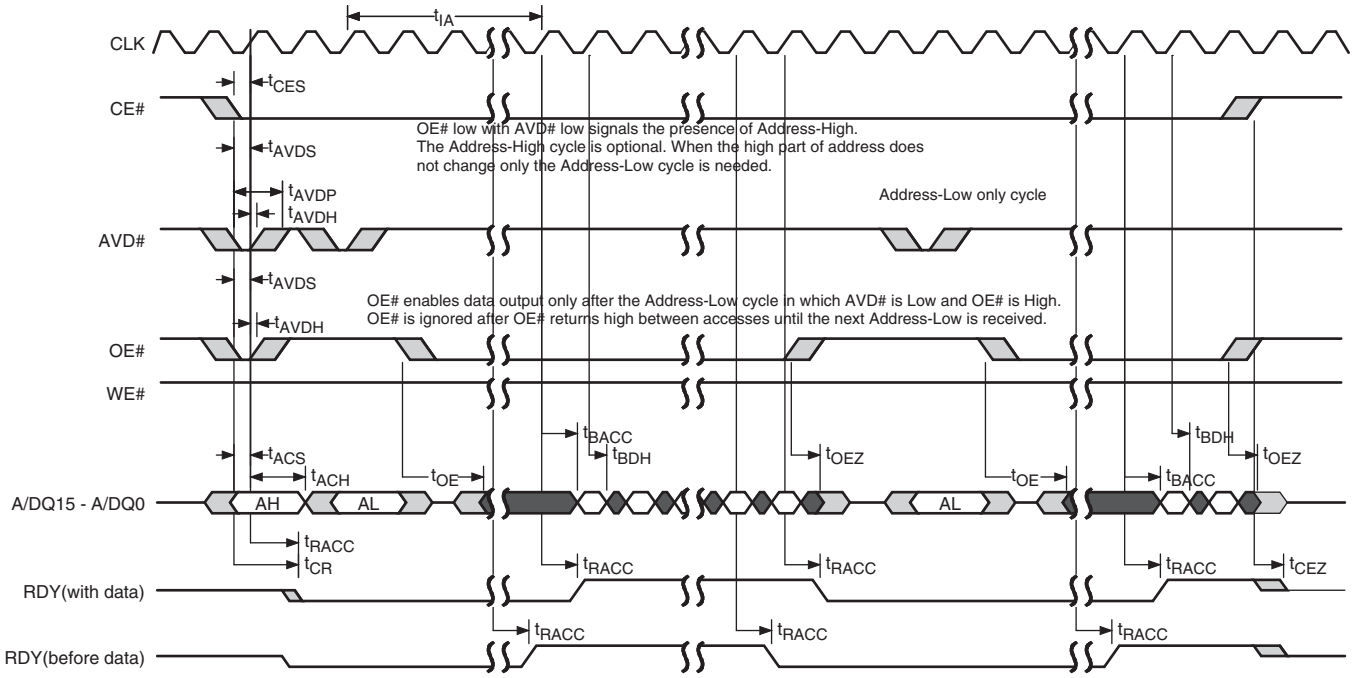
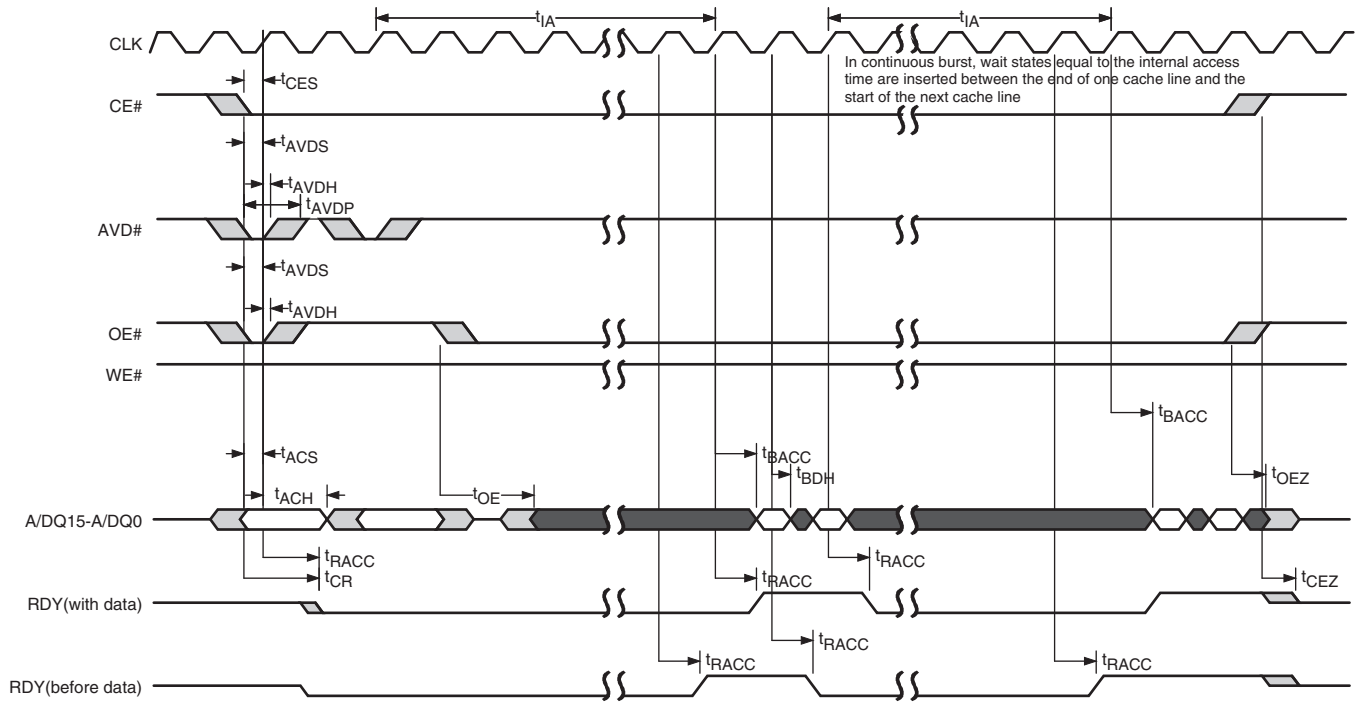


Figure 28. Synchronous Read Continuous Burst - AADM Interface



Downloaded from Arrow.com.

Figure 29. Synchronous Read Wrapped Burst - AADM Interface

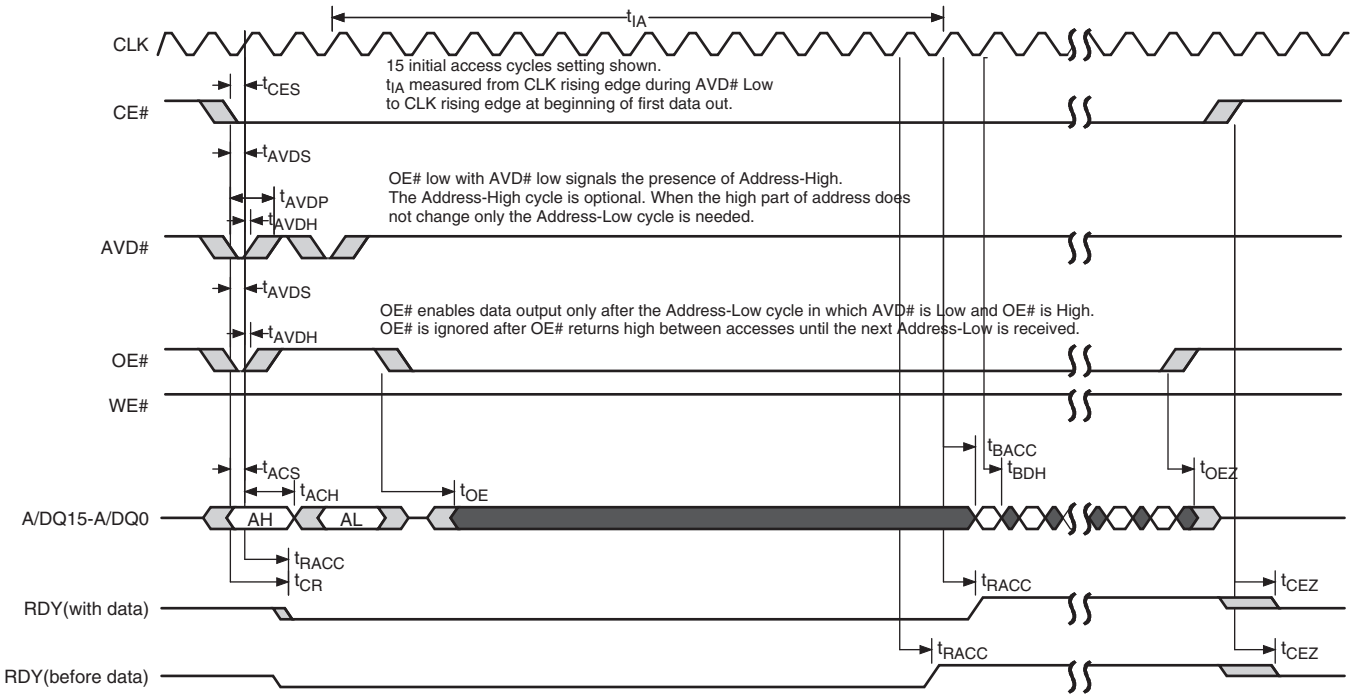
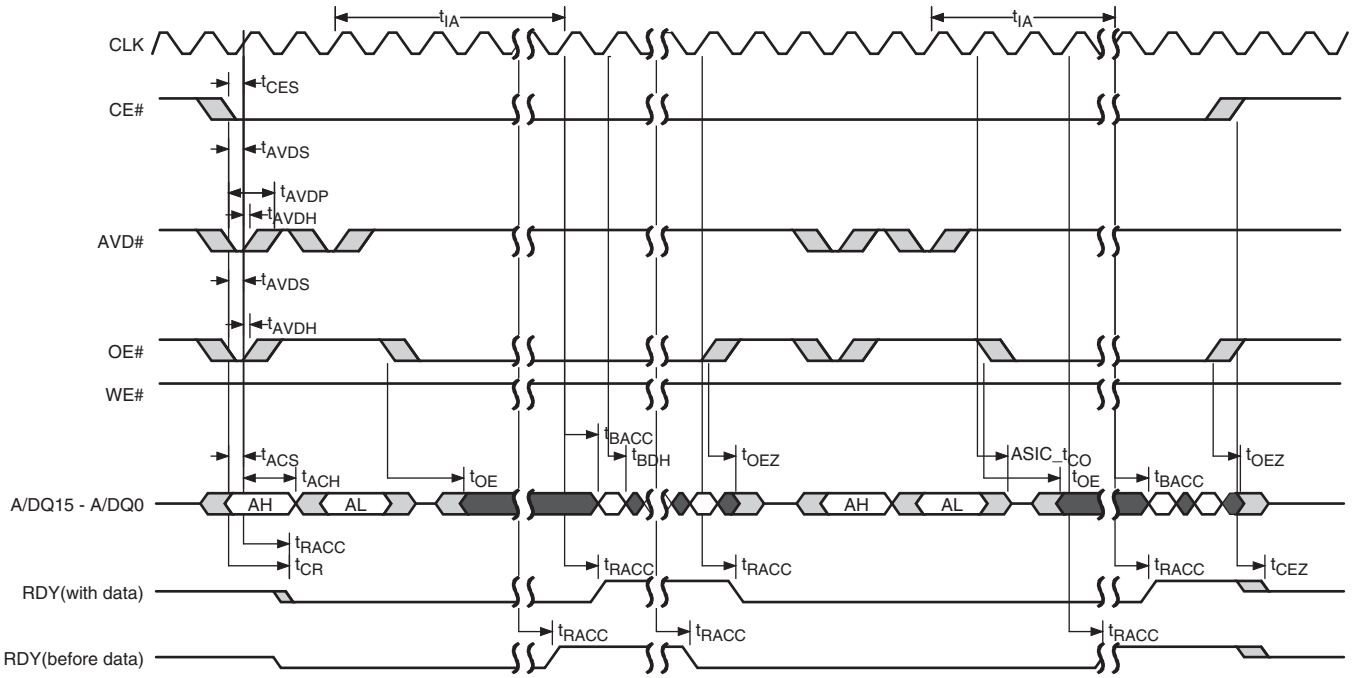


Figure 30. Synchronous Read Followed By Read Burst - AADM Interface



Downloaded from Arrow.com.

Figure 31. Synchronous Read Followed By Write - AADM Interface

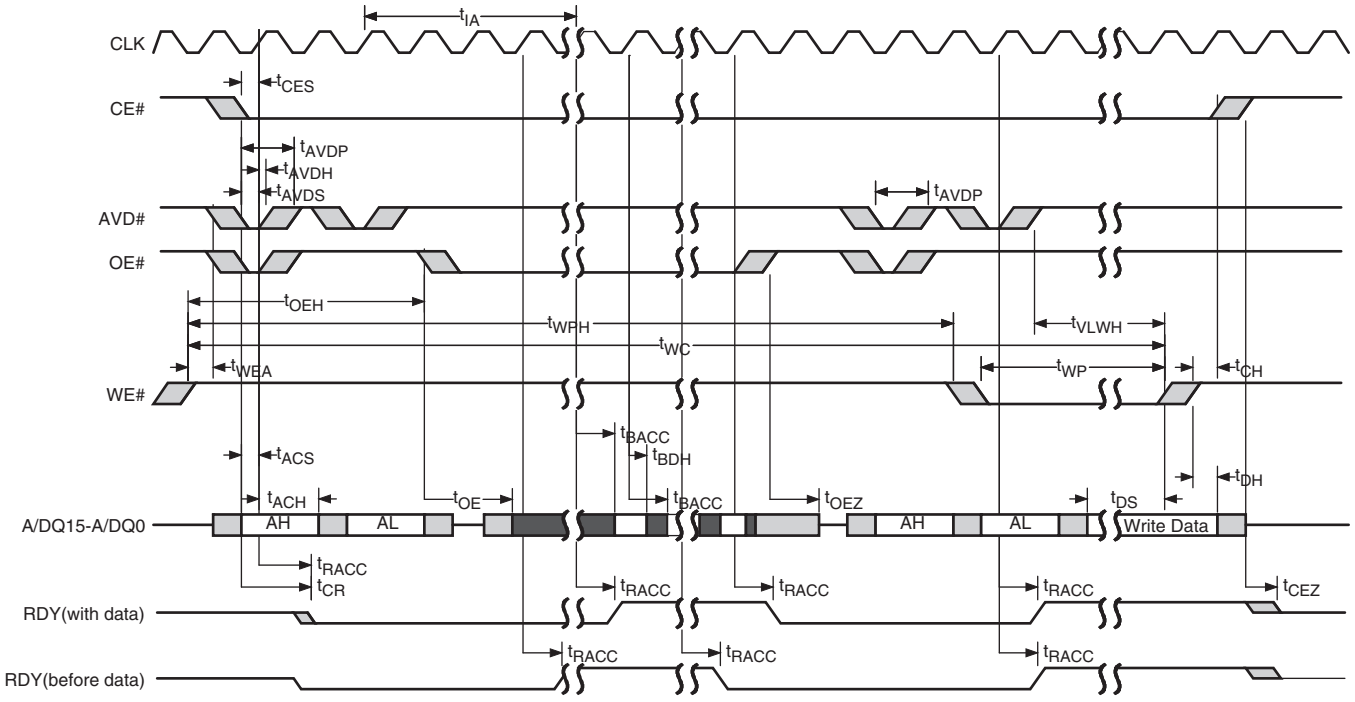
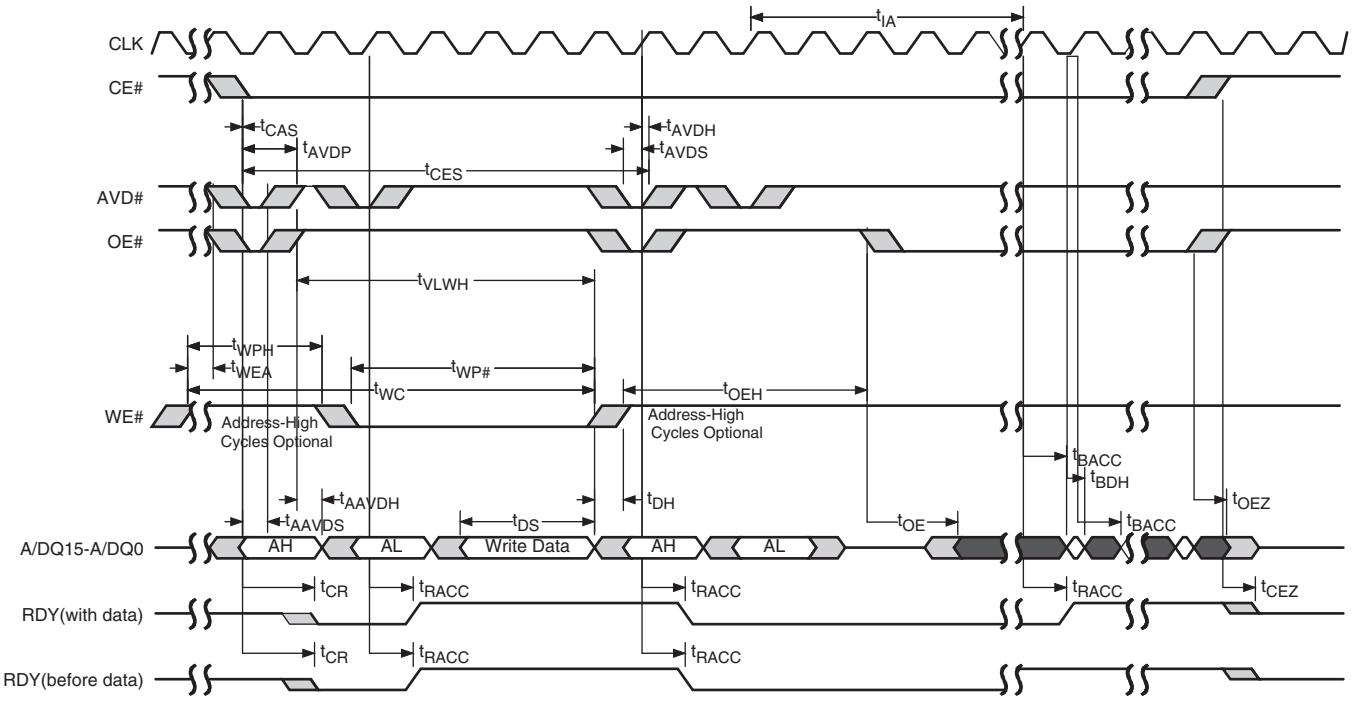
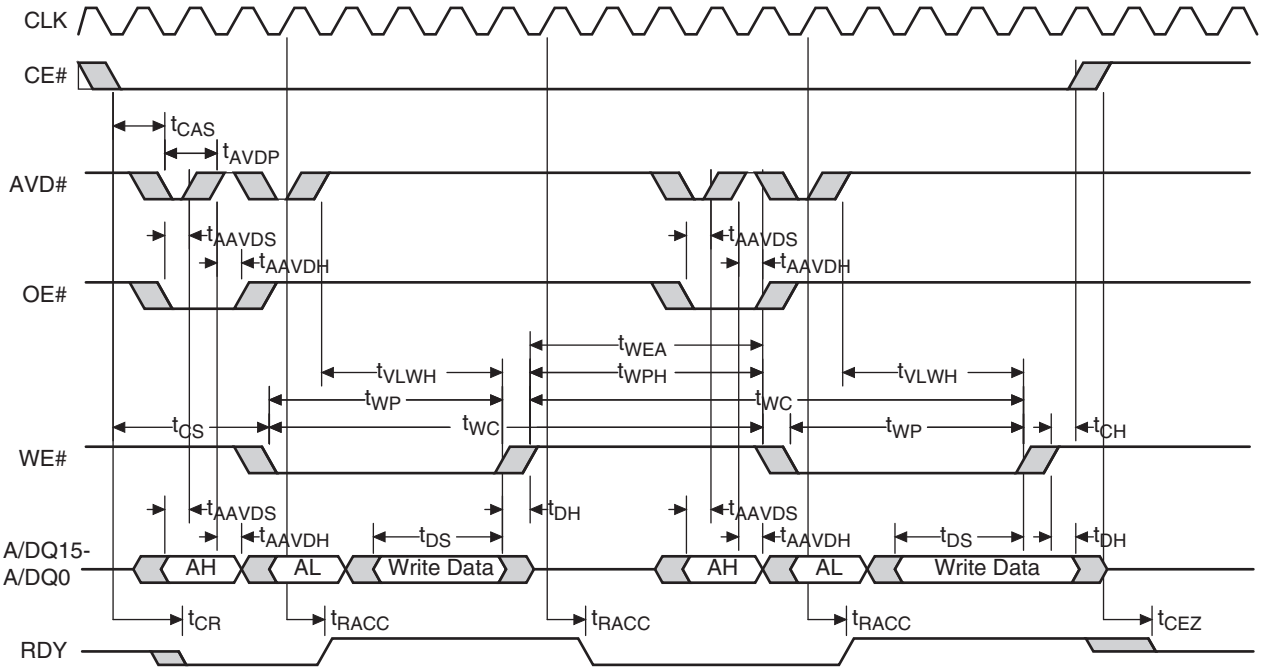


Figure 32. Synchronous Write Followed By Read Burst - AADM Interface



Downloaded from Arrow.com.

Figure 33. Synchronous Write Followed By Write - AADM Interface



Downloaded from Arrow.com.

12. Revision History

Document History Page

Document Title: S29VS256R/S29VS128R/S29XS256R/S29XS128R, 256/128-Mbit (32/16 Mbyte), 1.8 V, 16-bit Data Bus, Multiplexed MirrorBit® Flash Document Number: 002-00833				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	–	WIOB	05/15/2008	Initial release
*A	–	WIOB	08/01/2008	**DC Characteristics** Changed some values in the CMOS Compatible table **Device ID and Common Flash Memory Interface Address Map** Changed some values in the ID/CFI Data table **Memory Address Map** Added memory address map
*B	–	WIOB	09/12/2008	**Physical Dimensions/Connection Diagrams** Updated ball positions
*C	–	WIOB	03/10/2009	**Blank Check Command** Functional in Asynchronous Read Mode only **DC Characteristics** Changed some ICCB values **Global** Added 108 MHz; removed 66 MHz
*D	–	WIOB	05/26/2010	**Global** Modified document title **Features** Clarified some points **Ordering Information and Valid Combinations** Added Industrial Temperature range option **Address/Data Interface** Corrected typo **Device Bus Operations Table** Corrected A/DQ15-A/DQ0 column information for Asynchronous Read **Asynchronous Read** Clarified asynchronous read operation. **S29XS-R AADM Access** Clarified asynchronous AADM read access. **S29VS-R ADM Access** Standardized logic Low and High descriptions to VIL and VIH. Clarified wait states required by initial access and internal boundary crossings. **S29XS-R AADM Access** Standardized logic Low and High descriptions to VIL and VIH. Clarified wait states required by initial access and internal boundary crossings. **Writing Commands/Command Sequences** Clarified device behavior. **Program/Erase Operations** Removed redundant information. **Sector Lock Range Command** Clarified Sector Lock Range behavior **Figure Synchronous Read Mode** Added “ADM Interface” label **Figure Asynchronous Mode Read** Added “ADM Interface” label **Figure Asynchronous Program Operation Timings** Added “ADM Interface” label **Figure Back-to-Back Read/Write Cycle Timings** Added “ADM Interface” label

Downloaded from Arrow.com.

Document History Page (Continued)

Document Title: S29VS256R/S29VS128R/S29XS256R/S29XS128R, 256/128-Mbit (32/16 Mbyte), 1.8 V, 16-bit Data Bus, Multiplexed MirrorBit® Flash
Document Number: 002-00833

Rev.	ECN No.	Orig. of Change	Submission Date	Description of Change
*D (Cont.)	–	WIOB	05/26/2010	**Figure Latency with Boundary Crossing** Corrected CR8 setting in Notes 1 and 2 **Figure Latency with Boundary Crossing into Bank Performing Embedded Operation** Corrected CR8 setting in Notes 1 and 2 **Figures Asynchronous Read - AADM Interface to Asynchronous Write Followed By Write - AADM Interface** Clarified CLK waveform behavior **Figure Synchronous Write Followed By Read Burst - AADM Interface** Corrected Figure title **ADM Interface (S29VS256R and S29VS128R)** Clarified traditional interface **Table Wait State vs. Frequency** Modified title and added note **Table Address Latency for 10–13 Wait States** Added note **Table Address Latency for 9 Wait States** Added note **Figure Synchronous Read** Removed note 1 **CLK Characterization** Removed note 2 **Erase and Programming Performance** Corrected note 2
*E	–	WIOB	07/22/2010	**DC Characteristics** Changed ICC Read test conditions to OE#=H with relevant values **Performance Characteristics** Updated tables **Erase and Programming Performance** Changed typical programming times
*F	–	WIOB	11/18/2010	**Erase and Programming Performance** Changed maximum chip erase times **ID/CFI Data** Corrected Data and Description for Word Offset 03h, 55h, 56h Corrected Data for Word Offset 1Dh, 1Eh, 52h
*G	–	WIOB	07/30/2012	**Command Definitions** Corrected number of cycles for Write Buffer Load
*H	5043055	WIOB	12/17/2015	Changed status from Advance to Final. Updated to Cypress template.
*I	5632749	WIOB	02/16/2017	Updated Electrical Specifications: Updated V_{CC} Power-Up and Power Down: Added description. Added V_{RST}, t_{PD} parameters and their corresponding details in the table. Added Figure 10 and Figure 11. Updated to new template.
*J	5967674	AESATMP8	11/15/2017	Updated Cypress Logo and Copyright.
*K	6269679	PRIT	08/09/2018	Updated to new template. Completing Sunset Review.

Downloaded from Arrow.com.

Document History Page (Continued)

Document Title: S29VS256R/S29VS128R/S29XS256R/S29XS128R, 256/128-Mbit (32/16 Mbyte), 1.8 V, 16-bit Data Bus, Multiplexed MirrorBit® Flash Document Number: 002-00833				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
*L	6581939	PRIT	05/27/2019	Updated Physical Dimensions/Connection Diagrams: Updated Special Handling Instructions for FBGA Package: Updated VDJ044-44-Ball Very Thin Fine-Pitch Ball Grid Array, 6.2 mm x 7.7 mm: Removed existing spec. Added spec 002-24745 **. Updated to new template. Completing Sunset Review.

Downloaded from Arrow.com.

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

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

Products

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

PSoC® Solutions

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

Cypress Developer Community

Community | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

© Cypress Semiconductor Corporation, 2008–2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

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

Downloaded from Arrow.com.